Journal of Robotics and Control (JRC)
Volume 3, Issue 4, July 2022
ISSN: 2715-5072, DOI: 10.18196/jrc.v3i4.

Nonlinear Model Predictive Control-Based Collision Avoidance for Mobile Robot

Omar Y. Ismael [1*], Mohammed Almaged [2], Abdulla Ibrahim Abdulla [3]
[1, 2, 3] Department of Systems and Control Engineering, Ninevah University, Mosul, Iraq
Email: [1] omar.ismael@uoninevah.edu.iq, [2] mohammed.younus@uoninevah.edu.iq, [3] abdullah.abdullah@uoninevah.edu.iq
*Corresponding Author

Abstract—**This work proposes an efficient and safe single-layer Nonlinear Model Predictive Control (NMPC) system based on LiDAR to solve the problem of autonomous navigation in cluttered environments with previously unidentified static and dynamic obstacles. Initially, LiDAR sensor data is collected. Then, the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm, is used to cluster the (Lidar) points that belong to each obstacle together. Moreover, a Minimum Euclidean Distance (MED) between the robot and each obstacle with the aid of a safety margin is utilized to implement safety-critical obstacles avoidance. After that, to impose avoidance constraints with feasibility guarantees and without compromising stability, a NMPC framework for set-point stabilization is taken into consideration with a design strategy based on terminal inequality and equality constraints. Finally, a case study with an omnidirectional wheeled mobile robot (OWMR) is presented to assess the proposed NMPC formulation for set-point stabilization. Furthermore, the efficacy of the proposed system is tested by experiments in simulated real-world scenarios using a robot simulator named CoppeliaSim in combination with MATLAB which utilizes the CasADi Toolbox. The proposed control framework shows a positive performance in a narrow-cluttered environment with unknown obstacles.**

Keywords— NMPC; DBSCAN; Set-point Stabilization; Obstacle Avoidance

I. INTRODUCTION

From a personal and professional perspective, the technological advancement seen in the previous several decades has changed how individuals reshaped their everyday routines. Manufacturing robots, autonomous automobiles, underwater vehicles, home and warehouse autonomous robots, Unmanned Aerial Vehicles (UAVs), and many more sophisticated robotics systems and autonomous vehicles are all included in the technological development. Yet, as these systems spread, new technological problems appear, shedding more light on the difficulties in developing secure and dependable intelligent motion systems that can cooperate with people and complicated environments. In addition to the unique tasks that each system is intended for, dynamic control problems and path planning are critical when autonomous navigation is being considered. Applications for autonomous navigational systems include manufacturing [1], precision agriculture [2], cinematographic filming [3], mining [4], underwater and space exploration [5, 6], and pipeline inspection [7].

It is difficult to maintain autonomy in such a dynamic, and unstructured environment. The following presents the most challenging areas: i) reliable obstacle detection and forecasting in an unstructured environment, ii) examination of obstacles' uncertainty and parametric representation, and iii) algorithm of motion planning for dynamic obstacle trajectories in real-time. Planning and control problems are frequently dealt separately in the literature using multi-layer schemes [8-13]. If the connection between the layers is not handled properly in certain circumstances, the total output might show performance degradation. Control systems in particular frequently assume that the intended reference is attainable, though this hypothesis may not be accurate if the reference is specified in spite of the system's inherent dynamics limitations and operational constraints. Thus, it is crucial to research methods that take control and planning problems into account in a single, colligated scheme.

Among the studies that address these problems at single-layer, those that focus on optimum control are particularly intriguing since they allow for the definition of path planning by utilizing the design of the optimization problem through its cost function and constraints [14, 15]. From the standpoint of the control algorithm, Model Predictive Control (MPC) techniques in particular can provide convergence guarantees and online stability while solving the path planning problem [16]. The majority of research in the literature suggests single-layer algorithms while taking into account that MPC creates extra constraints for obstacle avoidance to solve the problem [17, 18]. Model predictive control (MPC) has become more popular in the control community as a result of its explicit handling of restricted control problems. Over a predefined prediction horizon, MPC determines a future control sequence while reducing an objective function where a set of system state and control action constraints are achieved. Different versions of MPC, such as linear MPC and NMPC, have been employed to address the aforementioned objectives of control for mobile robots. [19–21] provide research that employed linear MPC; in these investigations, MPC has only been utilized to accomplish the objective of path or trajectory tracking. The nonlinear model of motion is employed by NMPC, which has been utilized for regulation problems [22, 23]; tracking problems [24, 25], and both [26, 31]. Although stability of a finite horizon MPC is not trivially guaranteed [32, 33], it has been demonstrated that stability may be ensured by employing a terminal state equality constraint [34-36].

Robots must be able to recognize surrounding obstacles quickly and effectively in order to perform safety-critical navigation in dynamic situations. These can be achieved by the developments in sensor technology. Mobile robots, for instance, may look ahead a specific distance using LIDAR, Radar, and cameras, delivering information about the terrain and other traffic data [37]. Based on visual data from cameras, several approaches for object detection and tracking are utilized [38-42]. Nevertheless, in poor lighting and weather, this strategy does not work effectively. A different technique uses point cloud data; [43] developed a system for dynamic environment perception based on a LiDAR sensor. In this method the dynamic obstacles are enclosed with minimum bounding ellipses, and a stable dynamic obstacle avoidance is achieved. However, in cluttered environments, when there is a short distance between two or more obstacles, the ellipsoids or other shapes can overlap with each other and then, prevent the robot from navigating between these obstacles especially, when the path is narrow. This may result in the loss of recursive feasibility, in the way of avoiding obstacles that demand a large detour from the initial path. In this work, obstacles are clustered by the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [44] after collecting the LiDAR sensor data, and the minimum Euclidean distance between the robot and each obstacle is calculated. The proposed algorithm performs well in a narrow-cluttered environment with unknown static and dynamic obstacles.

Numerous techniques, including artificial potential fields [45], collision-free flight corridor [43] [46], DWA (Dynamic Window Approach) [47], gradient maps [48], social forces [49], and pre-computed motion primitives library [50] [51], can be used to avoid collisions in static and dynamic environments. These techniques, however, are ineffective in contexts with more complicated or fast-moving obstacles. The proposed strategy is based on MPC, which has been increasingly popular recently [52-54]. [43] presented a planning strategy in dynamic, unstructured environments based on Model Predictive Contouring Control (MPCC). Nevertheless, this algorithm only effectively avoids pedestrians, making it challenging to successfully avoid more complicated obstacles. In order to assure safety, [55] suggested a model predictive control architecture, and [56] examined its viability and safety by employing discrete time Dynamic Control Barrier Function (D-CBF) limitations in a receding horizon manner. This algorithm, however, struggles in a narrow-cluttered environment.

This paper proposes a NMPC algorithm with obstacle avoidance capabilities for a set-point stabilization motion system. The DBSCAN algorithm is adopted for clustering the dynamic and static obstacles after collecting the LiDAR sensor information. The environmental obstacles are induced as additional constraints, which are represented in this work as having minimum distance forms. Numerical findings are provided taking into account the set-point stabilization methodology used for an OWMR with holonomic constraints to support the suggested control strategy. A simulation is executed using the CoppeliaSim robot simulator in conjunction with MATLAB R2023a, which utilizes the CasADi Toolbox [57] with the Interior Point OPTimizer (IPOPT) solver [58], to evaluate the performance of the proposed method.

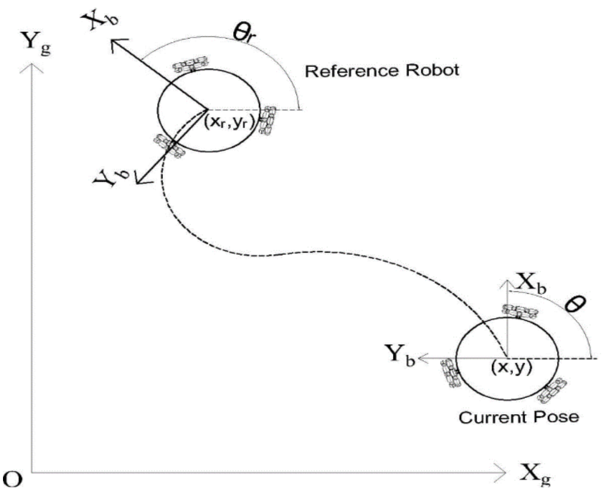

Fig. 1 OWMR Kinematic.

This work's contributions may be summed up as follows:

- New set-point stabilization control framework to establish collision-free trajectory in static and dynamic environments with previously unknown obstacles.
- It proposes a safe method for the detection of obstacles based on minimum Euclidean distance rather than enclosing the obstacles with a circle or minimum bounding ellipse.
- The proposed control framework can handle a large number of obstacles at the same time rapidly and efficiently.
- The effectiveness and real-time performance of the obstacles avoidance algorithm are tested in experiments similar to real-world scenarios using the CoppeliaSim robot simulator.

The rest of this paper is structured as follows: The kinematic model of the OWMR is described in Sect. 2. The NMPC framework for obstacle avoidance is presented in Sect. 3, numerical results are provided in Sect. 4 to support the suggested control strategy, and the study is concluded in Sect. 5.

II. KINEMATIC MODEL OF OWMR

The body frame (X_b, Y_b) and the global frame (X_g, Y_g) are the coordinate frames utilized in the modeling of the OWMR as illustrated in Fig. 1. The body frame is attached to the origin of the moving robot while the global frame is assigned to the fixed ground. θ is the angle that indicates the robot's orientation in the world frame. Every wheel is spaced equally by (L_g) to the center of mass of the OWMR (R). An OWMR kinematic model without slipping can be obtained as follows [59]:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = R^T(\theta)\,\mathbf{u} \tag{1}$$

here $\mathbf{x} = [x \quad y \quad \theta]^T \in \mathbb{R}^3$ represents the state vector in the global frame and $\mathbf{u} = [v_x \quad v_y \quad \omega]^T \in \mathbb{R}^3$ denotes the input vector that characterizes the robot velocities vector measured in the body frame whereas v_x, v_y and ω symbolize robot translational and rotational velocities respectively. $R(\theta)$ denotes the orthonormal rotation matrix that transforms between the robot's coordinate system and the global coordinate system and can be written as:

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

By substituting (2) in (1):

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{3}$$

The system (3) is controllable as it has its accessibility rank condition globally satisfied, and is in the control-affine form (4).

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} v_x + \begin{bmatrix} -\sin\theta \\ \cos\theta \\ 0 \end{bmatrix} v_y + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega \tag{4}$$

Taking the wheel velocities into consideration and with respect to the robot body coordinates, the lower level kinematic model can be defined as:

$$the\ \dot{\boldsymbol{\phi}} = \frac{1}{r} \begin{bmatrix} \sqrt{3}/2 & -1/2 & L_g \\ 0 & 1 & L_g \\ -\sqrt{3}/2 & -1/2 & L_g \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix} \tag{5}$$

where the $\dot{\boldsymbol{\phi}} = [\dot{\phi}_1 \quad \dot{\phi}_2 \quad \dot{\phi}_3]^T \in \mathbb{R}^3$ represents the vector of wheels angular velocities, and r denotes the radius of each wheel of the robot. The maximum wheel velocity is constrained by $\dot{\phi}_m$, namely $\forall i : \dot{\phi}_i \leq \dot{\phi}_m$, where the subscript (i = 1, 2, 3) represents the i^{th} wheel velocity since the motor's voltage and current are magnitudes restricted.

As shown in Fig. 1, a reference robot is defined to demonstrate the objective of the control algorithm which is the stabilization of an OWMR to a permissible equilibrium. This reference robot is subjected to the same constraints as system (1), and possesses a reference state vector $\mathbf{x}_r = [x_r \quad y_r \quad \theta_r]^T \in \mathbb{R}^3$ and a reference control vector $\mathbf{u}_r = [v_{xr} \quad v_{yr} \quad \omega_r]^T \in \mathbb{R}^3$. Therefore, the kinematic motion model can be represented as follow:

$$\dot{\mathbf{x}}_r = \begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \cos\theta_r & -\sin\theta_r & 0 \\ \sin\theta_r & \cos\theta_r & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_{xr} \\ v_{yr} \\ \omega_r \end{bmatrix} \tag{6}$$

The reference control vector $\mathbf{u}_r$ has zero values for both angular and linear velocities, and the reference state vector $\mathbf{x}_r$ has a constant value corresponding to the desired target pose for the point stabilization problem. The primary goal of the control algorithm at this point can be specified as simultaneous robot stabilization while providing safe navigation. The controller ought to also take into consideration the map boundaries, previously unknown obstacles, and the robot geometry. Furthermore, to design a practical controller, the robots' inputs saturation margins must be considered. The designed NMPC architecture shown below addresses the aforementioned difficulties and problems.

III. NONLINEAR MODEL PREDICTIVE CONTROL

NMPC is considered to be one of the most reliable optimal control techniques and it is utilized in this work due to its ability to easily deal with the system constraints and take future prediction into the controller design in regard to nonlinear system defined through the following differential equation:

$$\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)), \mathbf{x}(0) = \mathbf{x}_0 \tag{7}$$

where $f : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^3$ is the nonlinear mapping created by the robot model (7).

The objective of the controller is to determine an admissible control input $\mathbf{u}^*(t)$ that will lead system (7) to drive toward the equilibrium point described by:

$$\begin{aligned} \mathbf{x}_e(t) &= \mathbf{x}_r(t) - \mathbf{x}(t) = 0 \\ \mathbf{u}_e(t) &= \mathbf{u}_r(t) - \mathbf{u}(t) = 0 \end{aligned} \tag{8}$$

The aim of the control technique is to produce a minimum weighted cost function (J) over a prediction horizon (T) as described in (9) [60]:

$$J(t, \mathbf{x}_e(t), \mathbf{u}_e(t)) = P(\mathbf{x}_e(t+T)) + \int_t^{t+T} \ell(\tau, \mathbf{x}_e(\tau), \mathbf{u}_e(\tau))\, d\tau \tag{9}$$

Here, $\ell(\tau, \mathbf{x}_e(\tau), \mathbf{u}_e(\tau))$ is the running cost function that is integrated over T that is obtained from:

$$\ell(\tau, \mathbf{x}_e(\tau), \mathbf{u}_e(\tau)) = \mathbf{x}_e^T(\tau) Q \mathbf{x}_e(\tau) + \mathbf{u}_e^T(\tau) R \mathbf{u}_e(\tau) \tag{10}$$

The terminal state penalty $P(\mathbf{x}_e(t+T))$, assessed at the last step of the optimization horizon is presented as:

$$P(\mathbf{x}_e(t+T)) = \frac{1}{2} \mathbf{x}_e^T(\tau) F \mathbf{x}_e(\tau) \tag{11}$$

The Q , R and F represent the positive definite symmetric weight matrices while and $\mathbf{x}_e$ and $\mathbf{u}_e$ define the robot's state and control vector errors respectively. At time (t) to avoid obstacles in the environment, NMPC optimization problem can be formed as following:

$$\begin{aligned} &\min_{\mathbf{u}^*} \ J(t, \mathbf{x}_e(t), \mathbf{u}_e(t)) \\ \text{Subject to:} \quad &\dot{\mathbf{x}}(t) = f(\mathbf{x}(t), \mathbf{u}(t)) \\ &\mathbf{x}(t) \in X, \ \ (\tau \in [t, t+T]) \\ &\mathbf{u}(t) \in U, \ \ (\tau \in [t, t+T]) \end{aligned} \tag{12}$$

At which the set $X \in \mathbb{R}^3$ and the set $U \in \mathbb{R}^3$ identify the allowable state and control sets as determined by the upcoming sets of constraints. Firstly, the box constraints for the map boundaries and the OWMR input saturation limits that are provided by:

$$\begin{aligned} x_{min} &\leq x \leq x_{max} \\ y_{min} &\leq y \leq y_{max} \\ v_{x\,min} &\leq v_x \leq v_{x\,max} \\ v_{y\,min} &\leq v_y \leq v_{y\,max} \\ \omega_{min} &\leq \omega \leq \omega_{max} \end{aligned} \tag{13}$$

The map margins are indicated by the sets (x_{min}, x_{max}) and (y_{min}, y_{max}), and the robot input saturation limit described by $(v_{x\,min},\ v_{x\,max})$, $(v_{y\,min},\ v_{y\,max})$ and $(\omega_{min},\ \omega_{max})$.

Secondly, the following environmental constraint is taken into account to avoid dynamic and static obstacles:

$$\forall k: \sqrt{((x - xobs_k)^2 + (y - yobs_k)^2} \geq (R_b + \beta) \tag{14}$$

where $k = 1, 2, \ldots n$, n is a predetermined number of previously undefined obstacles that are clustered by the DBSCAN algorithm after collecting the LiDAR sensor information. $xobs_k$, $yobs_k$ are the coordinates of the nearest point in the obstacle surface to the robot, R_b is the radius of the robot that indicate its influence on the map, and β is the safety margin that can be added to enhance the safe navigation of the motion system when inside the non-obstructed space.

It is important to note that no prior information of the number, size, or geographic distribution of obstacles is necessary. Nevertheless, certain online information about obstacles is anticipated using measurement or estimation for control reasons.

Finding the most suitable terminal penalty and constraints is made possible by the stability theory provided in [22] as following:

- A continuous cost function is assumed with $\ell(0,0) = 0$ and $\ell(\mathbf{x}_e, \mathbf{u}_e) > 0$.
- Consider that the reference control signals are bounded, i.e., $[v_{xr} \quad v_{yr} \quad \omega_r]^T < [v_{xrmax} \quad v_{yrmax} \quad \omega_{rmax}]^T$, and at time $t = 0$, the open-loop optimization problem retains a definite solution.
- $P(\cdot)$ is presumed to be differential and continuous function, and fulfilling $P(\mathbf{0}) = 0$, and $P(\mathbf{x}_e) > 0$ for all $\mathbf{x}_e \neq 0$.
- A terminal-state controller $\mathbf{u}_e^{\mathrm{L}}$ exists such that the following condition is satisfied:

$$\dot{P}(\mathbf{x}_e) + \ell(\mathbf{x}_e, \mathbf{u}_e) \leq 0, \forall\, \mathbf{x}_e \in \Omega \tag{15}$$

where Ω represents the terminal-state region. Then, the closed-loop system is guaranteed to be asymptotically stable using the NMPC approach previously stated [22].

IV. Simulation and Results

The simulation for the OWMR dynamics and the environment is implemented inside the CoppeliaSim robotic simulator. CoppeliaSim is a very powerful robotic simulator that supports rigid and soft bodies dynamics simulation. Moreover, it contains numerus built-in robot models, tools, sensors, and actuators that can be utilized to build a virtual environment with ability for real-time interaction. CoppeliaSim can be programmed with various programming languages such as Lua, MATLAB, C++, and Python which makes it ideal choice for robot simulations.

In this simulation setup, it is assumed that OWMR knows its pose (x, y, θ) at each simulation step. Additionally, it is equipped with a laser rangefinder with a range of 3m and 360° field of view to detect the obstacles around the OWMR.

MATLAB is used for laser rangefinder data processing and NMPC implementation. The connection between

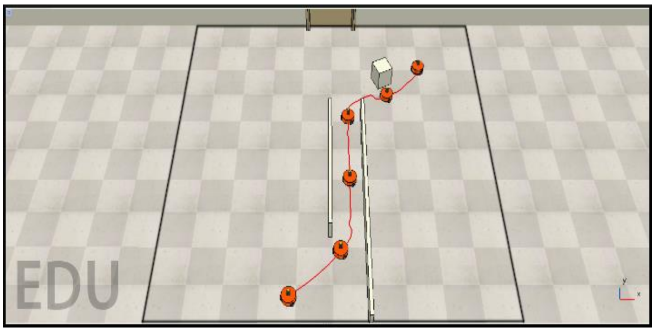

Fig. 2 OWMR performed trajectory for the first scenario.

CoppielliaSim and MATLAB is done using remote APIs. MATLAB receives raw data from the laser range finder and

uses the DBSCAN algorithm to cluster the data points that belong to each obstacle together. The output of the DBSCAN algorithm are sets of points where each one represents an obstacle in the scene. The minimum distance between each set of points and the center of the OWMR is then calculated. Furthermore, the CasADi toolbox inside MATLAB is used to perform both the system states integration, and optimal control computation. The NMPC is implemented in the CasADi toolbox, where both the system model (7), and the optimal control problem are defined. By using the multiple shooting method and IPOPT solver for nonlinear programming problems, the optimum control issue is resolved, and states integration is accomplished. The convergence criterion of IPOPT is kept at 10^{-8} and the maximum number of iterations was set to 2000. More information about CasADi toolbox for MATLAB is available in [52]. Runge-Kutta 4th (RK4) order method is employed for states integration. On a core i7 personal computer with 16 GB RAM and a 2.59 GHz CPU, all simulations were run.

The controller saturation limitations and the map margins have been chosen as follows:

$$\begin{aligned} -0.5 &\leq v_x \leq 0.5\ (m/s) \\ -0.5 &\leq v_y \leq 0.5\ (m/s) \\ -0.8 &\leq \omega \leq 0.8\ (rad/s) \\ -2.5 &\leq x \leq 2.5\ (m) \\ -2.5 &\leq y \leq 2.5\ (m) \end{aligned}$$

The radius of the robot in (14) is chosen as $R_b = 0.1\ (m)$, and robot wheels radii in (5) is the are selected as $r = 0.02\ (m)$. The weight matrices appear in (10 & 11) are selected as:

$$Q = F = \begin{bmatrix} 10 & 0 & 0 \\ 0 & 10 & 0 \\ 0 & 0 & 10 \end{bmatrix}, \text{and } R = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}$$

The prediction horizon parameters are determined as follows: the total number of horizon steps N = 5 (steps) and the time step $\Delta T = 0.2$ (seconds), resulting in a prediction horizon length of T = 1 (second).

To show the performance of the proposed collision avoidance scheme, two simulation scenarios are considered. In both scenarios, the robot is required to go to the target pose of $\mathbf{x}_r = (1.5m, 1.5m, 0°)$, and the map margins are highlighted with a black edged box.

The first scenario is shown in Fig. 2 where there are only static obstacles in the scene, the robot starts from the initial configuration specified by $\mathbf{x} = (-0.6m, -2m, 0°)$. As can be seen from Fig. 2, there is a narrow passage of 0.475m width. The OWMR successfully passes through the passage and then encounters a cuboid-shaped obstacle when exiting the passage which it bypasses and reaches the goal pose.

Fig. 3 illustrates the state vector $\mathbf{x}$ components of the OWMR for static obstacles avoidance scenario under the proposed controller. Along the simulation time, the proposed controller exhibits a smooth transition of the robot's states. Moreover, the states demonstrate a rapid convergence to their reference values. Fig. 4 shows the control actions applied to the OWMR for not moving obstacles avoidance case. As can be deduced from this figure, the proposed controller shows a smooth control action, and did not exceed their saturation limits.

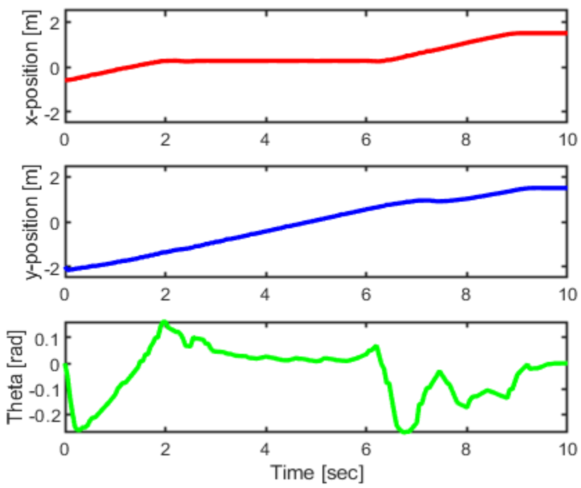

Fig. 3 OWMR state vector $\mathbf{x}$ components for the first scenario.

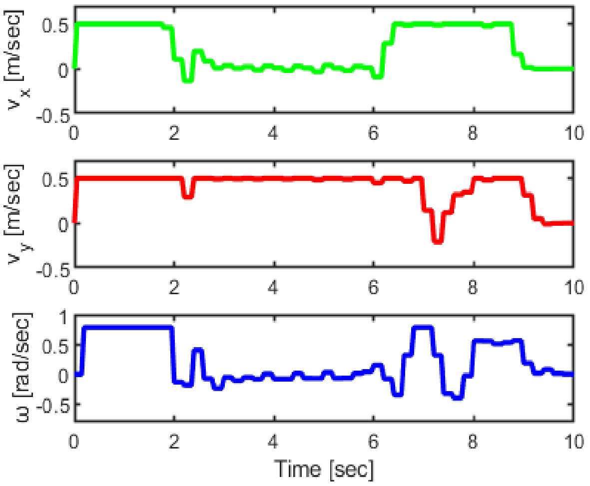

Fig. 4 OWMR controls vector $\mathbf{u}$ components for the first scenario.

The second scenario is more challenging and includes moving obstacles in addition to the static ones as presented in Fig. 5. In this scenario, the robot starts from the initial configuration specified by $\mathbf{x} = (-0.6m, -1.7m, 0°)$, and there are three moving obstacles which are colored in cyan, red, and blue. There are two static obstacles colored in pink close to the target position, green disk. As shown in Fig. 5, the OWMR encounters the cyan obstacle at first and avoids it successfully. After that, it faces the red cuboid and bypasses

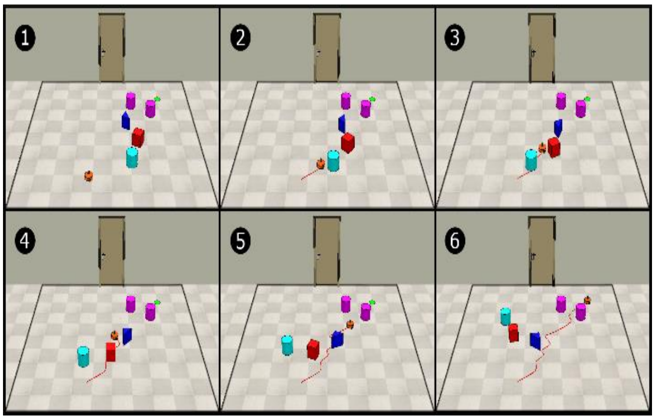

Fig. 5 OWMR performed trajectory for the second scenario.

it, and then avoids the blue obstacle. Finally, it passes between the static obstacles and reaches the target pose. To ensure dynamic obstacle avoidance, the following condition is imposed: the speed of the obstacle should be less than the speed of the robot.

Fig 6 shows the OWMR state vector $[x \quad y \quad \theta]^T$ components under the proposed controller for moving obstacles avoidance case. The proposed controller shows a smooth change in the robot's states over the course of the simulation. Additionally, the states demonstrate a quick convergence to their reference values. Fig. 7 depicts the OWMR's control action for dynamic obstacles avoidance scenario. As can be seen from the figure, the proposed controller did not exceed its saturation limitations and exhibited a smooth control action.

V. CONCLUSION

In this paper, a NMPC framework was used to stabilize OWMR to a specific target while avoiding obstacles in a cluttered environment. The DBSCAN algorithm was implemented for clustering the dynamic and static obstacles after collecting the information from the LiDAR sensor. The environmental obstacles were considered as further constraints, which are denoted in this work as having minimum distance forms. A simulation was performed using MATLAB that utilizes the CasADi Toolbox with IPOPT solver, in conjunction with the CoppeliaSim robot simulator. Two scenarios were considered based on whether the obstacle was stationary or moving. During numerical simulations of robot stabilization in environments with static and dynamic obstacles, the proposed controller algorithm has demonstrated superior performance in real time.

The future work of this research includes the practical implementation of the proposed control framework on a real OWMR in addition to the stability analysis of the proposed controller.

REFERENCES

[1] G. Li et al., "Hybrid Maps Enhanced Localization System for Mobile Manipulator in Harsh Manufacturing Workshop," IEEE Access, vol. 8, pp. 10782–10795, 2020, doi: 10.1109/access.2020.2965300.

[2] P. Tokekar, J. Vander Hook, D. Mulla, and V. Isler, "Sensor planning for a symbiotic UAV and UGV system for precision agriculture," IEEE transactions on robotics, vol. 32, no. 6, pp. 1498-1511, 2016, https://doi.org/10.1109/iros.2013.6697126.

[3] R. Bonatti, Y. Zhang, S. Choudhury, W. Wang, and S. Scherer, "Autonomous Drone Cinematographer: Using Artistic Principles to

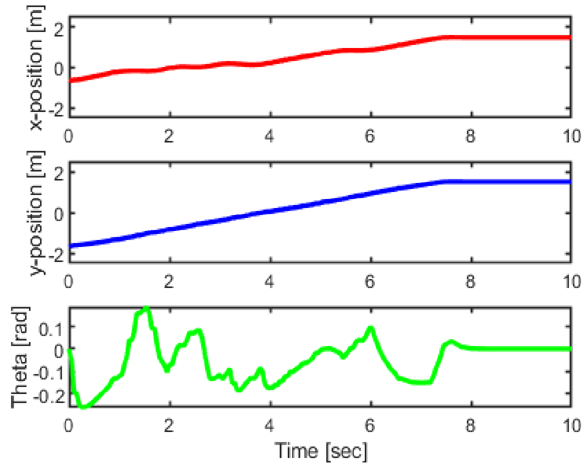

Fig. 6 OWMR state vector **x** components for the second scenario

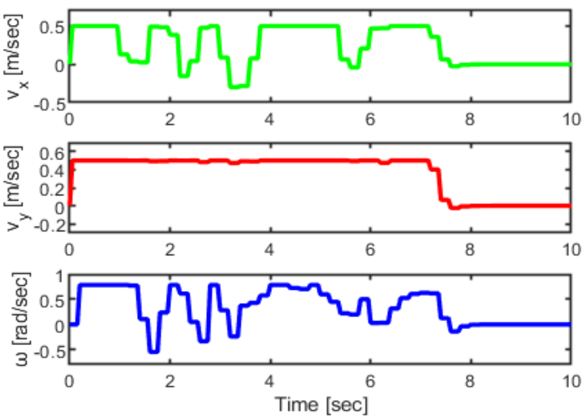

Fig. 7 OWMR controls vector **u** components for the second scenario.

Create Smooth, Safe, Occlusion-Free Trajectories for Aerial Filming," Proceedings of the 2018 International Symposium on Experimental Robotics, pp. 119–129, 2020, doi: 10.1007/978-3-030-33950-0_11.

[4] S. S. Mansouri, C. Kanellakis, D. Kominiak, and G. Nikolakopoulos, "Deploying MAVs for autonomous navigation in dark underground mine environments," Robotics and Autonomous Systems, vol. 126, p. 103472, Apr. 2020, doi: 10.1016/j.robot.2020.103472.

[5] A. Sahoo, S. K. Dwivedy, and P. S. Robi, "Advancements in the field of autonomous underwater vehicle," Ocean Engineering, vol. 181, pp. 145–160, Jun. 2019, doi: 10.1016/j.oceaneng.2019.04.011.

[6] R. C. Cardoso et al., "A Review of Verification and Validation for Space Autonomous Systems," Current Robotics Reports, vol. 2, no. 3, pp. 273–283, Jun. 2021, doi: 10.1007/s43154-021-00058-1.

[7] M. Z. Ab Rashid, M. F. Mohd Yakub, S. A. Zaki bin Shaikh Salim, N. Mamat, S. M. Syed Mohd Putra, and S. A. Roslan, "Modeling of the in-pipe inspection robot: A comprehensive review," Ocean Engineering, vol. 203, p. 107206, May 2020, doi: 10.1016/j.oceaneng.2020.107206.

[8] W. E. Dixon, D. M. Dawson, E. Zergeroglu, and A. Beha, "Nonlinear Control of Wheeled Mobile Robots," Lecture Notes in Control and Information Sciences, 2001, doi: 10.1007/bfb0113116.

[9] O. Y. Ismael, M. Qasim, M. N. Noaman, and A. Kurniawan, "Salp Swarm Algorithm-Based Nonlinear Robust Control of Magnetic Levitation System Using Feedback Linearization Approach," Proceedings of the 3rd International Conference on Electronics, Communications and Control Engineering, Apr. 2020, doi: 10.1145/3396730.3396734.

[10] M. N. Alghanim, M. Qasim, K. P. Valavanis, M. J. Rutherford, and M. Stefanovic, "Comparison of Controller Performance for UGV-Landing Platform Self-Leveling," 2020 28th Mediterranean Conference on Control and Automation (MED), Sep. 2020, doi: 10.1109/med48518.2020.9182837.

[11] O. Y. Ismael, M. Qasim, and M. N. Noaman, "Equilibrium Optimizer-Based Robust Sliding Mode Control of Magnetic Levitation System," Journal Européen des Systèmes Automatisés, vol. 54, no. 1, pp. 131–138, Feb. 2021, doi: 10.18280/jesa.540115.

[12] M. N. Alghanim, M. Qasim, K. P. Valavanis, M. J. Rutherford, and M. Stefanovic, "Passivity-Based Adaptive Controller for Dynamic Self-Leveling of a Custom-Built Landing Platform on Top of a UGV," 2020 28th Mediterranean Conference on Control and Automation (MED), Sep. 2020, doi: 10.1109/med48518.2020.9182807.

[13] Mohd. N. Zafar and J. C. Mohanta, "Methodology for Path Planning and Optimization of Mobile Robots: A Review," Procedia Computer Science, vol. 133, pp. 141–152, 2018, doi: 10.1016/j.procs.2018.07.018.

[14] S. A. Bonab and A. Emadi, "Optimization-based Path Planning for an Autonomous Vehicle in a Racing Track," IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society, Oct. 2019, doi: 10.1109/iecon.2019.8926856.

[15] N. D. Potdar, G. C. H. E. de Croon, and J. Alonso-Mora, "Online trajectory planning and control of a MAV payload system in dynamic environments," Autonomous Robots, vol. 44, no. 6, pp. 1065–1089, Jun. 2020, doi: 10.1007/s10514-020-09919-8.

[16] M. Hoy, A. S. Matveev, and A. V. Savkin, "Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey," Robotica, vol. 33, no. 3, pp. 463–497, Mar. 2014, doi: 10.1017/s0263574714000289.

[17] D. Kloeser, T. Schoels, T. Sartor, A. Zanelli, G. Prison, and M. Diehl, "NMPC for Racing Using a Singularity-Free Path-Parametric Model with Obstacle Avoidance," IFAC-PapersOnLine, vol. 53, no. 2, pp. 14324–14329, 2020, doi: 10.1016/j.ifacol.2020.12.1376.

[18] Y. Liu et al., "Robust nonlinear control approach to nontrivial maneuvers and obstacle avoidance for quadrotor UAV under disturbances," Robotics and Autonomous Systems, vol. 98, pp. 317–332, Dec. 2017, doi: 10.1016/j.robot.2017.08.011.

[19] G. Klančar and I. Škrjanc, "Tracking-error model-based predictive control for mobile robots in real time," Robotics and Autonomous Systems, vol. 55, no. 6, pp. 460–469, Jun. 2007, doi: 10.1016/j.robot.2007.01.002.

[20] G. V. Raffo, G. K. Gomes, J. E. Normey-Rico, C. R. Kelber, and L. B. Becker, "A Predictive Controller for Autonomous Vehicle Path Tracking," IEEE Transactions on Intelligent Transportation Systems, vol. 10, no. 1, pp. 92–102, Mar. 2009, doi: 10.1109/tits.2008.2011697.

[21] J. Backman, T. Oksanen, and A. Visala, "Navigation system for agricultural machines: Nonlinear Model Predictive path tracking," Computers and Electronics in Agriculture, vol. 82, pp. 32–43, Mar. 2012, doi: 10.1016/j.compag.2011.12.009.

[22] D. Gu and H. Hu, "A stabilizing receding horizon regulator for nonholonomic mobile robots," IEEE Transactions on Robotics, vol. 21, no. 5, pp. 1022–1028, Oct. 2005, doi: 10.1109/tro.2005.851357.

[23] M. W. Mehrez, K. Worthmann, J. P. V. Cenerini, M. Osman, W. W. Melek, and S. Jeon, "Model Predictive Control without terminal constraints or costs for holonomic mobile robots," Robotics and Autonomous Systems, vol. 127, p. 103468, May 2020, doi: 10.1016/j.robot.2020.103468.

[24] D. Gu and H. Hu, "Receding horizon tracking control of wheeled mobile robots," IEEE Transactions on Control Systems Technology, vol. 14, no. 4, pp. 743–749, Jul. 2006, doi: 10.1109/tcst.2006.872512.

[25] I. Sánchez, A. D'Jorge, G. V. Raffo, A. H. González, and A. Ferramosca, "Nonlinear Model Predictive Path Following Controller with Obstacle Avoidance," Journal of Intelligent & Robotic Systems, vol. 102, no. 1, Apr. 2021, doi: 10.1007/s10846-021-01373-7.

[26] F. Xie and R. Fierro, "First-state contractive model predictive control of nonholonomic mobile robots," 2008 American Control Conference, Jun. 2008, doi: 10.1109/acc.2008.4587034.

[27] T. Ding, Y. Zhang, G. Ma, Z. Cao, X. Zhao, and B. Tao, "Trajectory tracking of redundantly actuated mobile robot by MPC velocity control under steering strategy constraint," Mechatronics, vol. 84, p. 102779, Jun. 2022, doi: 10.1016/j.mechatronics.2022.102779.

[28] D. Wang, W. Wei, Y. Yeboah, Y. Li, and Y. Gao, "A Robust Model Predictive Control Strategy for Trajectory Tracking of Omni-directional Mobile Robots," Journal of Intelligent & Robotic Systems, vol. 98, no. 2, pp. 439–453, Dec. 2019, doi: 10.1007/s10846-019-01083-1.

[29] G. C. Karras and G. K. Fourlas, “Model Predictive Fault Tolerant Control for Omni-directional Mobile Robots,” Journal of Intelligent & Robotic Systems, vol. 97, no. 3–4, pp. 635–655, May 2019, doi: 10.1007/s10846-019-01029-7.

[30] T. P. Nascimento, C. E. T. Dórea, and L. M. G. Gonçalves, “Nonlinear model predictive control for trajectory tracking of nonholonomic mobile robots,” International Journal of Advanced Robotic Systems, vol. 15, no. 1, p. 172988141876046, Jan. 2018, doi: 10.1177/1729881418760461.

[31] X. Zhu, C. Ding, L. Jia, and Y. Feng, “Koopman operator based model predictive control for trajectory tracking of an omnidirectional mobile manipulator,” Measurement and Control, vol. 55, no. 9–10, pp. 1067–1077, Aug. 2022, doi: 10.1177/00202940221095559.

[32] J. B. Rawlings and K. R. Muske, “The stability of constrained receding horizon control,” IEEE Transactions on Automatic Control, vol. 38, no. 10, pp. 1512–1516, 1993, doi: 10.1109/9.241565.

[33] W. Esterhuizen, K. Worthmann, and S. Streif, “Recursive Feasibility of Continuous-Time Model Predictive Control Without Stabilising Constraints,” IEEE Control Systems Letters, vol. 5, no. 1, pp. 265–270, Jan. 2021, doi: 10.1109/lcsys.2020.3001514.

[34] L. Grüne and J. Pannek, “Nonlinear Model Predictive Control,” Communications and Control Engineering, 2017, doi: 10.1007/978-3-319-46024-6.

[35] S. A. Emami and A. Banazadeh, “Simultaneous trajectory tracking and aerial manipulation using a multi-stage model predictive control,” Aerospace Science and Technology, vol. 112, p. 106573, May 2021, doi: 10.1016/j.ast.2021.106573.

[36] J. Köhler, M. A. Müller, and F. Allgöwer, “A nonlinear tracking model predictive control scheme for dynamic target signals,” Automatica, vol. 118, p. 109030, Aug. 2020, doi: 10.1016/j.automatica.2020.109030.

[37] Z. Wang, J. Zhan, C. Duan, X. Guan, P. Lu and K. Yang, "A Review of Vehicle Detection Techniques for Intelligent Vehicles," in IEEE Transactions on Neural Networks and Learning Systems, vol. 34, no. 8, pp. 3811-3831, Aug. 2023, doi: 10.1109/TNNLS.2021.3128968.

[38] O. H. Jafari, D. Mitzel, and B. Leibe, “Real-time RGB-D based people detection and tracking for mobile robots and head-worn cameras,” 2014 IEEE International Conference on Robotics and Automation (ICRA), May 2014, doi: 10.1109/icra.2014.6907688.

[39] T. Eppenberger, G. Cesari, M. Dymczyk, R. Siegwart, and R. Dube, “Leveraging Stereo-Camera Data for Real-Time Dynamic Obstacle Detection and Tracking,” 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Oct. 2020, doi: 10.1109/iros45743.2020.9340699.

[40] M. Qasim and O. Y. Ismael, “Shared Control of a Robot Arm Using BCI and Computer Vision,” Journal Européen des Systèmes Automatisés, vol. 55, no. 1, pp. 139–146, Feb. 2022, doi: 10.18280/jesa.550115.

[41] M. N. Noaman, M. Qasim, and O. Y. Ismael, "Landmarks exploration algorithm for mobile robot indoor localization using VISION sensor," Journal of Engineering Science & Technology, vol. 16, no. 4, pp. 3165-3184, 2021.

[42] J. Lin, H. Zhu, and J. Alonso-Mora, “Robust Vision-based Obstacle Avoidance for Micro Aerial Vehicles in Dynamic Environments,” 2020 IEEE International Conference on Robotics and Automation (ICRA), May 2020, doi: 10.1109/icra40945.2020.9197481.

[43] B. Brito, B. Floor, L. Ferranti, and J. Alonso-Mora, “Model Predictive Contouring Control for Collision Avoidance in Unstructured Dynamic Environments,” IEEE Robotics and Automation Letters, vol. 4, no. 4, pp. 4459–4466, Oct. 2019, doi: 10.1109/lra.2019.2929976.

[44] M. Parimala, D. Lopez, and N. Senthilkumar, "A survey on density based clustering algorithms for mining large spatial databases," International Journal of Advanced Science and Technology, vol. 31, no. 1, pp. 59-66, 2011.

[45] O. Khatib, “Real-Time Obstacle Avoidance for Manipulators and Mobile Robots,” Autonomous Robot Vehicles, pp. 396–404, 1986, doi: 10.1007/978-1-4613-8997-2_29.

[46] F. Gao, W. Wu, Y. Lin, and S. Shen, “Online Safe Trajectory Generation for Quadrotors Using Fast Marching Method and Bernstein Basis Polynomial,” 2018 IEEE International Conference on Robotics and Automation (ICRA), May 2018, doi: 10.1109/icra.2018.8462878.

[47] D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” IEEE Robotics & Automation Magazine, vol. 4, no. 1, pp. 23–33, Mar. 1997, doi: 10.1109/100.580977.

[48] F. Gao, Y. Lin, and S. Shen, “Gradient-based online safe trajectory generation for quadrotor flight in complex environments,” 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sep. 2017, doi: 10.1109/iros.2017.8206214.

[49] G. Ferrer, A. Garrell and A. Sanfeliu, "Robot companion: A social-force based approach with human awareness-navigation in crowded environments," 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, Tokyo, Japan, 2013, pp. 1688-1694, doi: 10.1109/IROS.2013.6696576.

[50] B. T. Lopez and J. P. How, “Aggressive 3-D collision avoidance for high-speed navigation,” 2017 IEEE International Conference on Robotics and Automation (ICRA), May 2017, doi: 10.1109/icra.2017.7989677.

[51] A. Majumdar and R. Tedrake, “Funnel libraries for real-time robust feedback motion planning,” The International Journal of Robotics Research, vol. 36, no. 8, pp. 947–982, Jun. 2017, doi: 10.1177/0278364917712421.

[52] L. Hewing, K. P. Wabersich, M. Menner, and M. N. Zeilinger, “Learning-Based Model Predictive Control: Toward Safe Learning in Control,” Annual Review of Control, Robotics, and Autonomous Systems, vol. 3, no. 1, pp. 269–296, May 2020, doi: 10.1146/annurev-control-090419-075625.

[53] J. Berberich, J. Köhler, M. A. Müller and F. Allgöwer, "Data-Driven Model Predictive Control With Stability and Robustness Guarantees," in IEEE Transactions on Automatic Control, vol. 66, no. 4, pp. 1702-1717, April 2021, doi: 10.1109/TAC.2020.3000182.

[54] R. Carli, G. Cavone, N. Epicoco, P. Scarabaggio, and M. Dotoli, "Model predictive control to mitigate the COVID-19 outbreak in a multi-region scenario," Annual Reviews in Control, vol. 50, pp. 373-393, 2020, https://doi.org/10.1016/j.arcontrol.2020.09.005.

[55] J. Zeng, B. Zhang, and K. Sreenath, “Safety-Critical Model Predictive Control with Discrete-Time Control Barrier Function,” 2021 American Control Conference (ACC), May 2021, doi: 10.23919/acc50511.2021.9483029.

[56] Z. Jian et al., “Dynamic Control Barrier Function-based Model Predictive Control to Safety-Critical Obstacle-Avoidance of Mobile Robot,” 2023 IEEE International Conference on Robotics and Automation (ICRA), May 2023, doi: 10.1109/icra48891.2023.10160857.

[57] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADi: a software framework for nonlinear optimization and optimal control,” Mathematical Programming Computation, vol. 11, no. 1, pp. 1–36, Jul. 2018, doi: 10.1007/s12532-018-0139-4.

[58] A. Wächter and L. T. Biegler, “Line Search Filter Methods for Nonlinear Programming: Motivation and Global Convergence,” SIAM Journal on Optimization, vol. 16, no. 1, pp. 1–31, Jan. 2005, doi: 10.1137/s1052623403426556.

[59] J. C. L. Barreto S., A. G. S. Conceicao, C. E. T. Dorea, L. Martinez, and E. R. de Pieri, “Design and Implementation of Model-Predictive Control With Friction Compensation on an Omnidirectional Mobile Robot,” IEEE/ASME Transactions on Mechatronics, vol. 19, no. 2, pp. 467–476, Apr. 2014, doi: 10.1109/tmech.2013.2243161.

[60] W. B. Dunbar and R. M. Murray, “Model predictive control of coordinated multi-vehicle formations,” Proceedings of the 41st IEEE Conference on Decision and Control, 2002., doi: 10.1109/cdc.2002.1185108. Sf.