

## Article

# Motion Planning and Control of an Omnidirectional Mobile Robot in Dynamic Environments

Mahmood Reza Azizi <sup>1</sup>, Alireza Rastegarpanah <sup>1,2,\*</sup> and Rustam Stolkin <sup>1,2</sup>

<sup>1</sup> Department of Metallurgy and Materials Science, University of Birmingham, Edgbaston, Birmingham B15 2TT, UK; mhr\_azizi@yahoo.com (M.R.A.); r.stolkin@bham.ac.uk (R.S.)

<sup>2</sup> The Faraday Institution, Quad One, Harwell Science and Innovation Campus, Didcot OX11 0RA, UK

\* Correspondence: a.rastegarpanah@bham.ac.uk

**Abstract:** Motion control in dynamic environments is one of the most important problems in using mobile robots in collaboration with humans and other robots. In this paper, the motion control of a four-Mecanum-wheeled omnidirectional mobile robot (OMR) in dynamic environments is studied. The robot's differential equations of motion are extracted using Kane's method and converted to discrete state space form. A nonlinear model predictive control (NMPC) strategy is designed based on the derived mathematical model to stabilize the robot in desired positions and orientations. As a main contribution of this work, the velocity obstacles (VO) approach is reformulated to be introduced in the NMPC system to avoid the robot from collision with moving and fixed obstacles online. Considering the robot's physical restrictions, the parameters and functions used in the designed control system and collision avoidance strategy are determined through stability and performance analysis and some criteria are established for calculating the best values of these parameters. The effectiveness of the proposed controller and collision avoidance strategy is evaluated through a series of computer simulations. The simulation results show that the proposed strategy is efficient in stabilizing the robot in the desired configuration and in avoiding collision with obstacles, even in narrow spaces and with complicated arrangements of obstacles.

**Citation:** Azizi, M.R.; Rastegarpanah, A.; Stolkin, R. Motion Planning and Control of Omni Directional Mobile Robot in Dynamic Environments. *Robotics* **2021**, *10*, 48. <https://doi.org/10.3390/robotics10010048>

Received: 26 January 2021

Accepted: 9 March 2021

Published: 17 March 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).

**Keywords:** omnidirectional mobile robot; Kane's dynamics; model predictive control; velocity obstacles; obstacle avoidance

## 1. Introduction

With the advancement of technology and the invention of new wheels with special structures, omnidirectional mobile robots (OMRs) have been presented and widely used recently in industrial applications, space exploration, medical fields etc. [1–3]. In spite of conventional mobile robots that use two driver wheels to control the robot motion on restricted directions, OMRs are not limited by nonholonomic constraints and can move in any direction from every configuration [4]. This characteristic increases the maneuverability, swiftness and accuracy of the mobile robot, which makes the robot capable of working in small workspaces and dynamic environments with high precision [5].

The dynamics and control of OMRs with different types and numbers of omni wheels have attracted several researchers recently due to their mobility and special maneuverability. Therefore, the kinematics, dynamics and motion optimization of OMRs have been studied extensively in the previous studies. The kinematic and dynamic modeling of a four wheels OMR have been investigated using Lagrange formulation in [4–6]. Kim et al. [7] have investigated minimum energy trajectory generation for a three wheels OMR based on the simplified dynamic model of the robot, and designed an optimal controller to control the motion of the robot on a minimum energy trajectory with a fixed heading. Considering the effects of disturbances and friction forces on the wheels, the adaptive sliding mode controller has been designed to control a mobile robot with four poly wheels

robustly on desired trajectories [8]. A model predictive control and filtered smith predictor have been designed based on the kinematic model of a three wheels mobile robot to control its motion on specific trajectories [9]. Huang et al. has investigated the challenges of path tracking of an OMR in [10]. The robot's dynamic model and computed-torque-like-controller approach have been used to address these challenges. The slippage of the wheels and the friction forces have also been considered in the robot's model.

Although several types of omni-wheels such as Mecanum wheels [11] and orthogonal wheels have been presented in the previous studies, the Mecanum wheels showed better performance by maximum omnidirectional efficiency and robust omnidirectional maneuverability [4]. A Mecanum wheel is composed of small rollers installed in a certain angle around a circular hub. Using this construction provides two degrees of freedom for the wheel to move in two perpendicular directions and increases the maneuverability of the OMRs. Therefore, the modeling and control problems of OMRs with Mecanum wheels have been the subject of many studies in recent years.

Zimmermann et al. have derived the dynamic equation of an OMR with four Mecanum wheels based on the nonholonomic model of the wheel and shown that the nonholonomic constraint is integrable merely in pure translational and pure rotational motion [12]. Xie et al. used a dynamic window approach to minimize the consumed energy and consumed power of an OMR with four wheels based on its kinematic model and dynamic model of its motors [13]. Huang et al. proposed a robust second order sliding mode controller designed based on the dynamic model of Mecanum-wheeled mobile robot and adaptive laws have been defined to estimate the switching gains adaptively [10]. An extended state observer has been developed to estimate the bounds of uncertainties and unmeasured states, and sliding mode controller has been designed using the estimated parameters to improve the tracking performance of a four Mecanum-wheeled mobile robot [14]. The sliding mode control has also been utilized by other researchers to develop robust control systems for OMRs with Mecanum wheels in the presence of unmodeled dynamics and external disturbances [15–17]. Conceica et al. proposed a nonlinear model predictive control (NMPC) which was designed for trajectory tracking control of a Mecanum wheeled OMR based on simplified dynamic model of the robot [18]. Han et al. have developed a dynamic model of an OMR with Mecanum wheels and designed a robust linear model predictive control (MPC) based on the robot's linearized dynamic model [19]. Using the kinematic model of an OMR with Mecanum wheel and considering the robot's inputs limitations, an MPC system has been also developed in [20] for trajectory tracking and a delayed neural network has been proposed to solve the resultant optimization problem of MPC.

In addition to trajectory tracking and control of mobile robots, obstacle avoidance and motion planning in dynamic environments are two major problems with the mobile robot applications which is widely investigated in previous studies [21–23]. Whilst one of the main advantages of OMRs is their capability to work in collaboration with humans and other robots in dynamic environments, the problem of online motion planning and control of OMRs in such situations has been addressed by only a few researchers [24–26]. Zavlangas et al. [24] has used a fuzzy logic controller to plan the robot's motion in the presence of obstacles. However, the output of the controller is the acceleration of the robot and its dynamic behavior as well as its physical characteristics, and these limitations have not been considered in the proposed method. In addition, stabilization of the robot in the desired position and orientation has not been studied as a part of the proposed control system. Leng et al. developed a motion planning strategy for a four wheels OMR based on the artificial potential field (APF) method and developed a strategy to reduce the robot's speed on its trajectory to avoid collision with moving obstacles [25]. In this study, the robot has been restricted to move on a predefined trajectory determined to minimize the consumed energy of the robot. In addition, collision avoidance has been achieved by reducing the speed of the robot in the vicinity of moving obstacles. Consequently, the robot maneuverability and the performance of the proposed algorithm decreases in

crowded areas and especially in local minima points of APF. Jianhua et al. developed a collision avoidance strategy for a three wheels OMR by changing the speed of the robot in the perpendicular direction to the desired path line [26]. In this method the robot moves on the linear trajectory between the start and the target points and collision avoidance has been performed by changing the robots speed in the perpendicular direction to this line. However, this strategy has some restriction and may be failed in complicated environments and in the presence of multiple moving obstacles.

The motivation of this work is to develop an online motion planning and obstacle avoidance strategy for OMR as a part of its feedback control system to stabilize the robot in desired position and orientation in the presence of static and moving obstacles. This approach makes it possible to control the state variables and to plan the robot's trajectory between obstacles simultaneously based on the robot's dynamic model and its physical constraints and limitation. It has been extensively investigated in several recent studies and shown that the MPC approach has this capability to take the complicated geometrical and dynamical constraints of the system under control [27–30]. Owing to special properties of MPC technique, obstacle avoidance strategies such as the velocity obstacles (VO) method [31] can be combined with the MPC system and it can be implemented to plan the robots motion online. As the control inputs are calculated based on the future predicted behavior of the system in the MPC technique, such a combination could also lead to a more efficient motion planning algorithm in complicated obstacle arrangements and crowded workspaces.

Therefore, in this paper, a robust control system is designed using the NMPC technique; the proposed control system is designed based on the dynamic model of an OMR with four Mecanum wheels to stabilize the robot in the desired position and orientation in dynamic environments. After kinematic analysis, the dynamic model of the robot is extracted using Kane's method, considering the Mecanum wheels and rollers dynamics. As a main contribution of this study, a velocity obstacles (VO) strategy is formulated as some optimization constraints to combine with the NMPC control system and used to move the robot towards the target point among obstacles. To investigate the effectiveness of the control system and collision avoidance strategy, some design parameters are defined, and their values are specified through stability and performance analysis. Finally, the control system and the proposed obstacle avoidance strategy are assessed through simulation experiments that the results verify their effectiveness.

The rest of the paper is organized as follows: In Section 2, the kinematic and dynamic models of the robot are extracted. The control system design and development of obstacle avoidance strategy are presented in Section 3 based on the NMPC approach. In Section 4, some simulation experiments are presented to verify the effectiveness of the proposed controller. Finally, the paper is finished with conclusion in Section 5.

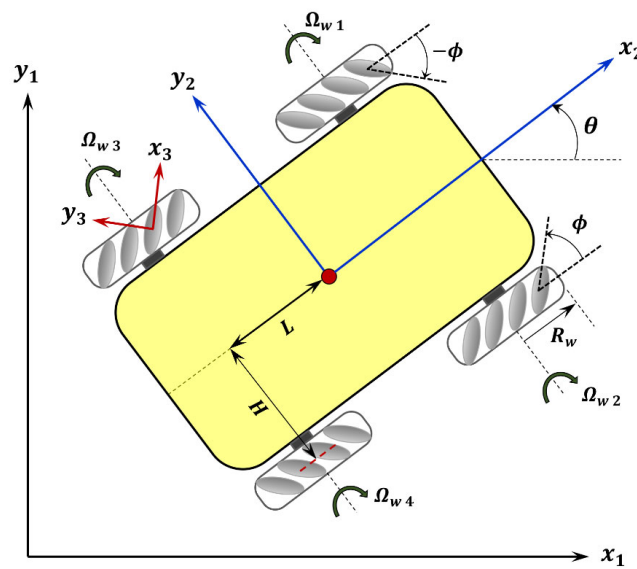
## 2. Mathematical Modeling

To develop a control system and collision avoidance strategy, a mathematical model of the robot should be extracted. Therefore, in this section the kinematic and dynamic modeling of the robot are performed and consequently the differential equations of motion of the robot are derived.

### 2.1. Kinematic Modeling

The schematic model of the OMR with four Mecanum wheels is illustrated in Figure 1. Compared with traditional nonholonomic wheeled mobile robots, the Mecanum wheels provide an additional degree of freedom for the robot to move also in a lateral direction. To determine the robot's position, velocity and acceleration, three coordinate frames are defined as shown in Figure 1. Coordinate frame {1} is the inertial reference frame. Coordinate frame {2} is the body coordinate attached to the robot's center of mass and rotates with it around the  $z$  axis. Coordinate frame  $\{3\}_i; i = 1..4$  is attached to the center of

rollers which are in contact with the ground in which its x axis coincides with the roller's axis of rotation and its z axis is parallel to the z axis of frame {2}. This coordinate frame rotates with the associated wheel; therefore, its angular velocity is equal to the wheel's angular velocity and the roller rotates around its x axis by relative angular speed  $\Omega_{r_i}$ . The position and orientation of the robot in frame {1} could be defined by three generalized coordinates  $\mathbf{q} = (x, y, \theta)$  as shown in Figure 1. As shown in Figure 1, the angles of rollers with respect to x axis of frame {2} are  $\mathbf{\Gamma} = \left[ \frac{\pi}{2} + \phi \quad \phi \quad \phi \quad \frac{\pi}{2} + \phi \right]$  respectively. The coordinate transformation matrix between frame {2} and {1} and frame {3} and {2} are as follows.



**Figure 1.** The schematic model of omnidirectional mobile robot (OMR) and the attached frames.

$${}^1_2\mathbf{R} = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}; {}^2_1\mathbf{R} = {}^1_2\mathbf{R}^T \quad (1)$$

$${}^2_3\mathbf{R}_i = \begin{bmatrix} \cos \Gamma_i & -\sin \Gamma_i & 0 \\ \sin \Gamma_i & \cos \Gamma_i & 0 \\ 0 & 0 & 1 \end{bmatrix}; i = 1..4 \quad (2)$$

To simplify the notation of the paper, the mobile platform (MP), the wheels and the rollers which are in contact with the ground are numbered by 1 to 9 respectively. Therefore, using time derivative of generalized coordinates, the linear and angular velocity of the MP is as follows:

$$\begin{aligned} {}^1\mathbf{V}_1 &= [\dot{x}, \dot{y}, 0]^T \\ {}^1\boldsymbol{\omega}_1 &= {}^2\boldsymbol{\omega}_1 = [0, 0, \dot{\theta}]^T \end{aligned} \quad (3)$$

$${}^2\mathbf{V}_1 = {}^2\mathbf{R}^1\mathbf{V}_1 \quad (4)$$

where the left superscript denotes the frame number that the vector is described in. The linear velocity of the center of each wheel in frame {2} is calculated in terms of its position with respect to the robot's center of gravity and the MP velocity.

$${}^2\mathbf{V}_i = {}^2\mathbf{V}_1 + {}^2\boldsymbol{\omega}_1 \times {}^2\mathbf{r}_{i/G}; i = 2 \dots 5 \quad (5)$$

where  ${}^2\mathbf{r}_{i/G}$  stands for the position vector of each wheel in frame {2}. It is assumed that each wheel can rotate independently with angular velocity  $\Omega_{wi}$ ;  $i = 1 \dots 4$  relative to MP. Therefore, the angular velocity of wheels is determined as follows.

$$\begin{aligned} {}^2\boldsymbol{\omega}_i &= [0 \quad \Omega_{wj} \quad \dot{\theta}]^T; i = 2 \dots 5, j = 1 \dots 4 \\ {}^1\boldsymbol{\omega}_i &= {}^1R^2 {}^2\boldsymbol{\omega}_i \end{aligned} \quad (6)$$

The rollers of the Mecanum wheels can also rotate freely with respect to the wheel's body by  $\Omega_{ri}$ ;  $i = 1 \dots 4$  and their angular velocity vectors could be written in the following form.

$$\begin{aligned} {}^3\boldsymbol{\omega}_i &= [\Omega_{rj} \quad 0 \quad \dot{\theta}]^T; i = 6 \dots 9, j = 1 \dots 4 \\ {}^2\boldsymbol{\omega}_i &= {}^2R^3 {}^3\boldsymbol{\omega}_i \end{aligned} \quad (7)$$

Using Equation (6) the linear velocity of the center of each roller is calculated as follows:

$${}^2\mathbf{V}_j = {}^2\mathbf{V}_i - [R_w \Omega_{wj} \quad 0 \quad 0]^T; i = 2 \dots 5, j = 6 \dots 9 \quad (8)$$

where  $R_w$  is the radius of wheels. It is assumed that the rollers of the Mecanum wheels is rolling without slipping on the ground. Therefore, the velocity of the contact point of each roller with the ground is zero and the linear velocity of rollers' center could also be determined using Equation (7).

$${}^2\mathbf{V}_j = {}^2\boldsymbol{\omega}_i \times [0 \quad 0 \quad R_r]^T; i = 6 \dots 9, j = 6 \dots 9 \quad (9)$$

where  $R_r$  is the radius of rollers. By equating Equation (8) and Equation (9) a system of eight algebraic equations is obtained that could be solved to calculate four wheels' angular speed with respect to MP ( $\Omega_{wi}$ ) and four roller's angular speed with respect to wheels' body ( $\Omega_{ri}$ ) as follows:

$$\boldsymbol{\Omega}_w = {}^1J_w \dot{\mathbf{q}} = {}^1J_w \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}; \quad \boldsymbol{\Omega}_r = {}^1J_r \dot{\mathbf{q}} = {}^1J_r \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} \quad (10)$$

where  $\boldsymbol{\Omega}_w$  and  $\boldsymbol{\Omega}_r$  ( $4 \times 1$ ) are vectors of angular speeds of the wheels and rollers respectively, and  $J_w$  is the Jacobian matrix between wheels angular speed and the MP velocity in frame {1} in the following form.

$$J_w = \frac{1}{R_w} \begin{bmatrix} \sin(\theta + \phi)/\sin \phi & -\cos(\theta + \phi)/\sin \phi & -(L \cos \phi + H \sin \phi)/\sin \phi \\ \cos(\theta + \phi)/\cos \phi & \sin(\theta - \phi)/\cos \phi & (H \cos \phi + L \sin \phi)/\cos \phi \\ \cos(\theta + \phi)/\cos \phi & \sin(\theta - \phi)/\cos \phi & -(H \cos \phi + L \sin \phi)/\cos \phi \\ \sin(\theta + \phi)/\sin \phi & -\cos(\theta + \phi)/\sin \phi & (L \cos \phi + H \sin \phi)/\sin \phi \end{bmatrix} \quad (11)$$

The Jacobian matrix  $J_r$  is also in a similar form and is not presented here due to the space limitation. In addition, these Jacobian matrixes can be transformed into frame {2} using coordinate transformation matrix from Equation (1).

It is obvious from Equation (10) and Equation (11) that the angular velocities of the wheels have the following relationship, which imposes a constraint originated from 3 DOF planar motion of MP.

$$\Omega_{w1} + \Omega_{w2} - \Omega_{w3} - \Omega_{w4} = 0 \quad (12)$$

The robot's linear and angular velocity could be determined in terms of wheels angular speeds using Equation (11) and the pseudo invers of Jacobian matrix  ${}^1J_w$  as follows.

$${}^1J^* = \left( {}^1J^T {}^1J \right)^{-1} {}^1J^T$$

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = {}^1J^* \Omega_w \quad (13)$$

Equation (13) is indeed the kinematic model of the OMR and can be used to calculate the velocity of the MP in terms of angular speed of the wheels.

To extract the dynamic model of the robot, the linear and angular acceleration of all robot parts should be specified. The linear and angular acceleration of the MP is determined as follows:

$${}^1\mathbf{a}_1 = [\ddot{x}, \ddot{y}, 0]^T$$

$${}^1\mathbf{a}_1 = {}^2\mathbf{a}_1 = [0, 0, \ddot{\theta}]^T \quad (14)$$

$${}^2\mathbf{a}_1 = {}^2\mathbf{R}^1\mathbf{a}_1$$

Substituting  $\Omega_{wi}$  and  $\Omega_{ri}$  from Equation (10) in Equations (6) and (8), the linear and angular velocity of the rollers and the angular velocity of the wheels are determined in terms of  $\mathbf{q}$  and  $\dot{\mathbf{q}}$ . Therefore, taking time derivative of Equations (5) and (8) in frame {1}, the linear and angular acceleration of these parts are calculated as follows:

$${}^2\mathbf{a}_i = \sum_{j=1}^3 \frac{\partial({}^2\boldsymbol{\omega}_i)}{\partial q_j} \dot{q}_j + \sum_{j=1}^3 \frac{\partial({}^2\boldsymbol{\omega}_i)}{\partial \dot{q}_j} \ddot{q}_j + {}^2\boldsymbol{\omega}_1 \times {}^2\boldsymbol{\omega}_i ; i = 2 \dots 9 \quad (15)$$

$${}^2\mathbf{a}_i = \sum_{j=1}^3 \frac{\partial({}^2\mathbf{V}_i)}{\partial q_j} \dot{q}_j + \sum_{j=1}^3 \frac{\partial({}^2\mathbf{V}_i)}{\partial \dot{q}_j} \ddot{q}_j + {}^2\boldsymbol{\omega}_1 \times {}^2\mathbf{V}_i ; i = 2 \dots 9 \quad (16)$$

## 2.2. Dynamic Modeling

In this paper, robot's differential equations of motion are extracted using Kane's method [32] which is a powerful approach to derive the dynamic model of complex multi-body systems. Considering  $\dot{\mathbf{q}} = (\dot{x}, \dot{y}, \dot{\theta})$  as three independent generalized velocities, the velocities of all parts of the robot are calculated in Section 2.1 in terms of the generalized velocities. Therefore, the linear and angular partial velocities of each part is defined as follows:

$${}^2\mathbf{V}_i^j = \frac{\partial({}^2\mathbf{V}_i)}{\partial \dot{\mathbf{q}}_j}; i = 1 \dots 9; j = 1 \dots 3 \quad (17)$$

$${}^2\boldsymbol{\omega}_i^j = \frac{\partial({}^2\boldsymbol{\omega}_i)}{\partial \dot{\mathbf{q}}_j}; i = 1 \dots 9; j = 1 \dots 3 \quad (18)$$

Substituting Equation (10) in Equation (6), partial relative angular velocities of wheels with respect to MP are also defined as follows:

$${}^2\boldsymbol{\omega}_{rel\ i}^j = \frac{\partial({}^2\boldsymbol{\omega}_k - {}^2\boldsymbol{\omega}_1)}{\partial \dot{\mathbf{q}}_j}; i = 1 \dots 4; j = 1 \dots 3; k = 2 \dots 5 \quad (19)$$

The inertia forces and moments of all robot's parts are determined using Equations (14)–(16):

$${}^2\mathbf{R}_i^* = -m_i {}^2\mathbf{a}_i; i = 1 \dots 9 \quad (20)$$

$${}^2\mathbf{T}_i^* = -\mathbf{I}_i {}^2\boldsymbol{\omega}_i - {}^2\boldsymbol{\omega}_i \times \mathbf{I}_i {}^2\boldsymbol{\omega}_i; i = 1 \dots 5 \quad (21)$$

where  $m_i$  is the mass of each part and  $\mathbf{I}_i$  is the moment of inertia matrix of each part about its principle axes. Since the principle axes of the rollers are parallel to the axes of frame {3} the inertia moments of rollers could be determined as follows:

$${}^2\mathbf{T}_i^* = {}^2\mathbf{R}_j (-\mathbf{I}_i {}^3\mathbf{R}_j {}^2\mathbf{a}_i) - {}^2\mathbf{R}_j {}^3(\boldsymbol{\omega}_i \times \mathbf{I}_i {}^3\boldsymbol{\omega}_i); i = 6 \dots 9; j = 1 \dots 4 \quad (22)$$

The only active forces exerted on robot's parts are the motors' torque and the viscous frictions between the wheels and the MP shafts, that are both associated with the relative angular velocities of the wheels with respect to the MP.

$$\begin{aligned} \mathbf{T}_i &= [0 \ 0 \ 0]^T; i = 1 \dots 9 \\ \mathbf{T}_{rel\ j} &= [0 \ u_j \ 0]^T - [\mu_f \Omega_{w\ j} \ 0]^T; j = 1 \dots 4 \end{aligned} \quad (23)$$

where  $u_j$ ;  $j = 1 \dots 4$  are motor's torque considered as the system control inputs and  $\mu_f$  is the coefficient of viscous friction. The generalized inertia and active forces and moments are calculated based on the Kane's definition as follows:

$$F_r^* = \sum_{i=1}^9 {}^2\mathbf{R}_i^* \cdot {}^2\mathbf{V}_i^r + \sum_{i=1}^9 {}^2\mathbf{T}_i^* \cdot {}^2\boldsymbol{\omega}_i^r; r = 1 \dots 3 \quad (24)$$

$$F_r = \sum_{i=1}^9 {}^2\mathbf{R}_i \cdot {}^2\mathbf{V}_i^r + \sum_{i=1}^9 {}^2\mathbf{T}_i \cdot {}^2\boldsymbol{\omega}_i^r + \sum_{i=1}^4 {}^2\mathbf{T}_{rel\ i} \cdot {}^2\boldsymbol{\omega}_{rel\ i}^r; r = 1 \dots 3 \quad (25)$$

Finally, the differential equations of motion of the robot are determined using Kane's equation in the following form:

$$F_r^* + F_r = 0; r = 1..3 \quad (26)$$

### 2.3. State Space Representation

Three differential equations of motion in Equation (26) describe the relationship between the generalized coordinates, their time derivatives and the motor's torque exerted on the robot's wheels. These equations could be rewritten in the following standard form:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} = -\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{A}(\theta)\mathbf{U} \quad (27)$$

where  $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{3 \times 3}$  is invertible positive definite mass matrix,  $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{3 \times 1}$  is a vector of centrifugal acceleration, Coriolis acceleration and viscous friction forces,  $\mathbf{A}(\theta) \in \mathbb{R}^{3 \times 4}$  is a matrix whose components are functions of MP orientation ( $\theta$ ) and robot geometrical parameters,  $\mathbf{U} = [u_1 \ u_2 \ u_3 \ u_4]^T \in \Phi \subset \mathbb{R}^4$  is the vector of motor's torque and  $\Phi$  is the input constraint set defined based on maximum allowable motors' torque as follows:

$$\Phi = \{ \mathbf{U} \in \mathbb{R}^4 \text{ s.t. } |u_i| \leq u_{\max}, i = 1..4 \} \quad (28)$$

Multiplying both sides of Equation (27) by  $\mathbf{M}(\mathbf{q})^{-1}$ ,  $\ddot{\mathbf{q}}$  is determined and the state space representation of the robot is obtained as follows.

$$\dot{\mathbf{X}} = \mathbf{g}_1(\mathbf{X}) + \mathbf{g}_2(\mathbf{X})\mathbf{U} \quad (29)$$

where  $\mathbf{X} = [\mathbf{q} \ \dot{\mathbf{q}}]^T \in \mathbb{R}^6$  is the vector of state variables,  $\mathbf{g}_1(\mathbf{X}): \mathbb{R}^6 \rightarrow \mathbb{R}^6$  is a nonlinear vector function of states,  $\mathbf{g}_2(\mathbf{X}): \mathbb{R}^6 \rightarrow \mathbb{R}^{6 \times 4}$  is a nonlinear input matrix,  $\mathbf{U}: \mathbb{R}_{0+} \rightarrow \Phi \subset \mathbb{R}^4$  is the control input vector and  $\dot{\mathbf{X}} = [\dot{\mathbf{q}} \ \ddot{\mathbf{q}}]^T \in \mathbb{R}^6$ . Mathematical model in Equation (29) is discretized to be used in designing the control system in the proceeding sections, as follows.

$$\mathbf{X}_{k+1} = \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k) = \int_{t_k}^{t_k + \Delta} (\mathbf{g}_1(\mathbf{X}(\tau)) + \mathbf{g}_2(\mathbf{X}(\tau))\mathbf{U}(\tau)) d\tau \approx (\mathbf{g}_1(\mathbf{X}_k) + \mathbf{g}_2(\mathbf{X}_k)\mathbf{U}_k)\Delta + \mathbf{X}_k \quad (30)$$

where  $\mathbf{X}_k \in \mathbb{R}^6$  is the system state in time  $t_k = k\Delta$ ;  $k = 0, 1, 2, \dots$ ,  $\mathbf{U}_k \in \Phi \subset \mathbb{R}^4$  is the vector of control inputs in time  $t_k$ ,  $\Delta$  is a sampling period and the control input signal  $\mathbf{U}(\tau) = \mathbf{U}_k$  is assumed to be constant in each sampling interval  $[t_k, t_{k+1}]$ .

### 3. Motion Control in Dynamic Environment

To use the OMR in collaboration with human or other mobile robots, its motion control in dynamic environments should be considered. Therefore, an obstacle avoidance strategy is developed for the robot to prevent its collision with static and moving obstacles. The obstacles are assumed to be cylindrical objects that move with known constant velocity on linear trajectories. Consequently, an obstacle with different shape could be considered as a cylindrical object with a radius equal to its largest dimension. It is also assumed that the radius, position and velocity of each obstacle could be evaluated once it is located in the robot's sensors' ranges.



### 3.1. Formulation of the VO Algorithm

In VO method firstly proposed by Fiorini et al. [31], a set of velocities called VO is determined for each obstacle and the velocity of the robot is calculated in which it is not a member of VO sets of all obstacles. This approach has been used in some previous studies [33–36] for the motion planning of mobile robots and vehicles in the presence of moving obstacles. Using the VO strategy, the problem of collision avoidance could be formulated as some velocity constraints that should be satisfied to guarantee the safe motion of the vehicle among obstacles. This method is formulated in this section in which it could be combined with the NMPC designed for stabilization of OMR in the next section.

To simplify the subsequent calculations, the robot is considered as a point and the largest dimension of the robot is added to the radius of all obstacles.

$$R_i^{en} = R_i^o + R_{robot} + R_{safe} \quad (31)$$

where  $R_i^{en}$  denotes the enlarged radius of  $i$ th obstacle,  $R_i^o$  is the nominal radius of  $i$ th obstacle,  $R_{robot}$  is equal to the largest dimension of the robot and  $R_{safe}$  is the radius of small safe region around the robot. Suppose that the robot is in position  ${}^1\mathbf{P}_r(t_k) = (x_k, y_k)^T$  in frame {1} at time  $t_k$ , the set of active obstacles  $\mathbf{O}_A(t_k)$  is defined as the number of all obstacles located in the robot's sensors ranges. More precisely:

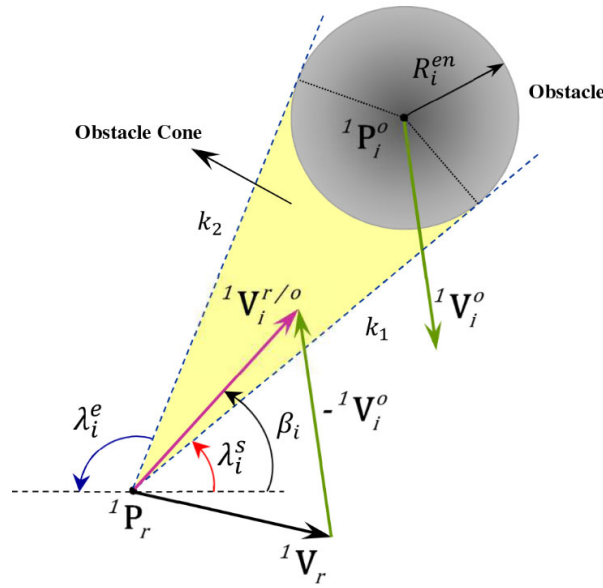
$$\mathbf{O}_A(t_k) = \left\{ i \in \mathbb{N} \mid i \leq N_O \text{ and } \left\| {}^1\mathbf{P}_r(t_k) - {}^1\mathbf{P}_i^o(t_k) \right\| - R_i^o - R_{robot} \leq \delta \right\} \quad (32)$$

where  $N_O$  is the number of obstacles,  ${}^1\mathbf{P}_i^o(t_k) = (x_i^o, y_i^o)^T$  is the position vector of  $i$ th obstacle in frame {1} at time  $t_k$ ,  $\delta$  is the robot's sensors ranges and  $\|\cdot\|$  denotes 2-norm operator. Suppose that the robot and an active enlarged obstacle move with velocities  ${}^1\mathbf{V}_r(t_k) = (\dot{x}_k, \dot{y}_k)^T$  and  ${}^1\mathbf{V}_i^o(t_k) = (\dot{x}_i^o, \dot{y}_i^o)^T$  in frame {1} respectively. The relative velocity of robot with respect to the obstacle is determined as follows.

$${}^1\mathbf{V}_i^{r/o}(t_k) = {}^1\mathbf{V}_r(t_k) - {}^1\mathbf{V}_i^o(t_k); i \in \mathbf{O}_A(t_k) \quad (33)$$

The collision cone is also defined as the planar conical region encircled by two tangent lines  $k_1$  and  $k_2$  as depicted in Figure 2. These tangent lines are specified by their angles  $\lambda_i^s$  and  $\lambda_i^e$  with respect to the x axis of frame {1}. These angles could be determined in each time  $t_k$  based on the enlarged obstacle radius and the position vectors of the robot and obstacle in frame {1}.

$$\begin{aligned} \lambda_i^s &= \tan^{-1} \left( \frac{y_i^o - y_r}{x_i^o - x_r} \right) - \sin^{-1} \left( \frac{R_i^{en}}{\left\| {}^1\mathbf{P}_r - {}^1\mathbf{P}_i^o \right\|_2} \right); i \in \mathbf{O}_A(t_k) \\ \lambda_i^e &= \tan^{-1} \left( \frac{y_i^o - y_r}{x_i^o - x_r} \right) + \sin^{-1} \left( \frac{R_i^{en}}{\left\| {}^1\mathbf{P}_r - {}^1\mathbf{P}_i^o \right\|_2} \right); i \in \mathbf{O}_A(t_k) \end{aligned} \quad (34)$$



**Figure 2.** The obstacle cone and relative velocity of the robot with respect to the obstacle.

The angle of the relative velocity vector  $^1V_i^{r/o}$  with respect to x axis of frame {1} is also determined as follows.

$$\beta_i = \tan^{-1} \left( \frac{{}^1V_i^{r/o} \cdot \mathbf{e}_y}{{}^1V_i^{r/o} \cdot \mathbf{e}_x} \right); i \in \mathbf{O}_A(t_k) \quad (35)$$

where  $\mathbf{e}_x$  and  $\mathbf{e}_y$  are unit vectors along x and y axes of frame {1} respectively. By defining the collision cone and relative velocity for each obstacle in time  $t_k$ , the collision avoidance criterion in each time instant is simply to prevent the robot from moving with the relative velocity vector inside the obstacle cone (Figure 2). The VO set associated with  $i$ th obstacle at time  $t_k$  is defined as follows.

$$\mathbf{VO}_i(t_k) = \left\{ {}^1V_r(t_k) \mid \lambda_i^s \leq \beta_i \leq \lambda_i^e, i \in \mathbf{O}_A(t_k) \right\} \quad (36)$$

It is obvious that the signs and values of angles  $\lambda_i^s$ ,  $\lambda_i^e$  and  $\beta_i$  should be properly specified in different quadrants such that the comparison in Equation (36) is meaningful. Using Equation (36), the total VO set in each time instant could be defined as follows.

$$\mathbf{VO}(t_k) = \bigcup_{i \in \mathbf{O}_A(t_k)} \mathbf{VO}_i(t_k) \quad (37)$$

where  $\bigcup$  is the operator of union of sets. In addition, the total set of robot's relative velocity vectors with respect to active obstacles in each time instant is also defined as follows.

$$\mathbf{V}_t^{r/o}(t_k) = \left\{ {}^1V_i^{r/o} \mid i \in \mathbf{O}_A(t_k) \right\} \quad (38)$$

Therefore, the collision avoidance criterion in each time  $t_k$  could be written in the following form.

$$\mathbf{V}_t^{r/o}(t_k) \cap \mathbf{VO}(t_k) = \emptyset \quad (39)$$

where  $\cap$  is the operator of subscribe of sets and  $\emptyset$  denotes an empty set. The criterion in Equation (39) is indeed a set of constraints on the robot's velocity vector that should be satisfied in each time step to prevent the robot from collision with obstacles. These constraints are used to specify the collision free trajectory online as a part of control system.

### 3.2. Model Predictive Controller

MPC is a control technique in which the control input signals are calculated based on the anticipated states of the system on a specified prediction horizon and solving online the resulting open loop optimal control problem (OCP). This construction provides a context to consider complicated constraints of the system online during the control process. Therefore, considering the collision avoidance criterion formulated in the previous section, a NMPC system is designed to move the robot towards the target position and orientation among obstacles.

To design a NMPC for discretized mathematical model of the OMR in Equation (30), the following cost function is defined.

$$V_N(\mathbf{X}_k, \tilde{\mathbf{U}}_{N-1}) = \sum_{m=0}^{N-1} \{C_1(\mathbf{X}_{k+m} - \mathbf{X}_d, \mathbf{U}_{k+m})\} + C_f(\mathbf{X}_{k+N} - \mathbf{X}_d) \quad (40)$$

where  $C_1: \mathbb{R}^6 \times \Phi \rightarrow \mathbb{R}_{0+}$  is a positive definite stage cost,  $C_f: \mathbb{R}^6 \rightarrow \mathbb{R}_{0+}$  is a positive definite terminal cost,  $N$  is the prediction horizon,  $\mathbf{X}_d = [\mathbf{q}_d, \dot{\mathbf{q}}_d]^T$  is the desired state,  $\tilde{\mathbf{U}}_N$  is a sequence of  $N$  input vectors  $\mathbf{U}_i \in \Phi$  and  $V_N(0, 0) = 0$ . Therefore, the position and orientation of the OMR could be controlled by solving the following optimal control problem ( $\text{OCP}_N$ ).

$$\text{OCP}_N(t_k): \min_{\tilde{\mathbf{U}}_N} \left\{ V_N(\mathbf{X}_k, \tilde{\mathbf{U}}_N) = \sum_{m=0}^{N-1} \{C_1(\mathbf{X}_{k+m} - \mathbf{X}_d, \mathbf{U}_{k+m})\} + C_f(\mathbf{X}_{k+N} - \mathbf{X}_d) \right\}$$

s.t :

1.  $\mathbf{X}_{k+1} = \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k)$
  2.  $\mathbf{X}_{k+m} \in \Psi_{k+m} \subset \mathbb{R}^6$
  3.  $\mathbf{X}_{k=0} = \mathbf{X}_0 \in \Psi_0$
  4.  $\mathbf{X}_{k+N} \in \Psi_f \subset \Psi_{k+N}$
  5.  $\mathbf{U}_0 = \mathbf{0} \in \Phi$
- (41)

where  $\Psi_k$  and  $\Psi_f$  are state constraints set at time  $t_k$  and terminal state constraints set respectively. These sets could be determined based on the available physical constraints and collision avoidance criterion from Equation (39).

The constraints in Equation (41) are related to the robot's mathematical model and state constraints imposed to guarantee the stability and collision avoidance in the workspace. Constraint 1 guarantees that the predicted states satisfy the robot state space model. Constraint 2 is a set of state constraints imposed to avoid collision with obstacles and defined workspace borders. Constraints 3 and 5 are associated with the initial values of states and inputs, and constraint 4 is a terminal constraint determined in detail through stability analysis in the next section.

By solving  $\text{OCP}_N$  in each time instant  $t_k$ , a set of  $N$  optimal control inputs sequence is obtained as follows:

$$\tilde{\mathbf{U}}_N^* = \{\mathbf{U}_k^*, \mathbf{U}_{k+2}^*, \dots, \mathbf{U}_{k+N-1}^*\} \quad (42)$$

The feedback control law is defined as the first element of this optimal inputs sequence as follows, and the remaining optimal input vectors are used to form an initial guess to solve  $\text{OCP}_N(t_{k+1})$  at the next step.

$$\mathbf{U}_k = \kappa(\mathbf{X}_k - \mathbf{X}_d) = \mathbf{U}_k^* \quad (43)$$

Therefore, the closed loop system could be written using the feedback law in the following form:

$$\mathbf{X}_{k+1} = \mathbf{f}(\mathbf{X}_k, \kappa(\mathbf{X}_k - \mathbf{X}_d)); \mathbf{X}_{k=0} = \mathbf{X}_0 \in \Psi_k \quad (44)$$

The state constraints set  $\Psi_k$  at each time  $t_k$  are composed of some constraints on the robot's position and velocity and should be determined based on the physical restrictions of environment, obstacle's positions and obstacle's velocities. The constraints on position are indeed associated with the boundaries of the robot workspace which in turn are functions of the robot's position at each time  ${}^1\mathbf{P}_r(t_k) = (x_k, y_k)$ . Mathematically:

$${}^1\mathbf{P}_r(t_k + j\Delta) \in \mathbf{S}_{k+j\Delta} \subset \mathbb{R}^2; j = 1..N \quad (45)$$

where  $\mathbf{S}_k$  is a closed region in frame {1} defined as:

$$\mathbf{S}_k = \left\{ (x_k, y_k) \in \mathbb{R}^2 \mid x_{\min}({}^1\mathbf{P}_r(t_k)) \leq x_k \leq x_{\max}({}^1\mathbf{P}_r(t_k)) \text{ and } y_{\min}({}^1\mathbf{P}_r(t_k)) \leq y_k \leq y_{\max}({}^1\mathbf{P}_r(t_k)) \right\} \quad (46)$$

where  $x_{\min}(\cdot)$ ,  $x_{\max}(\cdot)$ ,  $y_{\min}(\cdot)$  and  $y_{\max}(\cdot)$  are differentiable functions of the robot's position vector. It is obvious that for a rectangular workspace with boundaries parallel to the axes of frame {1}, these functions are simplified to constant values. To specify the state constraints associated with the collision avoidance criterion, suppose that system's state at time  $t_k$  is  $\mathbf{X}_k = [\mathbf{q}_k, \dot{\mathbf{q}}_k]^T$  and the locations and velocities of the active obstacles are calculated using robot's sensors. The locations of the active obstacles during the prediction horizon are calculated as follows.

$${}^1\mathbf{P}_i^o(t_k + j\Delta) = {}^1\mathbf{P}_i^o(t_k) + j\Delta {}^1\mathbf{V}_i^o; i \in \mathbf{O}_A(t_k), j = 1..N \quad (47)$$

Using Equation (47), angles  $\lambda_i^s$ ,  $\lambda_i^e$  and  $\beta_i$  are calculated at time  $t_k + j\Delta$ ;  $j = 1..N$  (Equations (34) and (35)) and the total velocity obstacles set could also be determined at each time using Equation (37). Therefore, the state constraints originated from collision avoidance criterion in Equation **Error! Reference source not found.** is as follows.

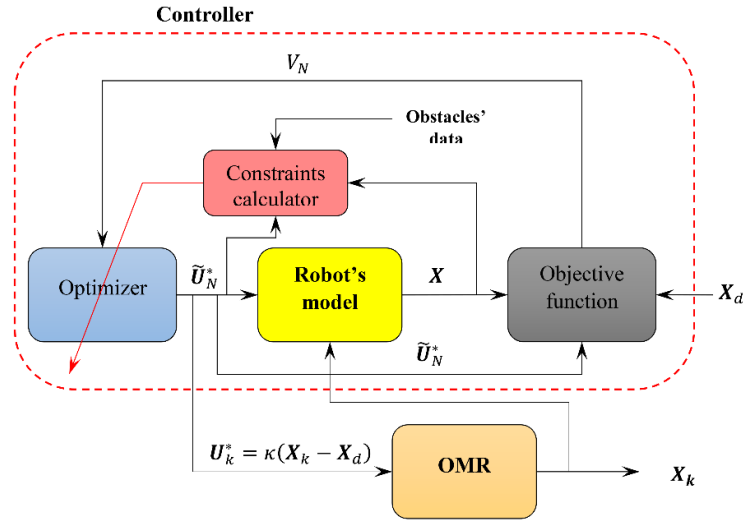
$$\mathbf{V}_t^{r/o}(t_k + j\Delta) \cap \mathbf{VO}(t_k + j\Delta) = \emptyset; j = 1..N \quad (48)$$

Finally, the state constraints set could be defined in the following form.

$$\Psi_{k+j} = \left\{ \mathbf{X}_{k+j\Delta} = [\mathbf{q}_{k+j\Delta}, \dot{\mathbf{q}}_{k+j\Delta}]^T \mid {}^1\mathbf{P}_r(t_k + j\Delta) \in \mathbf{S}_{k+j\Delta} \text{ and } \mathbf{V}_t^{r/o}(t_k + j\Delta) \cap \mathbf{VO}(t_k + j\Delta) = \emptyset \right\}; j = 1..N \quad (49)$$

This constraints set is indeed a set of  $4N + N \times N_k (O_A)$  nonlinear constraints on robot's position and velocity vector during the prediction horizon, where  $N_k (O_A)$  is the number of active obstacles at time  $t_k$ . These constraints guarantee that the robot remain inside the defined workspace during its motion and avoid collision with obstacles.

The structure of the designed control system and obstacle avoidance strategy is demonstrated in the following diagram. Figure 3 illustrates how the optimization problem and constraints are defined in each time step and the first element of the optimal control sequence is exerted to the real robot.



**Figure 3.** The structure of the designed control system.

### 3.3. Stability and Performance Analysis

The parameters and functions used to design the feedback control system in Equation (43) should be specified in which the convergence of the state variables to desired values and the acceptable performance of the obstacle avoidance strategy are guaranteed. Since the feedback control input is calculated based on the position and velocity of active obstacles, it is also assumed that the position and velocity of obstacles at time  $t_k$  are such that it is possible for the robot to remain inside the workspace boundaries (constraints in Equation (45)) and move towards the target position in  $t \geq t_k$  without collision. The first part of this assumption could be written mathematically as follows:

$$\forall \mathbf{X}_k \in \Psi_k \exists \mathbf{U}_k \in \Phi \text{ s.t } \mathbf{X}_{k+1} = \mathbf{f}(\mathbf{X}_k, \mathbf{U}_k) \in \Psi_{k+1} \quad (50)$$

The second part of the assumption is associated with the possibility of moving towards the desired position without collision with obstacle and could be written mathematically as follows.

$$\begin{aligned} \forall \mathbf{X}_k \in \Psi_k \exists m \in \mathbb{N} \text{ and } \{U_k, \dots, U_{k+m}\} \in \Phi \text{ s.t } \mathbf{X}_{k+j} = \mathbf{f}(\mathbf{X}_{k+j-1}, \mathbf{U}_{k+j-1}) \in \Psi_{k+j} \\ \text{and } C_f(\mathbf{X}_{k+m} - \mathbf{X}_d) - C_f(\mathbf{X}_k - \mathbf{X}_d) < 0 ; (j = 1..m) \end{aligned} \quad (51)$$

Condition in Equation (51) means that at each time  $t_k$ , the robot may stop at its position or move in opposite direction of desired position to avoid collision with obstacles, but it can eventually moves towards the goal at time  $t_{k+m}$  that the position of obstacles change. Owing to the robot capability for omnidirectional locomotion, satisfaction of (51)

is guaranteed for bounded speed of obstacles, existence of a route toward the target after the time  $t_{k+m}$  and proper choice of control system parameters. Considering (50) and (51), the stage and terminal cost functions  $C_1(\cdot, \cdot)$  and  $C_f(\cdot)$  are selected as functions of  $\mathbf{X} - \mathbf{X}_d$  in the following forms:

$$\begin{aligned} C_1(\mathbf{X} - \mathbf{X}_d, \mathbf{U}) &= (\mathbf{X} - \mathbf{X}_d)^T \mathbf{Q}_1 (\mathbf{X} - \mathbf{X}_d) + \mathbf{U}^T \mathbf{Q}_u \mathbf{U} \\ C_f(\mathbf{X} - \mathbf{X}_d) &= (\mathbf{X} - \mathbf{X}_d)^T \mathbf{Q}_f (\mathbf{X} - \mathbf{X}_d) \end{aligned} \quad (52)$$

where  $\mathbf{Q}_1 (6 \times 6)$ ,  $\mathbf{Q}_f (6 \times 6)$  and  $\mathbf{Q}_u (4 \times 4)$  are diagonal matrixes defined as follows:

$$\begin{aligned} \mathbf{Q}_1 &= \text{diag}(w_1, w_2, w_3, w_4, w_5, w_6) \\ \mathbf{Q}_f &= \zeta_f \mathbf{Q}_1 \\ \mathbf{Q}_u &= \zeta_u \mathbf{I}_4 \end{aligned} \quad (53)$$

where  $\mathbf{I}_4$  is an identity matrix and  $w_i$ , ( $i = 1..6$ ),  $\zeta_f$  and  $\zeta_u$  are non-negative values. These values should be determined such that the closed loop system in Equation (44) is stable. It is clear that terminal cost  $C_f(\cdot)$  is defined as a function of state errors such that assumption in Equation (51) could be satisfied by moving the robot toward the goal. Now, suppose that  $\mathbf{O}_A(t_k) = \emptyset$  and  $\Psi_f = \Psi_{k+N}$ , omnidirectional capability of the robot implies that condition in Equations (50) and (51) are satisfied for  $m = 1$ . Therefore,  $\Psi_f$  is a control invariant set and:

$$\begin{aligned} \sigma &= C_1(\mathbf{X}_k, \mathbf{U}_k) + C_f(\mathbf{X}_k) - C_f(\mathbf{X}_{k+1}) = C_1(\mathbf{X}_k, \mathbf{U}_k) + C_f(\mathbf{X}_k) - C_f(\mathbf{f}(\mathbf{X}_k, \mathbf{U}_k)) \leq \\ &w_{\max} \|\mathbf{X}_k - \mathbf{X}_d\|^2 + 4\zeta_u u_{\max}^2 + \zeta_f w_{\max} (\|\mathbf{X}_{k+1} - \mathbf{X}_d\|^2 - \|\mathbf{X}_k - \mathbf{X}_d\|^2) \end{aligned} \quad (54)$$

where  $w_{\max} = \max(w_1, \dots, w_6)$ . Considering that  $\|\mathbf{X}_{k+1} - \mathbf{X}_d\|^2 - \|\mathbf{X}_k - \mathbf{X}_d\|^2 < 0$  (assumption (51)),  $\sigma \leq 0$  if the following is held.

$$\zeta_f \geq \frac{w_{\max} \|\mathbf{X}_k - \mathbf{X}_d\|^2 + 4\zeta_u u_{\max}^2}{w_{\max} (\|\mathbf{X}_{k+1} - \mathbf{X}_d\|^2 - \|\mathbf{X}_k - \mathbf{X}_d\|^2)} \quad (55)$$

Since  $\|\mathbf{X}_{k+1} - \mathbf{X}_d\|^2 - \|\mathbf{X}_k - \mathbf{X}_d\|^2 \rightarrow 0$  as robot moves towards the desired state ( $\mathbf{X} \rightarrow \mathbf{X}_d$ ), for a large value of  $\zeta_f$ ,  $C_f$  is a control Lyapunov function (CLF) on  $\Psi_f - \boldsymbol{\varepsilon}$  where  $\boldsymbol{\varepsilon}$  is a small neighborhood of the desired state. Using a sufficiently large value for  $\zeta_f$ ,  $\boldsymbol{\varepsilon}$  is sufficiently small and cost function  $V_N$  monotonically decreases during the robot's motion [37]. Therefore, the robot's state variables converge to small vicinity of the desired state ( $\boldsymbol{\varepsilon}$ ) as  $t \rightarrow \infty$ .

If  $\mathbf{O}_A(t_k) \neq \emptyset$ , the state constraints (49) guarantees that the robot moves without collision with obstacles. Therefore, if assumption in Equation (51) is held, the robot either moves among the obstacles toward the goal or the cost function in Equation (40) may have a local optima in opposite direction of the goal, but after some time ( $m \geq 1$ ) that obstacles move and the related velocity constraints removed, the robot can continue its motion towards the target position according to the above analysis.

Therefore, the above analysis shows that if the layout and velocity of obstacles satisfy the conditions in Equations (50) and (51) by proper choice of control system parameters, the designed control system moves the robot on collision free trajectory and stabilizes it in the desired configuration.

### 3.4. Control System Parameters Setup

The parameters of the control system should be specified in such a way that satisfaction of Equations (50) and (51) convergence of the state errors to zero and the acceptable performance of the collision avoidance strategy are guaranteed. As analyzed mathematically in Equations (52)–(55), parameters  $\zeta_f$  and  $\zeta_u$  affect the stability of the control system and convergence of errors to zero in the vicinity of the target position. Parameter  $\zeta_u$  is added in the cost function in Equation (40) to guarantee that the control inputs tend to zero and the robot completely stops in the desired configuration. Indeed, its value is important merely when the robot is in the neighborhood of the desired state and the state errors are very small. Therefore, the value of  $\zeta_u$  could be simply selected as  $\zeta_u = 1$ . In addition, based on Equation (55),  $\zeta_f$  should be a large positive number to guarantee stability and convergence of the state errors to a small vicinity of the desired state.

Parameter  $\delta$  should be determined such that if the robot and an obstacle move toward each other at their maximum speeds, the robot is far enough away from the obstacle to avoid collision when the obstacle is detected by the robot's sensors for the first time. The maneuver that the robot could perform in such situation is to reduce its speed to zero with maximum acceleration and then move in opposite direction with maximum acceleration to reach the maximum allowable speed of the obstacles. Therefore, parameter  $\delta$  should be greater than the distance that the robot and the obstacle move toward each other during this maneuver's time ( $t_m$ ). More precisely:

$$\delta \geq (d_{stop}^r + v_{max}^o t_m + R_{safe}) \cdot n \quad ; \quad n \geq 1 \quad (56)$$

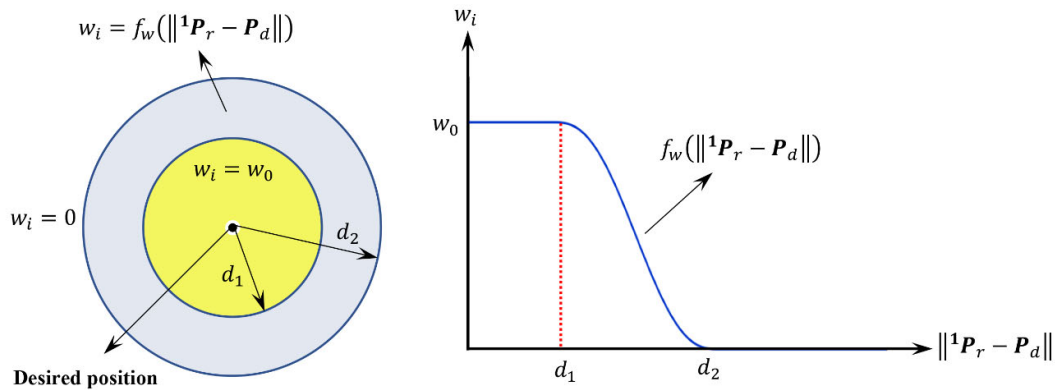
where  $v_{max}^o$  is the maximum allowable speed of obstacles and  $d_{stop}^r$  denotes the distance that the robot travels before stop. Integer  $n$  in Equation (56) is indeed a safety factor and its value could be specified considering the maximum real sensors rang. While the smaller values of  $n$  and  $\delta$  leads to ignore the effects of far obstacles and may increase the maneuverability of the robot, the larger value of  $\delta$  may increase the safety of the robot maneuvers. However, using the larger value of  $\delta$  imposes more restrictions on the robot motion and could reduce the maneuverability of the robot in some circumstances.

Parameter  $R_{safe}$  is the radius of a safe region around the robot introduced to guarantee the safety of the robot motion. While state constraints in Equation (49) guarantee that the robot doesn't collide with any obstacles at each discrete time and state ( $t_k$  and  $\mathbf{X}_k$  ( $k = 1..N$ )), the robot's motion is not restricted by these constraints during each sampling interval (between the times  $t_k$  and  $t_{k+1}$ ) and the robot may collide with close obstacles during these times. Since due to the impact of constraints in Equation (49) the distance of the robot from all active obstacles is greater than  $R_{safe}$  at each  $\mathbf{X}_k$  ( $k = 1..N$ ), the value of  $R_{safe}$  should be selected as the maximum distance that the robot could travel during each sampling period which is equal to  $R_{safe} = v_{max}^r \cdot \Delta$ . In this equation,  $v_{max}^r$  denotes the maximum robot's speed achieved after some time that the robot moves on straight line with maximum equal motor torques ( $u_1 = u_2 = u_3 = u_4 = \pm u_{max}$ ).

Parameters  $w_1$  to  $w_6$  also affect the performance of the control system and collision avoidance algorithm. These parameters are weights coefficients for state variable errors that define the priority of each state to be converged to its desired value. Consequently, since the value of the orientation angle  $\theta$  and the linear and angular velocity of the robot should be controlled merely at the desired position and are not important during the robot's motion, their associated weights are defined as follows:

$$\begin{aligned}
 w_1 &= w_2 = w_0 \\
 w_3 &= \begin{cases} w_0 & \|{}^1\mathbf{P}_r - \mathbf{P}_d\| < d_1 \\ f_w(\|{}^1\mathbf{P}_r - \mathbf{P}_d\|) & d_1 \leq \|{}^1\mathbf{P}_r - \mathbf{P}_d\| \leq d_2 \\ 0 & \|{}^1\mathbf{P}_r - \mathbf{P}_d\| > d_2 \end{cases} \\
 w_4 &= w_5 = w_6 = w_3
 \end{aligned} \tag{57}$$

where  $w_0$  is a constant final weights value,  $d_1$  and  $d_2$  are radius of two small circular region around the desired position,  $\mathbf{P}_d$  denotes target position and  $f_w(\cdot)$  is a scalar function that should be continuous and two time differentiable at  $d_1$  and  $d_2$ . The geometrical interpretation of Equation (57) is illustrated in Figure 4 in which the weight of position error is constant value  $w_0$  and the weights of other state variables increase from their initial value 0 to their final value  $w_0$  when the robot enters into a small neighborhood of the desired position.



**Figure 4.** Geometrical interpretation of weights variations in the vicinity of target position.

It is obvious that the length of prediction horizon ( $N$ ) could also have influence on the performance of the designed controller and collision avoidance algorithm. Since increasing the length of prediction horizon increases the computational time and cost significantly, the prediction horizon should be selected as the smallest value that leads to acceptable performance of the control system. Therefore, this value could be specified through trial and error and by evaluating the performance of the control system for different small values of  $N$ .

#### 4. Simulation Results and Discussion

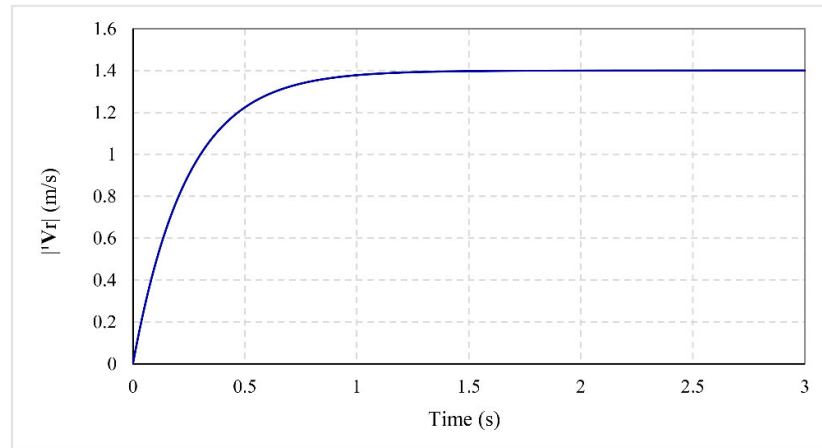
To evaluate the effectiveness of the designed controller and collision avoidance algorithm two simulation experiments are conducted using MATLAB software. In these simulations, the robot moves from its initial position and orientation among the obstacles and



stops at the desired state using the designed control system. The simulation is performed using the values  $L = 0.15\text{ m}$ ,  $H = 0.1\text{ m}$ ,  $R_w = 0.07\text{ m}$ ,  $R_r = 0.01\text{ m}$ ,  $\phi = 45^\circ$ ,  $u_{\max} = 1\text{ N.m}$  and  $\mu_f = 0.05\text{ N.m.s}$  for the robot geometrical and physical parameters and following values for masses and moment of inertia matrixes of the robot parts.

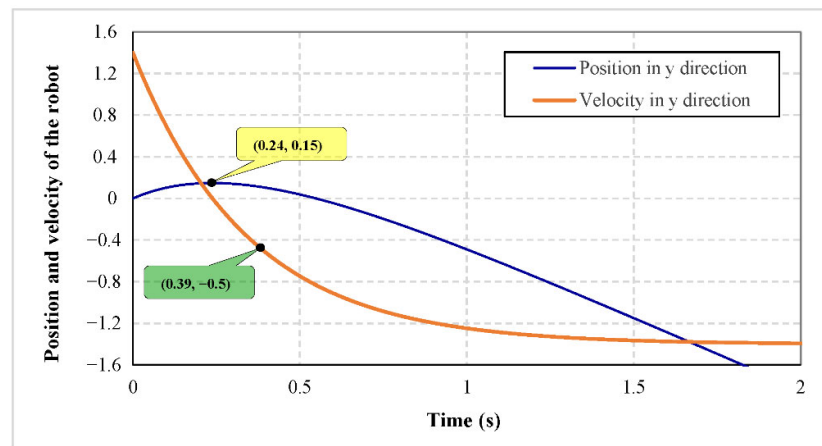
$$\begin{aligned}
 m_1 &= 4\text{ kg} \\
 m_2 &= m_3 = m_4 = m_5 = 1\text{ kg} \\
 m_6 &= m_7 = m_8 = m_9 = 0.2\text{ kg} \\
 I_{1zz} &= 0.17\text{ kgm}^2 \\
 I_2 &= I_3 = I_4 = I_5 = \text{diag}(13 \times 10^{-4}, 25 \times 10^{-4}, 13 \times 10^{-4}) \\
 I_6 &= I_7 = I_8 = I_9 = \text{diag}(3 \times 10^{-5}, 10^{-5}, 3 \times 10^{-5})
 \end{aligned} \tag{58}$$

The control input vector could be calculated by solving optimal control problem in Equation (41) in each time step. To simplify the optimization problem and to relax the constraints related to the mathematical model of the robot, recursive discretization method [38] is used to introduce the state space model in the cost function and to reduce the numbers of nonlinear constraints. In recursive discretization method, each future state vector is calculated recursively using the previous state vector and the associated control input, and the value of the cost function is determined using the calculated states and the inputs. To solve the resulting constrained optimization problem, sequential quadratic programming (SQP) method is used here. This method is provided by MATLAB optimization toolbox (fmincon function) and it is used to solve the optimization problem in Equation (41) in each step. The control system parameters are also selected based on (55) to (57) and other criteria discussed in the previous section as  $\Delta = 0.1\text{ s}$ ,  $N = 7$ ,  $R_{\text{robot}} = \sqrt{L^2 + H^2} = 0.1803$ ,  $w_0 = 1$ ,  $d_1 = 0.05$ ,  $d_2 = 0.3$ ,  $\zeta_u = 1$  and  $\zeta_f = 10^8$ . Furthermore, function  $f_w(\cdot)$  could be determined as a fifth-order polynomial based on six constraints  $f_w(d_1) = w_0$  and  $f_w(d_2) = f'_w(d_1) = f'_w(d_2) = f''_w(d_1) = f''_w(d_2) = 0$ . To calculate the maximum achievable speed of the robot  $v_{\max}^r$ , differential equations of motion of the robot in Equation (29) are solved for maximum allowable motors' torque ( $\mathbf{U} = [1, 1, 1, 1]^T \text{ N.m}$ ) and  $\theta_0 = 0$ . As demonstrated in Figure 5, the robot's speed has an upper bound at ( $v_{\max}^r = 1.4 \frac{\text{m}}{\text{s}}$ ), due to the effects of viscous friction between the wheels and shafts. Therefore, the radius of the safe region is determined as  $R_{\text{safe}} = v_{\max}^r \cdot \Delta = 0.14\text{ m}$ .



**Figure 5.** The maximum achievable velocity of the robot.

To calculate the values of parameters  $d_{stop}^r$  and  $t_m$ , the robot motion is simulated in a decelerating maneuver described in Section 3.4. Since  $d_{stop}^r$  is defined as the maximum distance that the robot travels to reduce its speed from  $v_{max}^r$  to zero, the decelerating maneuver in lateral motion is considered in which the robot moves with minimum acceleration. Therefore, the differential equations of motion are solved for  $\mathbf{U} = [1, -1, -1, 1]^T N.m$  and initial state  $\mathbf{X}_0 = [0, 0, 0, 0, v_{max}^r, 0]^T$  which is related to decelerating lateral motion of the robot. The variations of the position and velocity of the robot are shown in Figure 6. Suppose that the maximum allowable speed of obstacles is  $v_{max}^o = 0.5 \frac{m}{s}$ , the value of  $d_{stop}^r$  is the robot's position when its velocity reduces to zero, and the value of  $t_m$  is the time needed for the robot to reach to the maximum velocity of the obstacles in opposite direction ( $-v_{max}^o$ ).



**Figure 6.** Position and velocity of the robot in decelerating motion.

The parameters  $d_{stop}^r$  and  $t_m$  could be calculated from Figure 6 and according to their definition in Section 3.4 as  $d_{stop}^r = 0.15 m$  and  $t_m = 0.39 s$ . Therefore, using Equation (56),  $\delta \geq 0.5n$ , and considering  $n = 2$  and  $\delta = 1 m$ , the performance of the control system is assessed in two following examples.

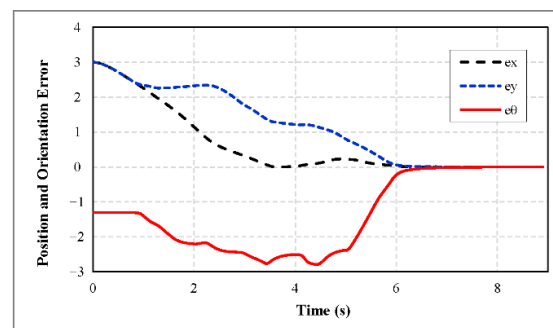
**Example 1:** The robot starts its motion from initial configuration  $\mathbf{X}_0 = [3, 3, \frac{\pi}{4}, 0, 0, 0]^T$  to the desired state  $\mathbf{X}_d = [0, 0, \frac{\pi}{2} + \frac{\pi}{6}, 0, 0, 0]^T$  inside a rectangular region. The robot's workspace is defined as follows and the initial positions, velocities and radiuses of obstacles are also specified as shown in Table 1.

$$\mathbf{S}_k = \left\{ (x_k, y_k) \in \mathbb{R}^2 \left| \begin{array}{l} -0.2 + R^* \leq x_k \leq 3.4 - R^* \text{ and} \\ -0.2 + R^* \leq y_k \leq 3.4 - R^* \end{array} \right. \right\}; R^* = R_{robot} + R_{safe} \quad (59)$$

**Table 1.** Obstacles specification in Example 1.

Obstacle Num.	1	2	3	4
$R_i^o$	0.2	0.2	0.15	0.15
${}^1\mathbf{P}_i^o$	(1.5, 1.5)	(2.5, 0.6)	(0.8, 2.5)	(2.25, 1)
${}^1\mathbf{V}_i^o$	(0, 0)	(−0.5, 0)	(0, −0.27)	(0, 0.4)

The position and orientation errors of the robot are demonstrated in Figure 7. It is clear that the position and orientation errors converge to zero and the robot is well stabilized in the desired position and orientation using feedback control law in Equation (43). To evaluate the performance of the designed control system in planning the robot's trajectory and avoiding collision with obstacles the distances between the robot and obstacles ( $\|{}^1\mathbf{P}_r - {}^1\mathbf{P}_i^o\| - R_i^o - R_{robot}$ ) during the robot's motion are illustrated in Figure 8. The figure shows that the robot preserves the safe distance with all the obstacles in the period of its motion. It is also to be noted that since the proposed collision avoidance strategy is formulated to be combined with discrete NMPC, the distances of the robot from the obstacles are not necessarily larger than  $R_{safe}$  during the robot's motion. Indeed, using velocity constraints in Equation (48), the robot does not collide with any obstacle, if the velocity vector of the robot remains outside of the velocity obstacle set for all  $t > t_k$ . Since the robot velocity is not constant, radius  $R_{safe}$  is added to the radius of obstacles to ensure collision with any obstacles not occurs during each sampling interval. This is why the value of  $R_{safe}$  is determined as the maximum distance that the robot could travel during each sampling interval ( $R_{safe} = v_{\max}^r \cdot \Delta$ ).



**Figure 7.** Position and orientation error of the robot.

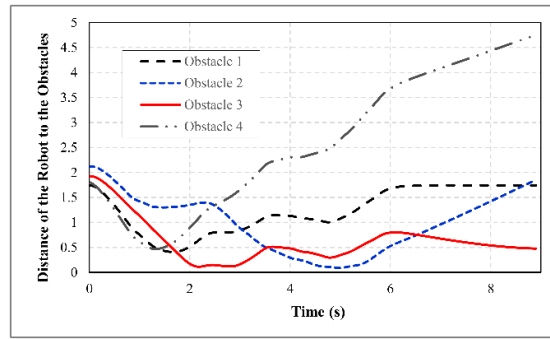


Figure 8. The robot's distance from the obstacles during its motion.

Figure 9 shows the snapshot of the robot trajectory in the presence of obstacles in which the inactive obstacles are shown by yellow color and the workspace boundaries are demonstrated by black lines. The figure shows that the arrangement and motion of the obstacles are such that the conditions in Equations (50) and (51) are satisfied and there exist a trajectory in each time step that the robot could become much closer to the target position.

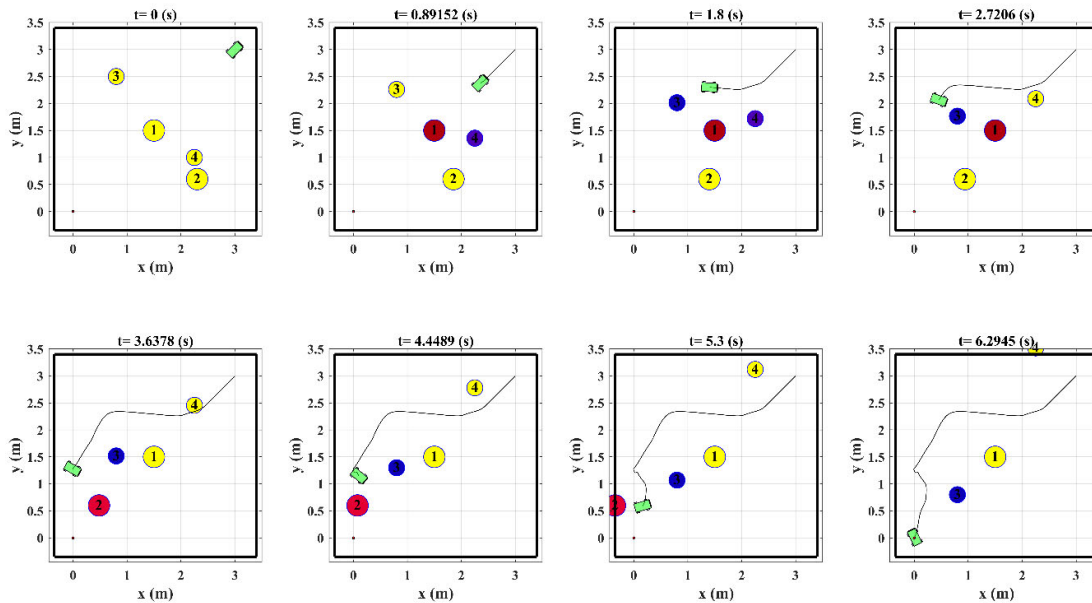


Figure 9. The snapshot of the robot trajectory among the obstacles in example 1.

**Example 2:** The robot's initial configuration and the desired state are selected as

$\mathbf{X}_0 = [1, 4.5, \pi, 0, 0, 0]^T$  and  $\mathbf{X}_d = [3.7, 0.6, \frac{\pi}{2}, 0, 0, 0]^T$  respectively. The robot's workspace is a region encircled by four following differentiable functions and could be defined mathematically using (46).

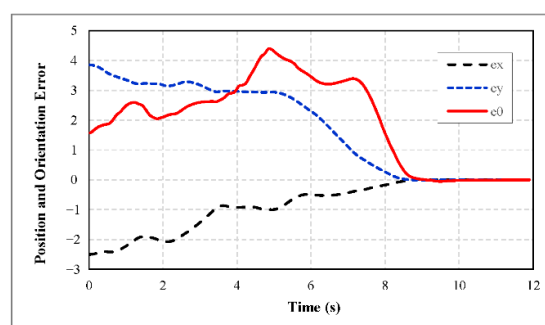
$$\begin{aligned} 1: x = 0; \quad 2: y = 5; \quad 3: y = -\frac{1}{5}x + 1 \\ 4: y = 2.5 + \sqrt{(2.7^2 - (x - 6)^2)} \end{aligned} \quad (60)$$

The initial position, velocity and radius of obstacles are selected according to Table 2.

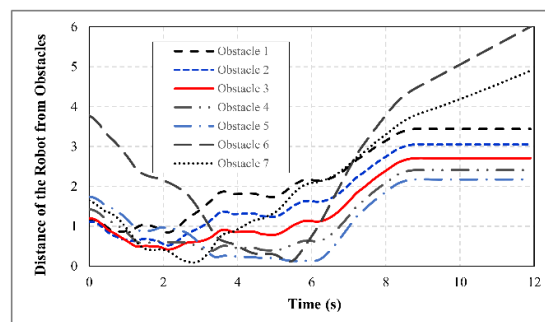
Figures 10–12 show the simulation results of this example using the above values for the robot and control system parameters. It is clear from Figure 10 that the position and orientation errors of the robot converge to a sufficiently small vicinity of zero as it is explained in Section 3.3.

**Table 2.** Obstacles specification in Example 2.

Obstacle Num.	1	2	3	4	5	6	7
$R_i^o$	0.27	0.2	0.2	0.2	0.27	0.13	0.18
${}^1\mathbf{P}_i^o$	(0.4,3)	(1,3)	(1.5,3)	(2,3)	(2.6,3)	(3.1,1.1)	(3.5,4.5)
${}^1\mathbf{V}_i^o$	(0,0)	(0,0)	(0,0)	(0,0)	(0,0)	(0,0.5)	(−0.5,−0.38)

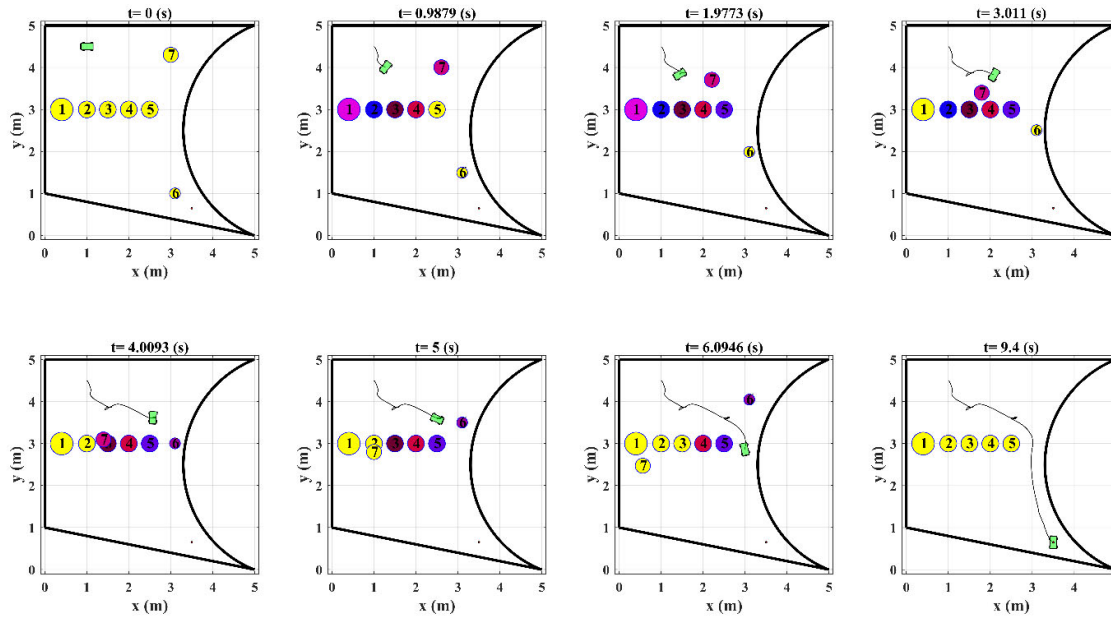


**Figure 10.** Position and orientation error of the robot.



**Figure 11.** The robot's distance from the obstacles during its motion.

The distance of the robot from each obstacle during the robot motion is demonstrated in Figure 11, which shows the effectiveness of the developed control strategy in protecting the robot from collision with obstacles.

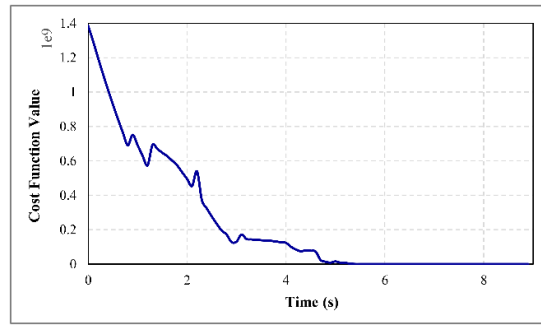


**Figure 12.** The snapshot of the robot trajectory among the obstacles in example 2.

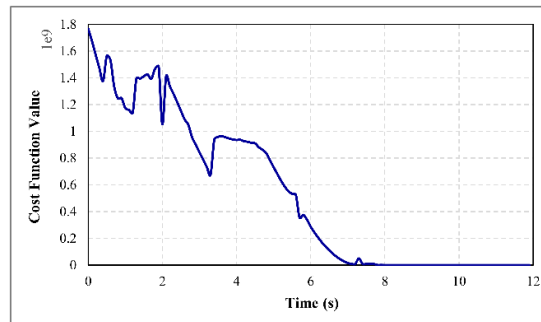
As illustrated in Figure 12, the snapshot of the robot trajectory shows that the robot remains inside the defined workspace during its motion and does not violate the related position constraints. Figure 12 demonstrates that the region between obstacle 5 and righter border of the workspace is the only safe region which the robot can move towards the target without collision with any obstacles. However, since this region is blocked by obstacle 6, the control system stops the robot in a fragment of its motion and waits for the path to be opened for the next robot movements. This maneuver is an example of situations describe by assumption in Equation (51) and shows the effectiveness of the motion planning algorithm in such complicated arrangement of obstacles.

In addition, the arrangement and velocity of obstacles in some step times may be such that the cost function  $V_N$  has a local minima point different with the desired state. In these situations, the velocity constraints in Equation (48) push the robot to reach the local minima point. However, since terminal cost function  $C_f(\cdot)$  is selected to be a CLF, the robot continues its motion towards the goal after that the obstacles move away and the minimum point of  $V_N(\cdot)$  is changed.

Figures 13 and 14 demonstrate the variation of the optimal value of the cost function  $V_N(\cdot)$  (Equation (40)) in examples 1 and 2 obtained by solving the nonlinear constrained optimization problem in Equation (41) in each time step. It is obvious that the cost functions decrease in both cases to reach their global minima at zero, which is associated with the stabilization of the robot in desired position and orientation. However, the optimal value functions have some optimums and there are some points that the optimal value of the cost functions increases temporarily. These optimums are in fact the local minimums of the cost function in Equation (40) considering the position and velocity constraints in Equation (49) defined based on the workspace borders and collision avoidance criterion at the corresponding times. As explained mathematically in Section 3.3 by conditions in Equations (50) and (51), these local minima points are associated with the instants that the control system has to move the robot in opposite direction of the target position to ensure that the position and velocity constraints are not violated by the robot motion.



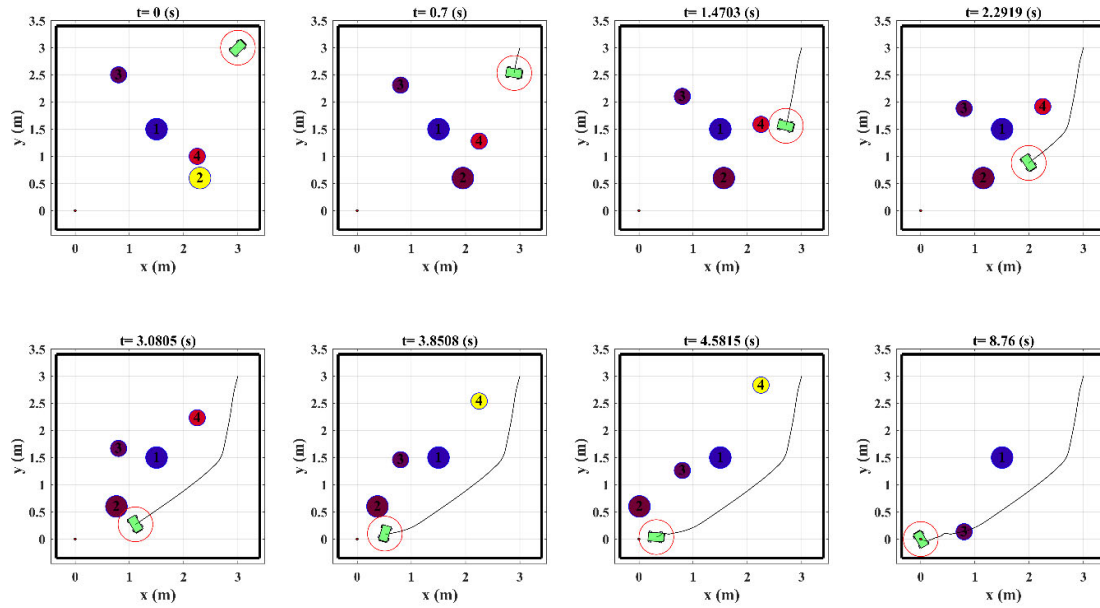
**Figure 13.** Optimal cost function value in example 1 during the robot's motion.



**Figure 14.** Optimal cost function value in example 2 during the robot's motion.

It follows from the analyses presented in Section 3.4 that two parameters  $n$  and  $\Delta$  can have major effects on the shape of the robot's trajectory before it reaches the vicinity of the target position. Parameter  $\Delta$  affects the maximum distance that the robot could travel in each sampling period and has a direct relationship with the safe region  $R_{safe}$ . On the other hand, parameter  $n$  affects the value of  $\delta$  which in turn has a main effect on the number of detected and active obstacles in each time instant. Therefore, increasing or decreasing the values of these parameters could change the route that the control system selects in each sampling time to move the robot toward the goal and prevent collision with obstacles.

To evaluate the effects of these parameters on the performance of the design control system, solving examples 1 and 2 are repeated using different values of  $\Delta$  and  $n$ . Figure 15 illustrates the snapshot of the robot's motion in example 1 for  $\Delta = 0.1\text{ s}$  and  $n = 4$ .



**Figure 15.** The snapshot of the robot's motion in example 1 for  $\Delta = 0.1\text{ s}$  and  $\delta = 2\text{ m}$ .

By increasing the value of  $\delta$ , the far obstacles are also detected, which has significant influences on the geometry of robot path toward the target position. Although increasing the value of  $\delta$  in example 1 helps the control system to consider the effects of far obstacles and find a safer path toward the goal, this may impose more restriction on the robot motion and reduce the robot maneuverability in more crowded environments.

The simulation results of example 2 for  $\Delta = 0.05\text{ s}$  and  $n = 1.4$  is shown in Figure 16. By reducing the value of  $\Delta$ , the radius of the safe region decreases to  $R_{\text{safe}} = 0.07\text{ m}$  and the robot could move closer to obstacles. Furthermore, using the smaller value for  $\delta$  leads to neglect the effects of far obstacles and reduce the number of velocity constraints on the robot motion. Compared to the simulation results illustrated in Figure 12, Figure 16 shows that the only effects of closer obstacles are considered. Therefore, the robot gets closer to obstacles before it changes its direction to avoid collision.



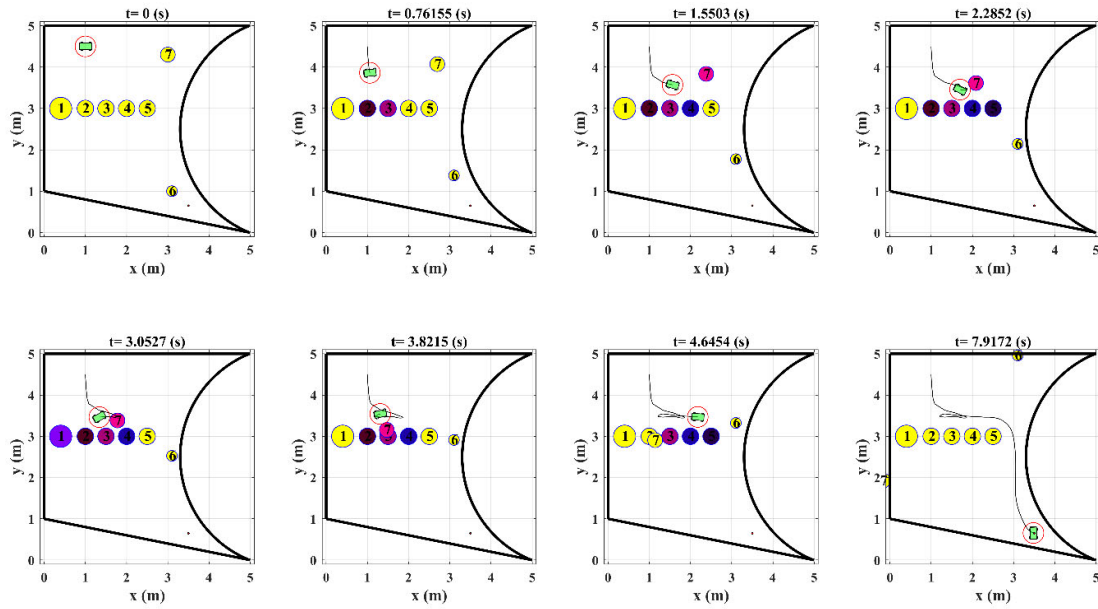


Figure 16. The snapshot of the robot's motion in example 1 for  $\Delta = 0.05\text{ s}$  and  $\delta = 0.6\text{ m}$ .

Figure 17 shows the simulation results of example 1 for  $\Delta = 0.03\text{ s}$  and  $n = 1$ . Using this value for  $\Delta$ , the radius of the safe region is  $R_{safe} = .042\text{ m}$  and therefore the robot could move in narrow passages and much closer to workspace borders. In addition, using the minimum allowable value for  $\delta$ , the robot travels longer distance on straight line trajectory than shown in Figure 9, until it detects the obstacle 1 for the first time. In comparison to Figures 9 and 15, Figure 17 shows that by using the smaller value for  $R_{safe}$  and  $\delta$ , the control system chooses the narrow space between the obstacles 1 and 4 for the robot's motion and the robot travel on a shorter path to reach the goal.

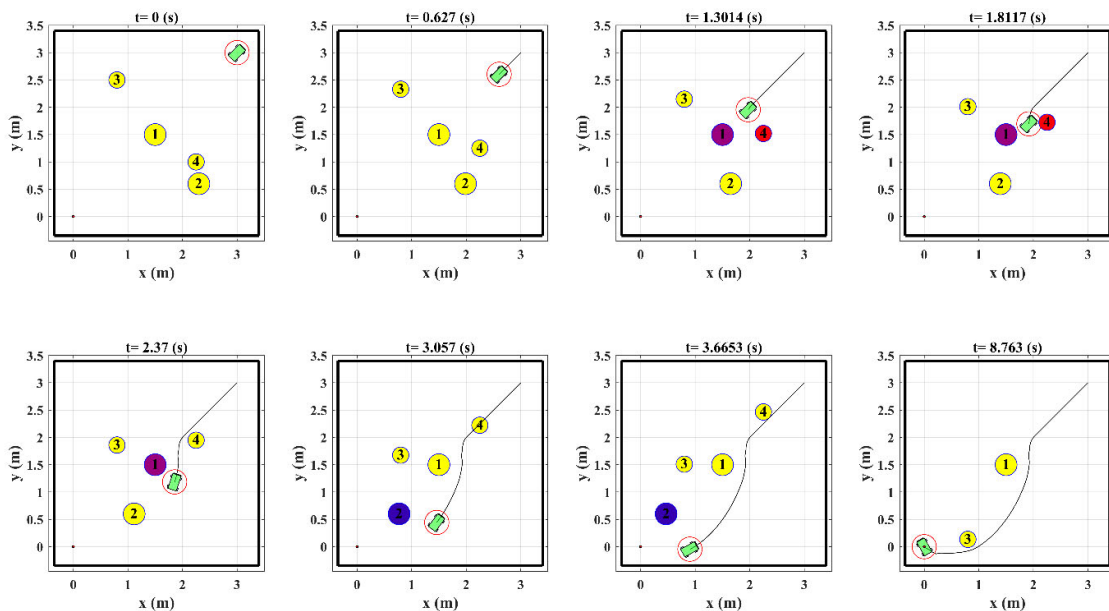


Figure 17. The snapshot of the robot's motion in example 1 for  $\Delta = 0.03\text{ s}$  and  $\delta = 0.4\text{ m}$ .

The effects of the parameters  $n$  and  $\Delta$  on the shape of the robot trajectory in the above simulations results are summarized in Table 3. These effects are evaluated and compared based on two quantitative and two qualitative metrics. The quantitative metrics are the length of trajectory and the total time required for the robot to reach to the target position. On the other hand, the qualitative metrics are using safer spaces for the robot's motion and the ability of moving in narrow spaces between obstacles. While the safer spaces are defined as part of workspace with low density of obstacles, the ability of moving in narrow spaces could be regarded as the capability of selecting narrower spaces between the obstacles to move the robot on faster trajectory or on closer to straight line trajectory.

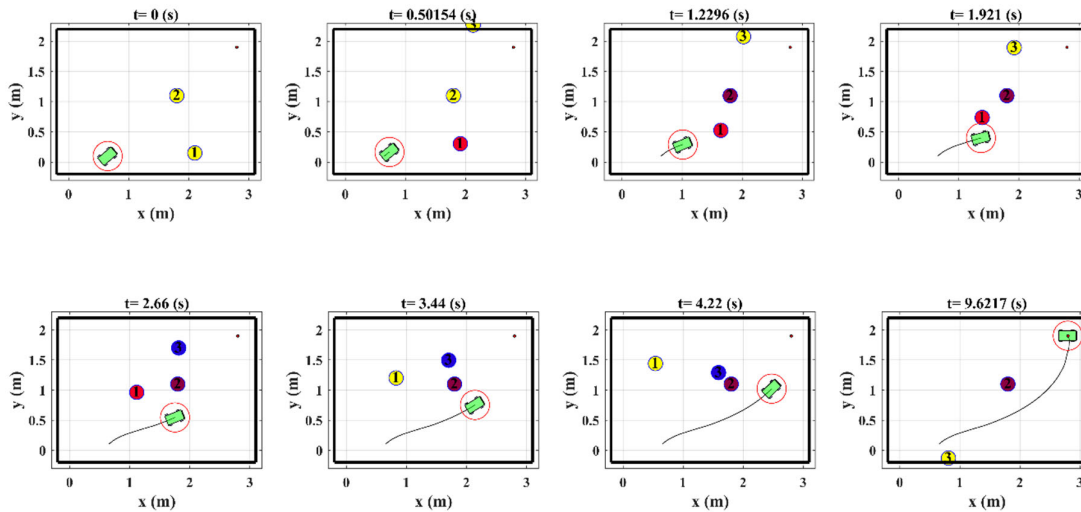
**Table 3.** The effects of parameters  $n$  and  $\Delta$  on the robot's trajectory.

Example No.	Parameters	Trajectory Length	Time of Motion	Use Safe Spaces	Use Narrow Spaces
1	$\Delta = 0.1, n = 2$	5.21 m	7 s	×	×
2	$\Delta = 0.1, n = 2$	6.034 m	9 s	√	×
1	$\Delta = 0.1, n = 4$	4.71 m	6.5 s	√	×
2	$\Delta = 0.05, n = 1.4$	6.77 m	8.5 s	×	×
1	$\Delta = 0.03, n = 1$	4.8 m	5 s	×	√

It is clear from Table 3 that by increasing the value of  $n$  the number of detected obstacles increases and by considering the effects of far obstacles, the control system could detect empty spaces in the workspace for the robot's motion. On the other hand, by decreasing the value of  $n$ , the control system neglects the effects of far obstacles and consequently tries to move the robot faster toward the target. Furthermore, by decreasing the value of  $\Delta$ , the value of  $R_{safe}$  also decreases and the control system could select narrow spaces between obstacles to move the robot toward the goal. As demonstrated by the data presented in Table 3, the fastest trajectory is obtained using the minimum value of  $n$  and  $\Delta$ . These values are related to simulation results presented in Figure 17 in which the control system not only neglect the effects of far obstacles in each sampling interval, but also use the narrow space between obstacles 1 and 4 for the robot motion. It is also clear from Table 3 that the safer trajectories in examples 1 and 2 are obtained using the larger value of  $n$ . Nevertheless, increasing the value of  $n$  leads to increase the number of imposed velocity constraints on the robot's motion and may have negative effects in crowded environments. Therefore, the appropriate values of  $n$  and  $\Delta$  could be selected based on the general layout of obstacles and other important factors in each circumstances.

To assess the performance of the proposed control system and obstacle avoidance strategy in comparison with other methods, the robot's motion between fixed and moving obstacles is simulated for the example 3 presented in [26]. The velocity and initial position of the obstacles are specified based on the data provided in [26]. In addition, the maximum allowable motors' torque and the value of the viscous friction coefficient are specified as  $u_{max} = 0.211 \text{ N.m}$  and  $\mu_f = 0.0238$  in such a way that the maximum acceleration and velocity of the robot are equal to the values reported in [26] ( $a_{max} = 1.2 \frac{m}{s^2}$  and  $v_{max}^r = 0.62 \frac{m}{s}$ ). In addition, considering  $\Delta = 0.1$ , the value of parameters  $R_{safe}$ ,  $t_m$  and  $d_{stop}^r$  are calculated as 0.062 m, 1.57 s and 0.1374 m, respectively. Since the maximum obstacle's speed in this example was  $0.48 \frac{m}{s}$ , the minimum value of  $\delta$  is calculated from Equation (56) as  $\delta \geq 0.953 \text{ m}$ . Using the initial and desired configuration of the robot, similar to [26], as  $\mathbf{X}_0 = [0.65, 0.1, 0.697, 0, 0, 0]^T$  and  $\mathbf{X}_d = [2.8, 1.9, 0, 0, 0, 0]^T$ , the

snapshot of the robot's motion among the obstacles is obtained by the proposed method in this paper and depicted in Figure 18.



**Figure 18.** The snapshot of the robot trajectory in example 3 given in [26].

Figure 18 shows that the robot successfully reaches the target configuration without collision with obstacles. Compared to the results of simulation provided in [26], the robot selects a safer trajectory toward the goal without facing the obstacle 3. This is because the NMPC system that chooses the fastest route toward the target after the robot turns right to escape from the obstacle 1. The simulation results presented in [26] demonstrate that the robot restricted to move on straight line trajectory between the start and the target position, and changed its direction once the obstacle detected. However, such a trajectory is not necessarily the best and the safest one. In addition, it may be even impossible for the robot to move on the trajectory calculated by this algorithm in complicated situations such as example 2 that the robot should stop in some positions and wait for the route to be opened.

## 5. Conclusions

In this paper, modeling, control and motion planning of a four wheel OMR has been investigated in dynamic environments. The robot moves using four Mecanum wheels and has the capability to rotate around itself and to move in every direction from every initial configuration. The kinematic model of the robot has been extracted and the differential equations of motion of the robot have been derived using Kane's method. To stabilize the robot in desired position and orientation, NMPC system has been designed and the problem of online trajectory planning of the robot in the presence of fixed and moving obstacles has been considered.

To plan the robot trajectory in dynamic environments online, the VO approach has been reformulated and introduced in the resulting optimization problem of NMPC system. The parameters and ingredients of the control system and the online trajectory planning algorithm have been specified through special stability and performance analysis of the closed loop system. Some mathematical relationships and criteria have been established between the control system parameters and the robot's physical and dynamical parameters to guarantee the effectiveness of the presented controller and motion planning algorithm. The stability of the proposed method as well as the performance of the online collision avoidance algorithm have been assessed through a series of computer simulations.

The results of simulations show that by selecting the values of the control system parameters based on the criteria obtained through stability and performance analysis, the

proposed method has acceptable results in stabilizing the OMR in desired configuration and online trajectory generation in dynamic environments. The simulation results also demonstrate that using the designed controller, the robot is able to perform acceptable maneuvers when the robot encircled by the obstacles in a small region and even when all the possible paths towards the target are temporarily blocked. Furthermore, the results show that by predicting the future state variables of the robot and future positions of active obstacles in NMPC, the motion planning algorithm performs more efficiently when the cost function has some local minima due to the special arrangement of the obstacles.

It is also worth mentioning that one of the main limitations of the proposed controller is associated with the computational time of NMPC system which is an important factor in real world implementation of the proposed method. This issue could be addressed using new proposed reinforcement learning algorithm for prediction. Therefore, evaluating the efficacy of the proposed method would be the subject of our future research.

**Author Contributions:** Conceptualization, A.R. and M.R.A.; methodology, A.R. and M.R.A.; software, M.R.A.; validation, A.R. and M.R.A.; formal analysis, A.R., M.R.A. and R.S.; investigation, A.R.; resources, R.S.; data curation, M.R.A.; writing—original draft preparation, A.R., M.R.A. and R.S.; writing—review and editing, A.R., M.R.A. and R.S.; visualization, M.R.A.; supervision, A.R. and R.S.; project administration, A.R.; funding acquisition, R.S. All authors have read and agreed to the published version of the manuscript.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Funding:** This research was conducted as part of the project called “Reuse and Recycling of Lithium-Ion Batteries” (RELIB). This work was supported by the Faraday Institution (grant number FIRG005).

**Data Availability Statement:** The software code and the data that support the findings of this study are openly available in Figshare with DOI (10.6084/m9.figshare.13293404), <https://figshare.com/s/20d9d42eb384b70ecbdb> (accessed on 17 March 2021).

**Conflicts of Interest:** The Authors declare that there is no conflict of interest.

## References

1. Doroftei, I.; Grosu, V.; Spinu, V. Omnidirectional mobile robot-design and implementation. In *Bioinspiration and Robotics*; Habib, M.K., Eds.; INTECH Open Access Publisher: Rijeka, Croatia, 2007; pp. 511–528.
2. Qian, J.; Zi, B.; Wang, D.; Ma, Y.; Zhang, D. The Design and Development of an Omni-Directional Mobile Robot Oriented to an Intelligent Manufacturing System. *Sensors* **2017**, *17*, 2073, doi:10.3390/s17092073.
3. Adăscăliței, F.; Doroftei, I. Practical applications for mobile robots based on mecanum wheels—a systematic survey. *Rom. Rev. Precis. Mech. Opt. Mechatronics*. **2011**, *40*, 21–29.
4. Tatar, M.O.; Popovici, C.; Mandru, D.; Ardelean, I.; Plesa, A. Design and development of an autonomous omni-directional mobile robot with Mecanum wheels. In Proceedings of the 2014 IEEE International Conference on Automation, Quality and Testing, Robotics, Cluj-Napoca, Romania, 22–24 May 2014; pp. 1–6.
5. Sheikhlari, A.; Fakharian, A. Adaptive optimal control via reinforcement learning for omni-directional wheeled robots. In Proceedings of the 2016 4th International Conference on Control, Instrumentation, and Automation (ICCIA), Qazvin, Iran, 27–28 January 2016; pp. 208–213.
6. Peng, T.; Qian, J.; Zi, B.; Liu, J.; Wang, X. Mechanical Design and Control System of an Omni-directional Mobile Robot for Material Conveying. *Procedia CIRP* **2016**, *56*, 412–415, doi:10.1016/j.procir.2016.10.068.
7. Kim, H.; Kim, B.K. Minimum-Energy Trajectory Generation for Cornering with a Fixed Heading for Three-Wheeled Omni-Directional Mobile Robots. *J. Intell. Robot. Syst.* **2013**, *75*, 205–221, doi:10.1007/s10846-013-9855-1.
8. Fu, X.; Wang, S.; Yang, J.; Wang, Y.; Liu, Z. Adaptive Sliding Mode Control for Omnidirectional Mobile Robot Based on a New Friction Modeling. In Proceedings of the 2017 International Conference on Computer Technology, Electronics and Communication (ICCTEC), Dalian, China, 19–21 December 2017; pp. 1042–1047.
9. Santos, J.; Conceição, A.G.; Santos, T.L. Trajectory tracking of Omni-directional Mobile Robots via Predictive Control plus a Filtered Smith Predictor. *IFAC Pap. Online* **2017**, *50*, 10250–10255, doi:10.1016/j.ifacol.2017.08.1286.
10. Huang, Y.; Cao, Q.; Leng, C. The path-tracking controller based on dynamic model with slip for one four-wheeled OMR. *Ind. Robot. Int. J.* **2010**, *37*, 193–201, doi:10.1108/01439911011018975.
11. Ilon, B.E. Rad Fuer Ein Laufstabiles, Selbstfahrendes Fahrzeug. DE 2354404, 16 May 1974.

12. Zimmermann, K.; Zeidis, I.; Abdelrahman, M. Dynamics of Mechanical Systems with Mecanum Wheels. In *Applied Non-Linear Dynamical Systems*; Springer Proceedings in Mathematics & Statistics; Awrejcewicz, J., Eds.; Springer: Cham, Switzerland, 2014; doi:10.1007/978-3-319-08266-0\_19.
13. Xie, L.; Henkel, C.; Stol, K.; Xu, W. Power-minimization and energy-reduction autonomous navigation of an omnidirectional Mecanum robot via the dynamic window approach local trajectory planning. *Int. J. Adv. Robot. Syst.* **2018**, *15*, 1–12; doi:10.1177/1729881418754563.
14. Yuan, Z.; Tian, Y.; Yin, Y.; Wang, S.; Liu, J.; Wu, L. Trajectory tracking control of a four mecanum wheeled mobile platform: An extended state observer-based sliding mode approach. *IET Control. Theory Appl.* **2020**, *14*, 415–426; doi:10.1049/iet-cta.2018.6127.
15. Lu, X.; Zhang, X.; Zhang, G.; Jia, S. Design of Adaptive Sliding Mode Controller for Four-Mecanum Wheel Mobile Robot. In Proceedings of the 2018 37th Chinese Control Conference (CCC), Wuhan, China, 25–27 July 2018; pp. 3983–3987.
16. Sun, Z.; Xie, H.; Zheng, J.; Man, Z.; He, D. Path-following control of Mecanum-wheels omnidirectional mobile robots using nonsingular terminal sliding mode. *Mech. Syst. Signal. Process.* **2021**, *147*, 107128; doi:10.1016/j.ymssp.2020.107128.
17. Lu, X.; Zhang, X.; Zhang, G.; Fan, J.; Jia, S. Neural network adaptive sliding mode control for omnidirectional vehicle with uncertainties. *ISA Trans.* **2019**, *86*, 201–214; doi:10.1016/j.isatra.2018.10.043.
18. Conceicao, A.S.; Oliveira, H.P.; Silva, A.S.; Oliveira, D.; Moreira, A.P. A Nonlinear Model Predictive Control of an Omni-Directional Mobile Robot. In Proceedings of the 2007 IEEE International Symposium on Industrial Electronics, Vigo, Spain, 4–7 June 2007; pp. 2161–2166.
19. Han, Y.; Zhu, Q. Robust Optimal Control of Omni-directional Mobile Robot using Model Predictive Control Method. In Proceedings of the 2019 Chinese Control Conference (CCC), Guangzhou, China, 27–30 July 2019; pp. 4679–4684.
20. Wang, D.; Wei, W.; Yeboah, Y.; Li, Y.; Gao, Y. A Robust Model Predictive Control Strategy for Trajectory Tracking of Omnidirectional Mobile Robots. *J. Intell. Robot. Syst.* **2019**, *98*, 439–453; doi:10.1007/s10846-019-01083-1.
21. Mohanan, M.G.; Salgoankar, A. A survey of robotic motion planning in dynamic environments. *Robot. Auton. Syst.* **2018**, *100*, 171–185; doi:10.1016/j.robot.2017.10.011.
22. López, J.; Sanchez-Vilariño, P.; Cacho, M.D.; Guillén, E.L. Obstacle avoidance in dynamic environments based on velocity space optimization. *Robot. Auton. Syst.* **2020**, *131*, 103569; doi:10.1016/j.robot.2020.103569.
23. Lashkari, N.; Biglarbegian, M.; Yang, S.X. Development of Novel Motion Planning and Controls for a Series of Physically Connected Robots in Dynamic Environments. *J. Intell. Robot. Syst.* **2018**, *95*, 291–310; doi:10.1007/s10846-018-0900-y.
24. Zavlangas, P.G.; Tzafestas, S.G. Motion control for mobile robot obstacle avoidance and navigation: A fuzzy logic-based approach. *Syst. Anal. Model. Simul.* **2003**, *43*, 1625–1637.
25. Leng, C.; Cao, Q.; Huang, Y. A motion planning method for omnidirectional mobile robot based on the anisotropic characteristics. *Int. J. Adv. Robot. Syst.* **2008**, *5*, 327–340.
26. Williams, R.L.; Wu, J. Dynamic Obstacle Avoidance for an Omnidirectional Mobile Robot. *J. Robot.* **2010**, *22*, 1–14.
27. Azizi, M.R.; Keighobadi, J. Point stabilization of nonholonomic spherical mobile robot using nonlinear model predictive control. *Robot. Auton. Syst.* **2017**, *98*, 347–359; doi:10.1016/j.robot.2017.09.015.
28. Cui, D.; Li, H. Model Predictive Control of Nonholonomic Mobile Robots with Backward Motion. *IFAC Pap. Online* **2019**, *52*, 195–200; doi:10.1016/j.ifacol.2019.12.407.
29. Mehrez, M.W.; Worthmann, K.; Cenerini, J.P.; Osman, M.; Melek, W.W.; Jeon, S. Model Predictive Control without terminal constraints or costs for holonomic mobile robots. *Robot. Auton. Syst.* **2020**, *127*, 103468; doi:10.1016/j.robot.2020.103468.
30. Killpack, M.D.; Kapusta, A.; Kemp, C.C. Model predictive control for fast reaching in clutter. *Auton. Robot.* **2015**, *40*, 537–560; doi:10.1007/s10514-015-9492-6.
31. Fiorini, P.; Shiller, Z. Motion Planning in Dynamic Environments Using Velocity Obstacles. *Int. J. Robot. Res.* **1998**, *17*, 760–772; doi:10.1177/027836499801700706.
32. Kane, T.R.; Levinson, D.A. *Dynamics, Theory and Applications*; McGraw Hill: New York, NY, USA, 1985.
33. Gyenes, Z.; Szadeczky-Kardoss, E.G. Motion planning for mobile robots using the safety velocity obstacles method. In Proceedings of the 2018 19th International Carpathian Control Conference (ICCC), Szilvásvárad, Hungary, 28–31 May 2018; pp. 389–394.
34. Shaobo, W.; Yingjun, Z.; Lianbo, L. A collision avoidance decision-making system for autonomous ship based on modified velocity obstacle method. *Ocean. Eng.* **2020**, *215*, 107910; doi:10.1016/j.oceaneng.2020.107910.
35. Fuad, M.; Agustinah, T.; Purwanto, D. Autonomous Indoor Vehicle Navigation Using Modified Steering Velocity Obstacles. In Proceedings of the 2020 International Seminar on Intelligent Technology and Its Applications (ISITIA), Surabaya, Indonesia, 22–23 July 2020; pp. 83–88.
36. Jeon, J.; Lee, B. Ellipse-based velocity obstacles for local navigation of holonomic mobile robot. *Electron. Lett.* **2014**, *50*, 1279–1281; doi:10.1049/el.2014.1592.
37. Mayne, D.; Rawlings, J.; Rao, C.; Scokaert, P. Constrained model predictive control: Stability and optimality. *Automatica* **2000**, *36*, 789–814; doi:10.1016/s0005-1098(99)00214-9.
38. Grüne, L.; Pannek, J. *Nonlinear Model Predictive Control: Theory and Algorithms*; Springer: London, UK, 2011.