RESEARCH

# Constraints and Model Predictive control of wheeled mobile robot in the presence of obstacles

Younis A Sabawi

**Sabawi YA. Constraints and Model Predictive control of wheeled mobile robot in the presence of obstacles. J Pure Appl Math. 2023; 7(5):312-325.**

## ABSTRACT

This work considers the problem of formation control for a team of nonholonomic wheeled mobile robots in an obstacle environment. Virtual structure formation approach and APF method are obtained to determine the reference trajectories of the robots. The virtual structure formation control strategy is obtained here to compute each individual robot trajectory whereas the reference path of the virtual center of the formation is generated by artificial potential fields. Many control strategies are discussed here. Next, a new control algorithm based on model predictive control and the nonlinear dynamics of the system is presented here. The control algorithm is applied to the nonlinear system using three different controllers traditional Model Predictive Control (MPC), Laguerre based MPC and nonlinear MPC. Using model predictive control, the dynamic model of the wheeled mobile robot is used to calculate the torques required to track the trajectory. The optimal solution of both formation and tracking problems is derived using a proposed control law. The control law could use the information from other robots to maintain the formation shape which can be guaranteed by using a new cost function. Simulations were accomplished using Matlab. The results demonstrate that the proposed controllers can achieve the trajectory tracking and formation keeping objectives also in the present of disturbance.

**Key words:** *Wheeled mobile robots; Lagrange multiplier method; Obstacles; Feedback linearization; Predictive control*



## INTRODUCTION

In numerous researches in the area of formation control of wheeled mobile robots have been accomplished in the recent years. Cooperative robotics was first proposed to the modern engineering research in the early 1980s. There are three main strategies to control cooperative robotic systems in the previous studies. In this section, an overview of the related works using these three different approaches of formation control will be briefly presented [1].

Behavior based approach starts by designing simple behaviors or motion primitives for each individual robot, such as, formation keeping, trajectory tracking, collision evasion, goal seeking, and obstacle avoidance. Then, more complex motion patterns can be generated by using a weighted sum of the relative importance of these primitives and the interaction of several robots [2]. In the behaviors of a group of objects that exhibit was studied [3]. It is assumed that a folk is the result of the interaction between the behavior of individual creatures. Behaviors were represented as rules or programs where a data structure held the internal state of each member of the formation. The essential features of these behaviors were expressed as an object. Each object needs a computational process to apply the behavioral programs to the internal data [4].

In [5] a new class of potential functions was presented to allow each robot to organize itself and take its location in the geometric formation while moving to the goal in an obstacle environment. A robust formation control algorithm was applied in [6] by keeping the distance and angle between each robot and its leader approximately constant. The algorithm uses only the relative positions between neighboring robots and obstacles. The idea of non-linear attractor dynamics and behavior generating are used to design controllers allow the team of robots to move considering a predefined formation [7]. A new kinematics model for the leader-follower system using Cartesian coordinates instead of the generally used polar coordinates was presented in [8], where an integrator back stepping based controller was obtained. Desai et al. also suggested an algorithm that enables a group of robots to change its formation while avoiding obstacles [9]. A combination of kinematic and dynamic controllers was presented [10]. The reference velocity generated by the kinematic controller was the command input to a sliding mode controller based on the dynamic system. A distributed formation control algorithm in [11] was proposed assuming that the full state of the leader is unknown by

*Department of Mathematics, Faculty of Science and Health, Koya University, Koya KOY45, Kurdistan Region-F. R. Iraq,*

*Correspondence: Younis A Sabawi, Department of Mathematics, Faculty of Science and Health, Koya University, Koya KOY45, Kurdistan Region-F. R. Iraq, e-mail: sotiris.spanogianopoulos@gmail.com*

*Received: 29 Sep, 2023, Manuscript No. puljpam-23-6740, Editor Assigned: 1 Oct, 2023, PreQC No. puljpam-23-6740 (PQ), Reviewed: 7 Oct, 2023, QC No. puljpam-23-6740 (Q), Revised: 12 Oct,2023, Manuscript No. puljpam-23-6740 (R), Published: 31 Oct, 2023, DOI:-10.37532/2752-8081.23.7(5).312-325*

follower robots. The control law was designed based on the formation distance error and the estimated states of the leader. Moreover, paper suggested a bearing-only leaderfollower formation control approach where the leader robot is observed by the followers [12]. In addition, Kalman filter was obtained to estimate the states of the system. Finally, an input-output feedback control was implemented which guarantees formation maintaining and stable movement for follower robots. Three level hybrid control algorithm to obtain both centralized and decentralized cooperative control via leader-follower method were proposed [13]. The global-level formation control was converted into decentralized control problems between N-1 followers and their leader. Two basic controllers were presented in the leader-follower control level to keep relative distance between the follower and the leader and avoid obstacles. In [14] translated the formation control problem from the configuration space into the image plan for each follower. Also, a nonlinear tracking control was designed to compensate the nonlinearity of the robot dynamics. In [15] a robust leader-follower formation controller consisting of a feedback linearization part and a sliding mode compensator is designed. In [16] a control law was designed based on faulttolerant which considered certain conditions including sensor limitations, obstacle and neighbors' collision avoidance, and leader-loss situation. This Control method is more applicable than other methods because it assumed that global states are not measurable and the communications among robots are not available. In [17] used a receding-horizon leader-follower control algorithm to maintain the formation of a group of mobile robots. To keep the desired relation between leader and followers a Separation–Bearing– Orientation Scheme (SBOS) and Separation–Separation–Orientation Scheme (SSOS). Unlike other approaches, Chen solved the formation control problem when robots move backwards. A modern control algorithm has succeeded in formation control is the Model Predictive Control (MPC) because it is able to solve the control problem considering constraints on control, states and output variables [18]. In [19] a distributed nonlinear model predictive control of constrained agents was presented. Paper proposed a control law contains a combination of linear model predictive control and input-output linearization [20]. Using feedback linearization canceled the nonlinearity of the dynamic system. Furthermore, MPC was implemented to find the optimal solution for the leader-follower formation control problem. Kuwata et al. also proposed a new distributed robust linear MPC to control a multivehicle system [21]. In contrast, a distributed receding horizon control of a platoon of vehicles with nonlinear discrete-time dynamics was presented in [22].

Given the above, most of the proposed strategies for formation control based on the kinematic equation of the wheeled mobile robot without taking into account the dynamic equations. Moreover, many of these control strategies ignored the nonlinearity of the system and were designed using only the linearized model of the system. Nevertheless, it is important to design the controller via the nonlinear dynamics to reach a certain level of stability.

In this work, we will discuss the problem of formation control of three wheeled robots using the virtual structure approach. In other words, the problem here is a trajectory tracking for a system of three wheeled mobile robots while maintaining a desired formation and avoiding obstacles. The virtual structure formation control strategy is obtained herein to compute each individual robot trajectory where the reference path of the virtual center of the formation is generated by artificial potential fields. Also, we will present two different linear model predictive controllers and a nonlinear model predictive controller. Formation keeping is guaranteed by a proposed cost function considers the formation error between each two robots. These three methods will be compared with a control system based on feedback linearization controller proposed in [23].

The thesis is organized as follows. Section 2 defines both the kinematic and dynamic model for a single differential drive mobile robot. Also, the linearized model of the dynamic equations is extracted in this chapter. The path planning problem is considered in chapter 3. The artificial potential field is used to generate the desired trajectory for the virtual center. Later in section 3, the virtual structure formation approach is expressed, where the trajectory of each single robot is derived. Section 4 discusses the control strategy used in reference and presents a new controller consists of afeedback linearization [24]. Later in section 4 a stability study of the proposed controller is presented followed by the simulation results and comparison with the controller in [24]. The design of three different Model Predictive Controllers (MPC) is discussed in Section 5, nonlinear MPC. This chapter also contains the simulation results of the proposed MPC controllers. A comparison between the NMPC and feedback linearization in the presence of noise also presented in section 5. Finally, the experiments results and the conclusion are reported respectively.

**Kinematic model**

Each robot of the team is considered to be a differential drive wheeled mobile robot. The team consists of N identical robot. Therefore, the static and dynamic model will be derived only for one robot. The differential drive wheeled mobile robot consists of two driven wheels mounted on a common axis, and each wheel can independently be driven forward and backward. To have the robot to perform a rolling motion, the velocity of each wheel must be different from each other making the robot rotate around the midpoint of the common axis. By varying the velocities of the two wheels, the robot will be able to follow the desired path.

Nonholonomic wheeled robot model: Assume that the robot can only move along the vehicle longitudinal axis and the wheels roll without slipping in the horizontal plane while keeping the sagittal plane (i.e. the plane that contain the wheel disk) in the vertical direction. By considering these assumptions, the robot will have a kinematic constraint called nonholonomic constraint. To illustrate this fact, consider single Pfaffian constraints [25]:

$$a^T(q)\dot{q} = 0 \tag{1}$$

If the constraint is holonomic, it can be integrated and written as

$$h(q) = c \tag{2}$$

Where $\frac{\partial h}{\partial q} = a^T(q)$ and c is an integration constant. Therefore, there is a loss of *accessibility* in the configuration space, because the motion of the mechanical system in C is confined to a particular *level surface* of the scalar function *h*. This surface, which depends on the initial configuration $q_0$ through the value of $h(q_0) = c$, has dimension *n-1*.

Instead, if the constraints (2) is nonholonomic, the generalized velocities are indeed constrained to belong to a subspace of dimension *n-1*, i.e., the null space of matrix $a^T(q)$. The fact that the constraint is non-integrable means that there is no loss of accessibility in C for the system. In other words, while the number of DOFs decreases to n-1 due to the constraint, the number of generalized coordinates cannot be reduced, not even locally.

Let us get back to our problem, the pure rolling constraint for the ith robot is as follows

$$\dot{x}_A{}^i \sin\varphi^i - \dot{y}_A{}^i \cos\varphi^i = 0 \tag{3}$$

in which means

$$\tan(\varphi^i) = \frac{\dot{y}_A{}^i}{\dot{x}_A{}^i} \tag{4}$$

and entails that, in the absence of slipping, the velocity of the contact point has zero component in the direction orthogonal to the sagittal plane. The angular velocity of the disk around the vertical axis instead is unconstrained.

Position of the center of mass can be written with respect to the midpoint of the wheel's common axis A as follows:

$$\begin{cases} x_c{}^i(t) = x_A{}^i + d\cos\varphi^i \\ y_c{}^i(t) = y_A{}^i + d\sin\varphi^i \\ \varphi^i(t) = \theta_A{}^i(t) \end{cases} \tag{5}$$

Rewriting the pure rolling and non-slipping constraint in term of $q$ by deriving equation (4) and substituting it in (2) yields to:

$$\left(\dot{x}_c{}^i + d\dot{\varphi}^i\sin\varphi^i\right)\sin\varphi^i - \left(\dot{y}_c{}^i + \dot{\varphi}^i d\cos\varphi^i\right)\cos\varphi^i = 0 \tag{6}$$

Simplifying equation (2-5) leads to:

$$\dot{x}_c{}^i\sin\varphi - \dot{y}_c{}^i\cos\varphi^i + \dot{\varphi}^i d = 0 \tag{7}$$

The above equation can be written in matrix form as:

$$A^i(q^i).\dot{q}^i = 0 \tag{8}$$

Where

$$A^i(q^i) = [\sin\varphi^i \quad \cos\varphi^i \quad d] \tag{9}$$

Each robot of the team is considered to be a differential drive wheeled mobile robot. The team consists of N identical robot. Therefore, the static and dynamic model will be derived only for one robot. The differential drive wheeled mobile robot consists of two driven wheels mounted on a common axis, and each wheel can independently be driven forward and backward, see figure (1). To have the robot to perform a rolling motion, the velocity of each wheel must be different from each other making the robot rotate around the midpoint of the common axis. By varying the velocities of the two wheels, the robot will be able to follow the desired path.

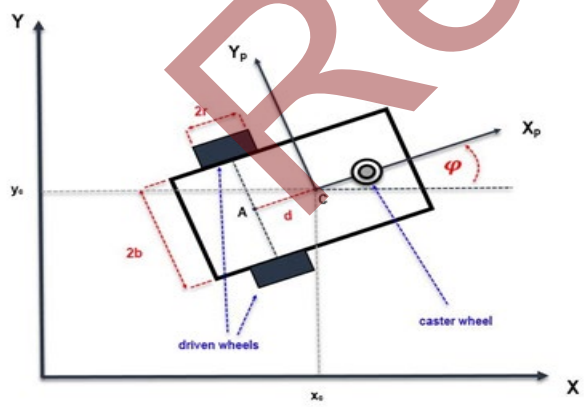



**Figure 1)** *The kinematic model of the differential drive mobile robot.*

Assume that the center of mass of the ith mobile robot is at some position $(x_c{}^i, y_c{}^i)$ and orientation $\varphi^i$ which can be expressed as a state vector $q^i(t) = [x_c{}^i(t) \quad y_c{}^i(t) \quad \varphi^i(t)]^T$ . Point A denotes the midpoint of the segment joining the two wheel centers.

where $\dot{x}_c{}^i, \dot{y}_c{}^i$ are the ith robot velocity components in the Cartesian plane and $\dot{\varphi}^i$ is the angular velocity. Therefore $\dot{q}^i = [\dot{x}_c{}^i, \dot{y}_c{}^i, \dot{\varphi}^i]^T$ is presented generalized velocities vector.

Generalized velocities are indeed constrained to belong to a subspace of dimension n−1, i.e., the null space of matrix $A^i(q^i)$. Nevertheless, the fact that the constraint is non-integrable means that there is no loss of accessibility in configuration of the system. In other words, while the number of DOFs decreases to n − 1 due to the constraint, the number of generalized coordinates cannot be reduced, not even locally. Therefore, kinematic model of the i[th] robot can be written as follows:

$$\dot{q}^i(t) = S^i(q^i)v^i \tag{10}$$

where $v^i = [\dot{\theta}_r{}^i, \dot{\theta}_l{}^i]^T$ and $\dot{\theta}_r{}^i, \dot{\theta}_l{}^i$ are the velocities of the right and left wheels, respectively.

To find the matrix $S^i(q^i)$ , Differentiating (1) with respect to time yields:

$$\begin{cases} \dot{x}_c{}^i(t) = v^i\cos\varphi^i - d\omega^i\sin\varphi^i \\ \dot{y}_c{}^i(t) = v^i\sin\varphi^i + d\omega^i\cos\varphi^i \\ \dot{\varphi}^i(t) = \omega^i \end{cases} \tag{11}$$

Rewriting equations (2) in matrix form yields:

$$\dot{q}_c{}^i(t) = \begin{bmatrix} \cos\varphi^i & -d\sin\varphi^i \\ \sin\varphi^i & +d\cos\varphi^i \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v^i \\ \omega^i \end{bmatrix} \tag{12}$$

where $v^i$ and $\omega^i$ are the linear and the angular velocities of the ith vehicle, respectively. Since the linear velocity of each wheel in the ith robot frame is $v_{r,l}{}^i = r\dot{\theta}_{r,l}{}^i$ , the linear and the angular velocities of the vehicle in the ith robot frame can be computed as:

$$v^i = \frac{v_r{}^i + v_l{}^i}{2} = r\frac{\dot{\theta}_r{}^i + \dot{\theta}_l{}^i}{2} \tag{13}$$

$$\omega^i = r\frac{\dot{\theta}_r{}^i - \dot{\theta}_l{}^i}{2b} \tag{14}$$

By substituting (1) and (2) in (4,5), the kinematic model of the ith wheeled mobile robot in (15) can be derived by the following equation:

$$\dot{q}^i(t) = \begin{bmatrix} \frac{r}{2b}(b\cos\varphi^i - d\sin\varphi^i) & \frac{r}{2b}(b\cos\varphi^i + d\sin\varphi^i) \\ \frac{r}{2b}(b\sin\varphi^i + d\cos\varphi^i) & \frac{r}{2b}(b\sin\varphi^i - ds\cos\varphi^i) \\ \frac{r}{2b} & -\frac{r}{2b} \end{bmatrix} \begin{bmatrix} \dot{\theta}_r{}^i \\ \dot{\theta}_l{}^i \end{bmatrix} \tag{15}$$

**Dynamic model**

In the following, the Lagrange formulation is used to obtain the dynamic model of the wheeled mobile robot system subject to the kinematic constraint in the form (16). The Lagrangian $\mathcal{L}$ of a mechanical system is defined as the difference between its kinetic and potential energy:

$$\mathcal{L}(q,\dot{q}) = T(q,\dot{q}) - U(q) \tag{16}$$

where T and U are the kinetic energy and the potential energy of the system, respectively. The Lagrange equations in this case are:

$$\frac{d}{dt}\left(\frac{\partial\mathcal{L}^i}{\partial\dot{q}_k{}^i}\right) - \frac{\partial\mathcal{L}^i}{\partial q_k{}^i} = B^i(q^i)\tau^i + A^i(q^i)^T\lambda^i \tag{17}$$

where $B^i(q^i)$ is t he inpu t tran sformation matr ix, $\boldsymbol{\tau^i} = [\tau_r{}^i \quad \tau_l{}^i]$ represents the actuating torques of the wheels and $\lambda^i$ is the vector of Lagrangian multipliers for the i[th] robot. The term $\mathbf{A}(\boldsymbol{q^i})^T\lambda^i$ represent the vector of reaction forces at the generalized coordinate level.

The kinetic energy of the ith robot:

$$T^i = \frac{1}{2}m\left(\dot{x}_C^{i\,2} + \dot{y}_C^{i\,2}\right) + \frac{1}{2}I\dot{\varphi}^{i\,2} \tag{18}$$

$$\therefore m = (m_p + 2m_w), I = I_{zp} + 2I_{zw} + 2m_w(d^2 + b^2)$$

where $m_p$ is the mass of each platform without the mass of driven wheels and actuators, $m_w$ is the mass of the wheels and their actuators and $I_{zp}$ , $I_{zw}$ are the moment of inertia of the wheels and the platform, respectively.

Since the potential energy of the system is equal to zero, the Lagrangian function is equal to the kinetic energy i.e. $\mathcal{L}^i(q^i, \dot{q}^i) = T^i(q^i, \dot{q}^i)$.

Obtaining equation (17) with the derived Lagrangian function gives

$$\begin{cases} \frac{d}{dt}\left(\frac{\partial \mathcal{L}^i}{\partial \dot{x}_C{}^i}\right) = \frac{d}{dt}\left(m\dot{x}_C{}^i\right) = m\ddot{x}_C{}^i, \frac{\partial \mathcal{L}^i}{\partial x_C{}^i} = 0 \\ \frac{d}{dt}\left(\frac{\partial \mathcal{L}^i}{\partial \dot{y}_C{}^i}\right) = \frac{d}{dt}\left(m\dot{y}_C{}^i\right) = m\ddot{y}_C{}^i, \frac{\partial \mathcal{L}^i}{\partial y_C{}^i} = 0 \\ \frac{d}{dt}\left(\frac{\partial \mathcal{L}^i}{\partial \dot{\varphi}^i}\right) = \frac{d}{dt}(I\dot{\varphi}^i) = I\ddot{\varphi}^i, \frac{\partial \mathcal{L}^i}{\partial \varphi^i} = 0 \end{cases} \tag{19}$$

This results in the following equations of motion:

$$\begin{cases} m\ddot{x}_C{}^i = F_x{}^i + C_1{}^i \\ m\ddot{y}_C{}^i = F_y{}^i + C_2{}^i \\ I\ddot{\varphi}^i = \tau^i + C_3{}^i \end{cases} \tag{20}$$

where $C_1{}^i$ , $C_2{}^i$ and $C_3{}^i$ are coefficients related to the kinetic constraints and $F_x{}^i, F_y{}^i, \tau^i$ are the actuator forces acting in x and y directions and actuator torque, respectively.

$$F_x{}^i = \left(F_R{}^i + F_L{}^i\right)\cos\varphi^i \tag{21}$$

$$F_y{}^i = \left(F_R{}^i + F_L{}^i\right)\sin\varphi^i \tag{22}$$

$$\tau^i = \left(F_R{}^i - F_L{}^i\right)b \tag{23}$$

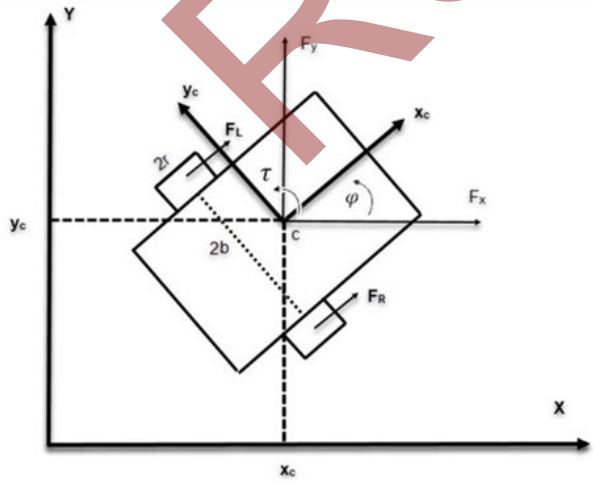



**Figure 2)** *Forces acted on one robot.*

where $F_R{}^i$ *and* $F_L{}^i$ are the forces exerted by the right and left wheels of the ith robot, respectively, see figure 2,

$$F_R{}^i = \frac{\tau_R{}^i}{r} \tag{24}$$

$$F_L{}^i = \frac{\tau_L{}^i}{r} \tag{25}$$

Substituting (24, 25) in equations (21) to (23),

$$\begin{bmatrix} F_x{}^i \\ F_y{}^i \\ \tau^i \end{bmatrix} = \frac{1}{r}\begin{bmatrix} \cos\varphi^i & \cos\varphi^i \\ \sin\varphi^i & \sin\varphi^i \\ b & -b \end{bmatrix}\begin{bmatrix} \tau_R{}^i \\ \tau_L{}^i \end{bmatrix} \tag{26}$$

Equation (20) can be written in matrix form as:

$$M^i(q^i)\ddot{q}^i + C^i(q^i, \dot{q}^i) + G^i(q^i) = B^i(q^i)\tau^i + A^{i^T}(q^i)\lambda^i \tag{27}$$

Where:

$$M^i(q^i) = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I \end{bmatrix}$$

$$C^i(q^i, \dot{q}^i) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$B^i(q^i) = \begin{bmatrix} \cos\varphi^i & \cos\varphi^i \\ \sin\varphi^i & \sin\varphi^i \\ b & -b \end{bmatrix}$$

$$A^{i^T}(q^i)\lambda^i = \begin{bmatrix} C_1{}^i \\ C_2{}^i \\ C_3{}^i \end{bmatrix}$$

It can be proved that the columns of $S^i(q^i)$ are a basis for the null space of $A^i(q^i)$, so that:

$$S^{i^T}(q^i)A^{i^T}(q^i) = 0 \tag{28}$$

Moreover, the Lagrange multiplier can be eliminated by pre-multiplying both sides of equation (27) by $S^{i^T}(q^i)$.

$$S^{i^T}(q^i)M^i(q^i)\ddot{q}^i + S^{i^T}(q^i)C^i(q^i, \dot{q}^i) + S^{i^T}(q^i)G^i(q^i) = S^{i^T}(q^i)B^i\tau^i + S^{i^T}(q^i)A^{i^T}(q^i)\lambda^i \tag{29}$$

Differentiating the kinematic equation (8) results in:

$$\ddot{q}^i = \dot{S}^i(q^i)v^i + S^i(q^i)\dot{v}^i \tag{30}$$

Substituting (28) and (30) in (29) yields to:

$$S^{i^T}(q)M^i\dot{S}^i(q^i)v^i + S^{i^T}(q^i)M^iS^i(q^i)\dot{v}^i + S^{i^T}(q^i)C^i = S^{i^T}(q^i)B^i \tag{31}$$

Finally, the reduced dynamic model is expressed as:

$$\bar{M}^i(q^i)\dot{v}^i + \bar{C}^i(q^i, \dot{q}^i) = \bar{B}^i(q^i)\tau^i \tag{32}$$

where:

$$\bar{M}^i(q^i) = S^{i^T}(q^i)M^iS^i(q^i)$$

$$\bar{C}^i(q^i, \dot{q}^i) = S^{i^T}(q^i)M^i\dot{S}^i(q^i)v^i + S^{i^T}(q^i)C^i$$

$$\bar{B}^i(q^i) = S^{i^T}(q^i)B^i$$

Now consider a new state vector for the system as:

$$x^i = \begin{bmatrix} x^i \\ y^i \\ \varphi^i \\ \dot{\theta}_R{}^i \\ \dot{\theta}_L{}^i \end{bmatrix} = \begin{bmatrix} q^i \\ v^i \end{bmatrix}, u^i = \tau^i \tag{33}$$

Therefore, the equation of motion in the state space form can be represented as

$$\dot{x}^i = f^i(x^i) + G^i(x^i)u^i = f_x{}^i(x^i, u^i) \tag{34}$$

where:

$$f^i(x^i) = \begin{bmatrix} S^i v^i \\ -\bar{M}^i(q^i)^{-1}\bar{C}^i(q^i, \dot{q}^i) \end{bmatrix}, G^i(x^i) = \begin{bmatrix} 0 \\ \bar{M}^i(q^i)^{-1}\bar{B}^i(q^i) \end{bmatrix}$$

The more detailed equations are expressed as follows:

$$f^i(x^i) = \begin{bmatrix} \frac{r}{2}\dot{\theta}_R{}^i\left(\cos\varphi^i - \frac{d}{b}\sin\varphi^i\right) + \frac{r}{2}\dot{\theta}_L{}^i\left(\cos\varphi^i + \frac{d}{b}\sin\varphi^i\right) \\ \frac{r}{2}\dot{\theta}_R{}^i\left(\sin\varphi^i + \frac{d}{b}\cos\varphi^i\right) + \frac{r}{2}\dot{\theta}_L{}^i\left(\sin\varphi^i - \frac{d}{b}\cos\varphi^i\right) \\ \frac{r}{2b}\dot{\theta}_R{}^i - \frac{r}{2b}\dot{\theta}_L{}^i \\ -\frac{d}{c}(I\dot{\theta}_L{}^i - I\dot{\theta}_R{}^i + b^2m\dot{\theta}_R{}^i + b^2m\dot{\theta}_L{}^i - d^2m\dot{\theta}_R{}^i + d^2m\dot{\theta}_L{}^i)\dot{\varphi}^i \\ \frac{d}{c}(-I\dot{\theta}_L{}^i + I\dot{\theta}_R{}^i + b^2m\dot{\theta}_R{}^i + b^2m\dot{\theta}_L{}^i + d^2m\dot{\theta}_R{}^i - d^2m\dot{\theta}_L{}^i)\dot{\varphi}^i \end{bmatrix} \tag{35}$$

Nonholonomic Wheeled Robot Model

$$G^i(x^i) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ c_1 & c_2 \\ c_2 & c_1 \end{bmatrix} \tag{36}$$

where c, $c_1$ and $c_2$ are constants defined by:

$$c = 2b(md^2 + I)$$

$$c_1 = \frac{mb^2 + md^2 + I}{mr^2(md^2 + I)}$$

$$c_2 = \frac{-mb^2 + md^2 + I}{mr^2(md^2 + I)}$$

**Linearized model**

A linear model can be derived from the nonlinear model (34) by linearizing around a reference trajectory. The reference trajectory can be applied from the desired path $(x_r{}^i(t), u_r{}^i(t))$ the ith robot should follow. To this end, the Taylor series expansion [37] of the system (34) is obtained around the trajectory and as a result:

$$\dot{x}^i = f_x{}^i(x_r{}^i, u_r{}^i) + \frac{\partial f_x{}^i(x^i,u^i)}{\partial x^i}\Big|_{\substack{x^i = x_r{}^i \\ u^i = u_r{}^i}}(x^i - x_r{}^i) + \frac{\partial f_x{}^i(x,u)}{\partial u^i}\Big|_{\substack{x^i = x_r{}^i \\ u^i = u_r{}^i}}(u^i - u_r{}^i) + \mathcal{F}^i \tag{37}$$

where $\mathcal{F}^i$ contains the high order terms of the expansion which can be ignored in linear approximations. Therefore, system (34) after the linearization around the nominal trajectory $x_r{}^i(t)$ is expressed as:

$$\dot{e}^i = A^i(t)e^i + B^i(t)e_u{}^i \tag{38}$$

where $e^i$ is the error vector of state variables, $e_u{}^i$ is the error vector of control variables respect to the reference trajectory $(x_r{}^i(t), u_r{}^i(t))$ and the matrices $A^i(t)$ and $B^i(t)$ are defined as:

$$A^i(t) = \frac{\partial f_x{}^i(x^i,u^i)}{\partial x^i}\Big|_{\substack{x^i = x_r{}^i \\ u^i = u_r{}^i}}, B(t) = \frac{\partial f_x{}^i(x,u)}{\partial u^i}\Big|_{\substack{x^i = x_r{}^i \\ u^i = u_r{}^i}} \tag{39}$$

As a result, we have:

$$A^i(t) = \begin{bmatrix} 0 & 0 & -\frac{r}{2}\dot{\theta}_R{}^i\left(\varphi^i + \frac{d}{b}\right) - \frac{r}{2}\dot{\theta}_L{}^i\left(\varphi^i - \frac{d}{b}\right) & \frac{r}{2}\left(1 - \frac{d}{b}\varphi^i\right) & \frac{r}{2}\left(1 + \frac{d}{b}\varphi^i\right) \\ 0 & 0 & \frac{r}{2}\dot{\theta}_R{}^i\left(1 - \frac{d}{b}\varphi^i\right) + \frac{r}{2}\dot{\theta}_L{}^i\left(1 + \frac{d}{b}\varphi^i\right) & \frac{r}{2}\left(\varphi^i + \frac{d}{b}\right) & \frac{r}{2} * \left(\varphi^i - \frac{d}{b}\right) \\ 0 & 0 & 0 & \frac{r}{2b} & -\frac{r}{2b} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$B^i(t) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ c_1 & c_2 \\ c_2 & c_1 \end{bmatrix}$$

Rewriting state space equations in discrete-time format might be needed in the design of the controller. To convert the continuous-time dynamic model to discrete-time formulation, Euler's method can give an approximate solution using the following equations :

$$A_d{}^i(k) = I_{n\times n} + T_s A_c{}^i(t)$$

$$B_d{}^i(k) = T_s B_c{}^i(t)$$

which is a good approximation for a short sampling time $T_s$. Matrices $\mathbf{A}_c{}^i$ and $\mathbf{B}_c{}^i$ are the continuous-time state space matrices, and for the system (38) are equal to matrices $\mathbf{A}^i$ and $\mathbf{B}^i$, respectively.

**Formation problem *via* virtual structure**

Assume that the formation contains $n$ identical nonholonomic wheeled mobile robots, and there is a virtual center for the group with configuration vector $q_{vc} = [x_{vc}, y_{vc}, \varphi_{vc}]^T$, which represents the desired position of the virtual center of the formation. The idea behind virtual center approach is that each robot has to keep specified distance from the virtual center of the formation. By the assumption that the virtual center acts like a real robot, that its center of mass is at the midpoint of the wheels' common axis, its kinematic model is presented in equation (41).

$$\begin{cases} \dot{x}_{vc}(t) = v_{vc}\cos\varphi_{vc} \\ \dot{y}_{vc}(t) = v_{vc}\sin\varphi_{vc} \\ \dot{\varphi}_{vc}(t) = \omega_{vc} \end{cases} \tag{40}$$

where $v_{vc}$, $\omega_{vc}$ are linear and angular velocities of the virtual center, respectively. By using the APF path generating method we can define the linear velocities and the orientation of the virtual center in fixed coordinate frame. Then, by integrating these velocities with respect to time we get the $x_{vc}$ $an$ $y_{vc}$. By using these information, the desired position for each robot can easily be derived as:

$$\begin{bmatrix} x_{di} \\ y_{di} \end{bmatrix} = \begin{bmatrix} x_{vc} \\ y_{vc} \end{bmatrix} + \begin{bmatrix} \cos\varphi_{vc} & -\sin\varphi_{vc} \\ \sin\varphi_{vc} & \cos\varphi_{vc} \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} \tag{41}$$

**Artificial potential fields**

The main idea of artificial potential field approaches is that an artificial field U is constructed in the work space based on the goal and obstacles configuration. The field is obtained by an attractive potential to the goal and repulsive potentials from the obstacles [15]. Moreover, the point that represents the robot moves in configuration space under the influence of the total potential U. To this end, at each robot configuration the potential creates an artificial force equals to the negative gradient of the total potential which is obtained as a velocity input to the dynamic system of the virtual center. Therefore, the artificial force specifies the direction and the velocity of the virtual center planned motion.

As mentioned earlier, the total potential field U is a combination of the attractive and the repulsive fields as follows:

$$U_{total}(q_{vc}) = U_{att}(q_{vc}) + U_{rep}(q_{vc}) \quad (42)$$

where $\boldsymbol{U}_{total}(\boldsymbol{q}_{vc})$ is the total potential field, $\boldsymbol{U}_{att}(\boldsymbol{q}_{vc})$ is the attractive field and $\boldsymbol{U}_{rep}(\boldsymbol{q}_{vc})$ is the repulsive field. Different functions have been discussed in related researches.

The method of path planning and obstacles avoidance was first proposed by [26]. In this reference, the attraction potential field is obtained as follows

$$U_{att} = \frac{1}{2}\xi {\rho_g}^2 \quad (43)$$

where $\xi$ is a positive scaling parameter, m = 2 and $\rho_g$ is the distance between the robot and the goal point. Considering $\boldsymbol{q}_{goal}$ is the goal configuration and $q$ represents the point of the robot, $\boldsymbol{\rho}_g$ can be expressed as:

$$\rho_g = \|q - q_{goal}\| \quad (44)$$

The corresponding attractive force is then given by the negative gradient of the attractive potential:

$$F_{att} = -\nabla U_{att} = -\xi(q_{goal} - q) \quad (45)$$

This force tends to zero as the robot comes closer to the goal. It can be noticed that the attractive force is not bounded as the robot moves away from the goal. If the goal is very far from the robot, this may reduce a very large attractive force. Therefore, the attractive field can be redefined as [17] for m=2:

$$U_{att} = \begin{cases} \frac{1}{2}\xi {\rho_g}^2 & \rho_g \le d \\ d\xi\rho_g - \frac{1}{2}\xi d^2 & \rho_g > d \end{cases} \quad (46)$$

The corresponding attractive force in this case is:

$$F_{att} = -\nabla U_{att} = \begin{cases} -\xi(q - q_{goal}) & \rho_g \le d \\ \frac{-d\xi(q-q_{goal})}{\rho_g} & \rho_g > d \end{cases} \quad (47)$$

However, the repulsive field is introduced as [26]:

$$U_{rep} = \begin{cases} \frac{1}{2}\eta(\frac{1}{\rho_{ob}} - \frac{1}{\rho_0})^2 & \rho_{ob} \le \rho_0 \\ 0 & \rho_{ob} > \rho_0 \end{cases} \quad (48)$$

where $\boldsymbol{\rho}_0$ is the influence distance of the obstacle, $\eta$ is a positive scaling parameter and $\boldsymbol{\rho}_{ob}$ is the distance between the robot and the obstacle which is given as:

$$\rho_{ob} = \|q - q_{obstacle}\| \quad (49)$$

where $\boldsymbol{q}_{obstacle}$ indicates the point on the obstacle, where the distance between this point and the robot is minimal between the obstacle and the robot. The repulsive force is expressed as follows:

$$F_{rep} = -\nabla U_{rep} = \begin{cases} -\eta(\frac{1}{\rho_{ob}} - \frac{1}{\rho_0})\frac{1}{{\rho_{ob}}^2} \times \frac{\partial \rho_{ob}}{\partial q} & \rho_{ob} \le \rho_0 \\ 0 & \rho_{ob} > \rho_0 \end{cases} \quad (50)$$

There are several problems when implementing this function. Trapping situations due to local minima and oscillating in the presence of obstacles are drawbacks of using this repulsive function. Another problem is that the goal can be unreachable when the obstacles are so close to it. In this case, when the robot comes close to the goal it approaches also the obstacles and it will stop without reaching the goal. To avoid such problems, a new repulsive potential function, proposed in [26], is used instead.

The new repulsive functions are introduced by considering the relative distance between the robot and the goal in the calculation of the repulsive field. This approach was inspired by the fact that if the repulsive potentials converge to zero as the robot approaches the goal, then the goal will be at the global minimum. The proposed repulsive function is presented as follows:

$$U_{rep} = \begin{cases} \frac{1}{2}\eta(\frac{1}{\rho_{ob}} - \frac{1}{\rho_0})^2 {\rho_g}^n & \rho_{ob} \le \rho_0 \\ 0 & \rho_{ob} > \rho_0 \end{cases} \quad (51)$$

The last term is added to ensure that the total potential arrives at its global minimum if and only if $q = q_{goal}$ . The resulted force can be written as:

$$F_{rep} = \begin{cases} -\eta\left(\frac{1}{\rho_{ob}} - \frac{1}{\rho_0}\right)\frac{{\rho_g}^n}{{\rho_{ob}}^2} \times \frac{\partial \rho_{ob}}{\partial q} + \frac{n}{2}\eta(\frac{1}{\rho_{ob}} - \frac{1}{\rho_0})^2 {\rho_g}^{n-1} \times \frac{\partial \rho_g}{\partial q} & \rho_{ob} \le \rho_0 \\ 0 & \rho_{ob} > \rho_0 \end{cases} \quad (52)$$

The attractive and repulsive functions used in this work are defined based on the new potential function proposed in [26].

Path planning problem here can be solved as an optimization problem by finding the global minimum in the total field $U_{total}$ starting from initial configuration $\boldsymbol{q}_v(t_0)$. The solution of this problem can be given by the gradient descent. Thus, the desired velocity of the virtual center can be considered to be proportional to the negative gradient of $U_{total}$ as:

$$v_{vc} = -K\nabla U_{total}(q_{vc}) \quad (53)$$

where $K$ is a positive constant and $\nabla\boldsymbol{U}_{total}(\boldsymbol{q}_{vc})$ is derived from equation (52).

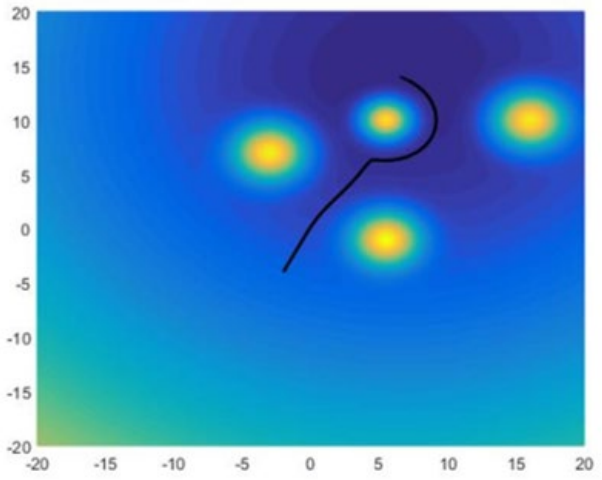
In this work, $_{ob}$ is the distance between the virtual center and the obstacle and $\boldsymbol{\rho}_0$ is chosen to be the minimum acceptable distance between the virtual center and the obstacle. To ensure that none of the robots of the formation collide with the obstacles, $\rho_0$ is determined as:

$$\rho_0 = R + r_{obs} \quad (54)$$

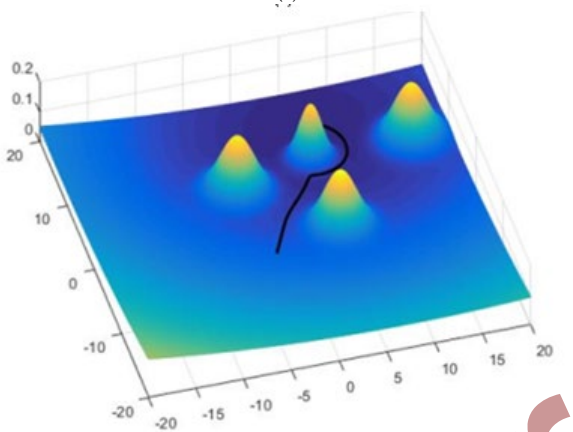
where $r_{obs}$ is the obstacle radius and $R$ is the radius of the formation which indicates the longest distance between the robots to the virtual center, which can be calculated using the following equation:

$$R = max\left\{\sqrt{{x_i}^2 + {y_i}^2}\right\}_{i=1:N} \quad (55)$$

Figure 3 illustrates an example of the potential surface of an obstacle environment. There are four obstacles distributed randomly. To illustrate the effect shows the path generated using the mentioned method.



(a)



(b)

**Figure 3)** *(a) Projection of the obstacles' potential surface, (b) Potential function of 4 obstacle.*

The second part could use the information from other robots to maintain the formation shape which can be guaranteed by using a new cost function as follows:

$$J_{total}{}^{i} = J_t{}^{i} + J_f{}^{i} \tag{56}$$

The dynamic equation of the system represented in a state space form (34) will be considered in the design of the controllers. Since the system is nonlinear, to apply the traditional and Laguerre MPC first we will need to use the linearized system (38) around the reference trajectory $\boldsymbol{x}(t)$. The discrete time presentation of the linearized system can be expressed as follows:

$$x_m(k+1) = A_m x_m(k) + B_m u(k) \tag{57}$$

where the state space matrices in discrete time are defined as:

$$\begin{cases} A_m(k) = I_{n\times n} + T_s A(t) \\ B_m(k) = T_s B(t) \end{cases} \tag{58}$$

where $n$ is the number of state variables, $\boldsymbol{I}_{n\times n}$ is the identity matrix of size $n$ and $T_s$ is the sampling interval.

## TRADITIONAL MPC

The design of discrete time predictive control with constraints is introduced here based on statespace model. The state space form of a general linearized MIMO system in discrete time can be described as follows :

$$\begin{cases} x_m(k+1) = A_m x_m(k) + B_m u(k) + B_d w(k) \\ y(k) = C_m x_m(k) \end{cases} \tag{59}$$

where $u$ is the manipulated variable or input variable; $y$ is the process output; and $x_m$ is the state variable vector with assumed dimension $n$. Taking a difference operation on both sides of (59)

$$x_m(k+1) - x_m(k) = A_m\big(x_m(k) - x_m(k-1)\big) + B_m\big(u(k) - u(k-1)\big) + B_d\big(w(k) - w(k-1)\big) \tag{60}$$

By defining the incremental of a variable $\boldsymbol{v}$(k) by:

$$\Delta v(k) = v(k) - v(k-1) \tag{61}$$

Equation (60) can be rewritten as:

$$\begin{cases} \Delta x_m(k+1) = A_m \Delta x_m(k) + B_m \Delta u(k) + B_d \epsilon(k) \\ \Delta y(k+1) = C_m A_m \Delta x_m(k) + C_m B_m \Delta u(k) + C_m B_d \epsilon(k) \end{cases} \tag{62}$$

where $\Delta \boldsymbol{u}(k)$ is the input to the incremental state space model. To connect $\Delta \boldsymbol{x}(k)$ to the output $\boldsymbol{y}(k)$ a new state vector will be considered:

$$x(k) = [\Delta x_m(k)^T \quad y(k)^T]^T \tag{63}$$

Therefore, the augmented system can be represented as:

$$\begin{bmatrix} \Delta x_m(k+1) \\ y(k+1) \end{bmatrix} = \begin{bmatrix} A_m & 0_m^T \\ C_m A_m & I_{q\times q} \end{bmatrix} \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} + \begin{bmatrix} B_m \\ C_m B_m \end{bmatrix} \Delta u(k) + \begin{bmatrix} B_d \\ C_m B_d \end{bmatrix} \epsilon(k)$$

$$y(k) = [0_m \quad I_{q\times q}] \begin{bmatrix} \Delta x_m(k) \\ y(k) \end{bmatrix} \tag{64}$$

Where q and m are the number of outputs and inputs, respectively. For simplicity, equations (64) can be rewritten as follows:

$$x(k+1) = Ax(k) + B\Delta u(k) + B_\epsilon \epsilon(k) \tag{65}$$

$$y(k) = Cx(k)$$

Considering the following vectors:

$$\Delta U = [\Delta u(k_i)^T \quad \Delta u(k_i+1)^T \quad \dots \quad \Delta u(k_i+N_c-1)^T]^T$$

$$Y = \big[y(k_i+1|k_i)^T \quad y(k_i+2|k_i)^T \quad \dots \quad y\big(k_i+N_p|k_i\big)^T\big]^T \tag{66}$$

According the future predictive states, control increment and output can be derived based on the state space model (**A**, **B**, **C**). Hence, after certain calculations the vector $\boldsymbol{Y}$ is obtained as:

$$Y = Fx(k_i) + \Phi \Delta U \tag{67}$$

Where

$$F = \begin{bmatrix} CA \\ CA^2 \\ CA^3 \\ \vdots \\ CA^{N_p} \end{bmatrix}$$

$$\Phi = \begin{bmatrix} CB & 0 & 0 & \cdots & 0 \\ CAB & CB & 0 & \cdots & 0 \\ CA^2B & CAB & CB & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ CA^{N_p-1}B & CA^{N_p-2}B & CA^{N_p-3}B & \cdots & CA^{N_p-N_c}B \end{bmatrix} \quad (68)$$

Where Nc is the control horizon standing for the number of parameters used to calculate the future control trajectory. Np is the predictive horizon which indicates the optimization window. The main objective of using the model predictive control is to make the predictive output close enough or approximately equal to the set point signal which can be time varying in the meaning of a reference trajectory. To do so, the problem is translated to find the optimal solution for the control vector $\boldsymbol{\Delta U}$ such that the error between the predictive output and its desired trajectory converge toward zero. A cost function to be minimized is chosen to achieve the objectives of the control law.

However, the problem here is a combination of trajectory tracking and formation keeping, so it is more convenient to take that into consideration. To this end, the cost function proposed in (56) is obtained to achieve both trajectory tracking and formation maintaining objective. The proposed cost function contains two separate parts as:

$$J_{total}{}^{i} = J_t{}^{i} + J_f{}^{i} \quad (69)$$

Subject to inequality constraints:

$$M\Delta U \leq Y \quad (70)$$

Where $J_t{}^{i}$ stands for the part of the control signal that leads the ith robot to follow its desired trajectory, while $J_f{}^{i}$ is the other part of the control signal that guarantees formation keeping. Matrix **M** is a matrix reflecting the constraints. The trajectory tracking cost function for the ith robot is defined as:

$$J_t{}^{i} = {E_t{}^{i}}^T Q E_t{}^{i} + \Delta U^{iT} R \Delta U^{i} \quad (71)$$

Where $\boldsymbol{E_t{}^{i}}$ is the tracking error for the ith robot, **Q** is a positive-definite weighting matrix, **R** is a positive semi-definite weighting matrix and $\boldsymbol{\Delta U^{i}}$ is the control increment for robot i. Both matrices are usually diagonal with positive elements. In addition, the formation keeping cost function is descried as:

$$J_f{}^{i} = \sum_{\substack{j=1 \\ j\neq i}}^{n} {E_{f,j}{}^{i}}^T P E_{f,j}{}^{i} \quad (72)$$

Where $\boldsymbol{E}\ f,ji$ is the formation error between the ith and jth robots and **P** is a positive-definite diagonal weighting matrix. The total cost function is:

$$J = \Delta U^{Ti}\left(\Phi^{iT} Q \Phi^{i} + R + (n-1)\Phi^{iT} P \Phi^{i}\right)\Delta U - 2\left({R_s}^{i} - F^{i}x^{i}(k_m)\right)^{T} Q\Phi^{i}\Delta U^{i} + \left({R_s}^{i} - F^{i}x^{i}(k_m)\right)^{T} Q\left({R_s}^{i} - F^{i}x^{i}(k_m)\right) + \sum_{\substack{j=1 \\ j\neq i}}^{n}\left(-2{E_{f,j}^{i}}^{T} P\Phi^{i}\Delta U^{i} + {E_{f,j}^{i}}^{T} P E_{f,j}^{i}\right) \quad (73)$$

Finding the optimal control signal $\Delta\boldsymbol{U}$ that minimizes the cost function (70) subject to the constraints (69) is a Quadratic programming (QP) problem. Hence, the cost function to minimize is written as follows:

$$J = \frac{1}{2}\Delta U^{T} H \Delta U + f^{T}\Delta U \quad (74)$$

$$S.t \quad M\Delta U \leq \gamma$$

**Constraints and model predictive control**

The constraints acting on a process can originate from amplitude limits in the control signal, slew rate limits of the actuator and limits on the output signals, and can be described, respectively, by:

$$\begin{aligned} \Delta U^{min} &\leq \Delta U(k) \leq \Delta U^{max} \\ u^{min} &\leq u(k) - u(k-1) \leq u^{max} \\ y^{min} &\leq y(k) \leq y^{max} \end{aligned} \quad (75)$$

Since the increment of the control is the only variable that can be manipulated, the output constraints must be expressed as a function of the control parameter, $\Delta\boldsymbol{U}$, using the prediction equation (5-15). For an *m* input and *n* output system with a predictive horizon of N, these constraints can be expressed as

$$\begin{aligned} 1\Delta U^{min} &\leq \Delta U(k) \leq 1\Delta U^{max} \\ 1u^{min} &\leq M_u \Delta U(k) - 1u(k-1) \leq 1u^{max} \\ 1y^{min} &\leq \Phi\Delta U(k) + Fx(k_i) \leq 1y^{max} \end{aligned} \quad (76)$$

Where **1** is an $(N \times n) \times m$ matrix formed by $N(m \times m)$ identity matrices and $M_u, M_{du}, M_y$ are block matrices. The constraints in equation (76) need to be written in the form below:

$$M\Delta U \leq \gamma \quad (77)$$

With:

$$M = \begin{bmatrix} M_{du} \\ -M_{du} \\ M_u \\ -M_u \\ M_y \\ -M_y \end{bmatrix}, \quad \gamma = \begin{bmatrix} 1\Delta U^{max} \\ -1\Delta U^{min} \\ 1u^{max} - 1u(k-1) \\ -1u^{min} + 1u(k-1) \\ 1y^{max} - Fx(k_i) \\ -1y^{min} + Fx(k_i) \end{bmatrix} \quad (78)$$

Where $U^{max}, U^{min}$ are the upper and lower limits of the control variable and $y^{max}, y^{min}$ are the upper and lower bounds of the output variable.

All of the terms must be specified for the whole prediction horizon. Usually, block diagonal matrices are the most suitable. There should be a block of correct dimension for each time step in the prediction horizon. For this purpose, Kronecker product of matrices can be quite advantageous.

$$\begin{aligned} M_{du} &= I_{N_c \times N_c} \otimes I_{m \times m} \\ M_u &= I_{N_c \times N_c} \otimes I_{m \times m} \\ M_y &= I_{N_c \times N_c} \otimes \Phi \end{aligned} \quad (79)$$

**Stability of NMPC**

It is important to investigate stability and performance of NMPC. Stability of the NMPC can be achieved by adding stabilizing terminal constraints in addition to Lyapunov function terminal cost. It is showed in the literature that viability of the state constraint set can be replaced by viability of the endpoint constraint so that will guarantee feasibility of the NMPC optimal control problem along the closed loop trajectories. By doing this, the finite horizon NMPC will be estimated about the infinite horizon performance of NMPC closed loop system.

An effective strategy to construct stabilizing terminal constraints set is to include the desired trajectory in the optimization constraints. The

basic idea of the equilibrium terminal constraints is as follows: because the closed loop system is needed to converge to the desired reference, this requirement is added as a constraint set to the optimal control problem. And since "convergence" is not easy to systemize for the finite horizon predictions it sounds reasonable to constrain the predictions to end exactly at the desired equilibrium. To this end, we consider the following assumptions:

a) The point $\boldsymbol{x}_* \in \mathbb{X}$ is an equilibrium for an admissible control value $\boldsymbol{u}_*$ i.e. there exist a control value $\boldsymbol{u}_* \in \mathbb{U}(\boldsymbol{x}_*)$ with $f(\boldsymbol{x}_*, \boldsymbol{u}_*) = \boldsymbol{x}_*$.

b) The running cost $J : \mathrm{X} \times U \rightarrow \mathbb{R}_0^+$ satisfies $(\boldsymbol{x}_*, \boldsymbol{u}_*) = \boldsymbol{x}_*$ for $\boldsymbol{u}_*$ from a.

These equilibrium endpoint constraints help to ensure stability in an intuitive and easy way for implementation. However, it has disadvantage of that the considered system must be controllable to $\boldsymbol{x}_*$ in finite time in order to ensure that the feasible sets indeed contain a neighborhood of the desired time varying trajectory. Therefore, it is not practical for the systems which are stable but uncontrollable. To overcome this drawback, we use a terminal cost function.

The optimal control problem (OCPN,e) is chosen as

$$minimize \quad J_N(\boldsymbol{x}_0, \boldsymbol{u}(.)) := \sum_{k=0}^{N-1} \ell(\boldsymbol{x}_u(k, \boldsymbol{x}_0), \boldsymbol{u}(k)) + F(\boldsymbol{x}_u(N, \boldsymbol{x}_0))$$

With respect to $\boldsymbol{u}(.) \in \mathbb{U}_{\mathbb{X}_0}^N(\boldsymbol{x}_0) \quad with \quad \boldsymbol{x}_* \in \mathbb{X}_0$

$$subject\ to \quad \boldsymbol{x}(0, \boldsymbol{x}_0) = \boldsymbol{x}_0, \quad \boldsymbol{x}_u(k+1, \boldsymbol{x}_0) = \boldsymbol{f}(\boldsymbol{x}_u(k, \boldsymbol{x}_0), \boldsymbol{u}(k))$$

Next, the properties of terminal constraint set $\mathbb{X}_0 \subseteq \mathbb{X}$ and the terminal cost $F$: $\mathbb{X}_0 \rightarrow \mathbb{R}^+_0$ will be specified. To do so, we will consider the following assumption:

Assumption 1: For the closed endpoint constraint set $\mathbb{X}_0 \subseteq \mathbb{X}$ defining $\mathbb{U}_{\mathbb{X}_0}^N(\boldsymbol{x}_0)$ and the terminal cost $F$: $\mathbb{X}_0 \rightarrow \mathbb{R}^+_0$ we assume:

a) $\mathbb{X}_0$ is viable, i.e., for each $\boldsymbol{x} \in \mathbb{X}_0$ there exists an admissible control value $\boldsymbol{u}_x \in \mathbb{U}(x)$ such that

$(\boldsymbol{x}, \boldsymbol{u}_x) \in \mathbb{X}_0$

Holds.

a) The terminal cost: $\mathbb{X}_0 \rightarrow \mathbb{R}^+_0$ is such that for each $\boldsymbol{x} \in \mathbb{X}_0$ there exists an admissible control value $\boldsymbol{u}_x \in \mathbb{U}(x)$ for which following relation hold.

$$F(f(\boldsymbol{x}, \boldsymbol{u}_x)) + \ell(\boldsymbol{x}, \boldsymbol{u}_x) \leq F(x)$$

This assumption indicates that $F$ is a local control Lyapunov function of the control system. The approach of adding $F$ is often referred to as quasi-infinite horizon NMPC. The reason for this denomination is that if the terminal cost $F$ is an approximation of the infinite horizon optimal value function $V_\infty$, then the finite horizon dynamic programming principle is an approximation of the infinite horizon dynamic programming principle can be interpreted as an approximation to the infinite horizon problem ($OCP_\infty^n$).

## SIMULATION

### Simulation parameters

The system consists of three nonholonomic differential drive wheeled mobile robots. The three robots are identically the same and each has the parameters shown in table 1.

**TABLE 1**
**Single robot properties**

| Mass of the robot, m | 4.5kg |
|---|---|
| Moment of inertia, I | 0.0572 kg.m$^2$ |
| Distance between the vehicle longitudinal axis and each wheel, b | 0.125m |
| Radius of each driven wheel, r | 0.05m |
| Distance between center of mass and the center point of common wheel axis, d | 0.11m |

The formation parameters $x_i$, $y_i$ are given in table 2:

**TABLE 2**
**Formation parameters**

| | $X_i$ | $Y_i$ |
|---|---|---|
| Robot 1 | 0.104 | 0.136 |
| Robot 2 | -0.712 | -0.656 |
| Robot 3 | -0.88 | 0.88 |

The reference trajectory of the virtual center is generated using the proposed APF functions. The initial position and orientation of the virtual center is: $x_{vc}(t0) = [-0.2 \quad -1.5 \quad 0]^T$. Suppose that the initial velocities of the robots are zeros. The initial position and orientation of each robot is:

$$x_1(t0) = [-0.75 \quad -1.3975 \quad 0.327]^T,$$
$$x_2(t0) = [-0.398 \quad -2.702 \quad 0.799]^T,$$
$$x_3(t0) = [-1.485 \quad -1.397 \quad 0.718]^T$$

The goal point for the virtual center to reach is: $x_{vc}(t0) = [2.2 \quad 4.8]^T$. Moreover, the group of the mobile robots forms a circle with a radius $R = 2.12$ where the virtual center is the center of the circle. The robots move in an environment, where there are three static obstacles. The position and radius for each obstacle can be found in table 3.

**TABLE 3**
**Obstacles specifications**

| Obstacle | Position | Radious |
|---|---|---|
| **Obs1** | [-1.25;1.3] | 1.5 |
| **Obs2** | [0.5;3.1] | 1 |
| **Obs3** | [1.6;1] | 1.5 |

Constrains was designed for both input and state variables. In fact, we constrain the wheels' torques and the wheels' velocities, see table (4).

**TABLE 4**
**Constraints on inputs and states**

| | limitations |
|---|---|
| Wheel torque (N.m) | [-0.25,0.25] |
| Wheel angular velocity (rad/sec) | [-12,12] |

To demonstrate the effectiveness of the proposed control law, a disturbance is applied to one of the agents in the system. In fact, actuators don not work perfectly all the time. To illustrate the effect of this phenomenon, a disturbance is exerted only on the first robot during the time interval of seconds.

The same parameters are used also for the other MPC strategies which they will be discussed later in this chapter. For each controller two cases will be discussed to visualize the effectiveness and importance of the control law. First, formation control part of the control law will be deleted. The second case will contain the formation control.

**Simulation results**

The software used in this simulation is MATLAB and the simulations are done in the simulink environment. The available computer system used for the simulation has Windows 8.1 specifications with Intel 7-core CPU with 2.8 Hz and 8 GB of RAM. The open loop control scheme of the simulation is illustrated in figure 4. The input to the approximated model is the difference between feedforward and original inputs. The error is combined with the reference trajectory to get the real trajectory using the linearized model. However, the input to the nonlinear model is the feedforward control and the output is the real trajectory.

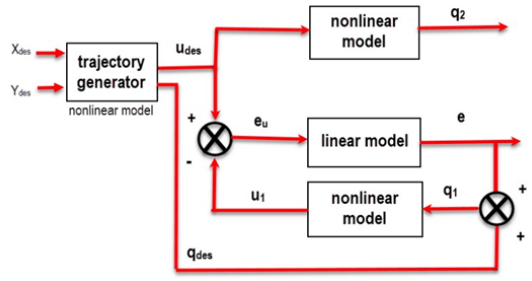



**Figure 4)** *open loop control for system verification.*

Control parameters are chosen to minimize the error as follows:

$$Q = \begin{bmatrix} 110 & 0 & 0 & 0 & 0 \\ 0 & 140 & 0 & 0 & 0 \\ 0 & 0 & 50 & 0 & 0 \\ 0 & 0 & 0 & 0.05 & 0 \\ 0 & 0 & 0 & 0 & 0.05 \end{bmatrix}$$

$$L = \begin{bmatrix} 20 & 0 & 0 & 0 & 0 \\ 0 & 30 & 0 & 0 & 0 \\ 0 & 0 & 8 & 0 & 0 \\ 0 & 0 & 0 & 0.012 & 0 \\ 0 & 0 & 0 & 0 & 0.012 \end{bmatrix}$$

$$R = \begin{bmatrix} 2000 & 0 \\ 0 & 2000 \end{bmatrix}, \quad N_c = 10$$

a) Without formation control: Figure (4) shows the desired trajectory of the virtual center, the real paths of the robots and the obstacles (Figure 5).

Figures (9-a), (9-b) and (9-c) illustrate the errors of the three robots.

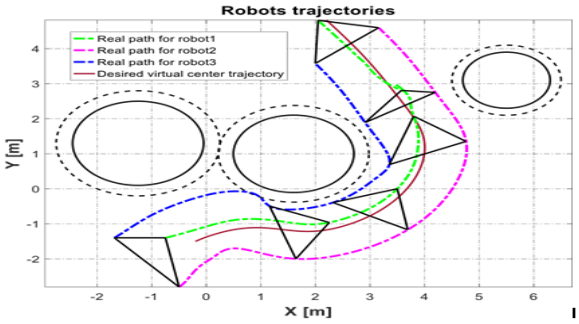



**Figure 5)** *Virtual center and robot's trajectories and obstacles using MPC with the presence of disturbance on the first robot without formation control.*

Figure (6 and 7) indicates the formation error between the three robots, which is relatively large in this case.

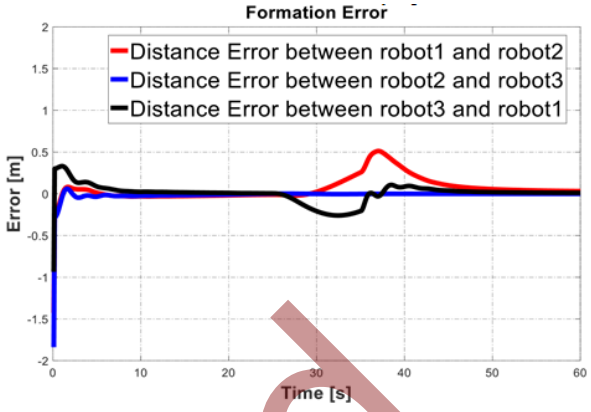



**Figure 6)** *Formation error between the robots.*

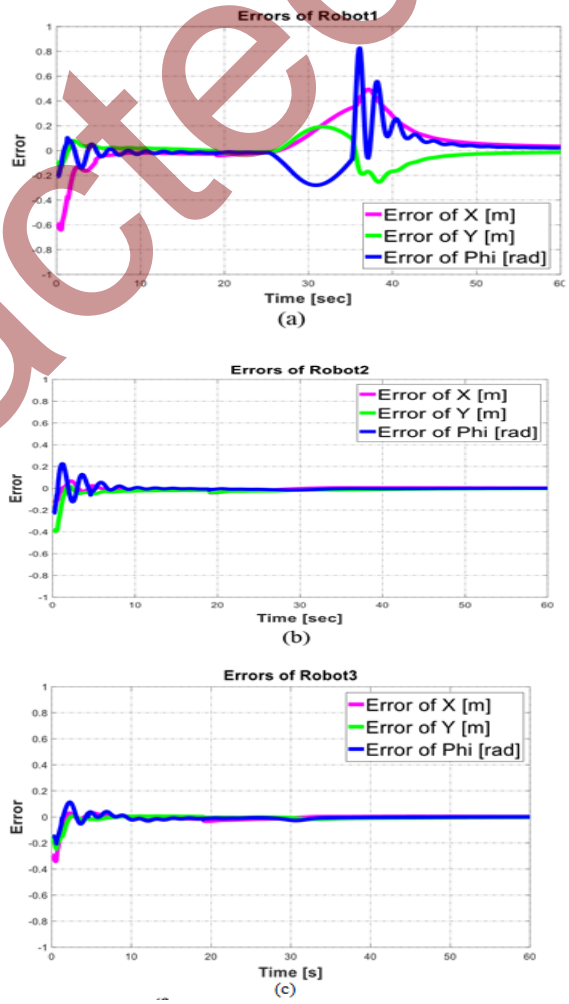



**Figure 7)** *Error signals of X, Y and $\varphi$. (a) errors of robot 1. (b) errors of robot 2. (c) errors of robot 3.*

It is obvious that the robots cannot maintain the desired formation when the formation cost function is ignored. It can be seen that when the first robot has disturbance, the other two robot continue their paths without reacting to the occurred disturbance leading to a large formation error.

The required wheels torque for robot 1 is demonstrated in figure 8, which are bounded so that the constraints are satisfied.

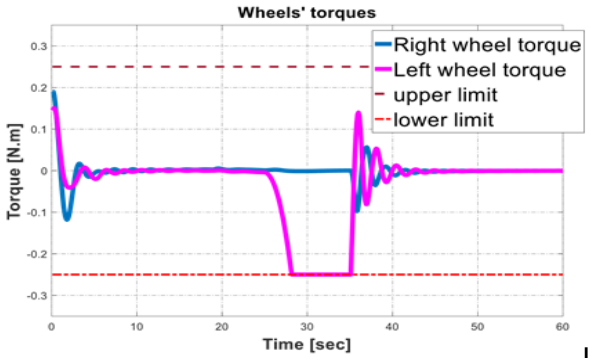



**Figure 8)** *First robot wheels' torque.*

b) With formation control: Figures 9, 10 and 11 illustrate results of implementing MPC with formation control. The trajectory of the virtual structure and the robots' real paths are shown in figure 9.

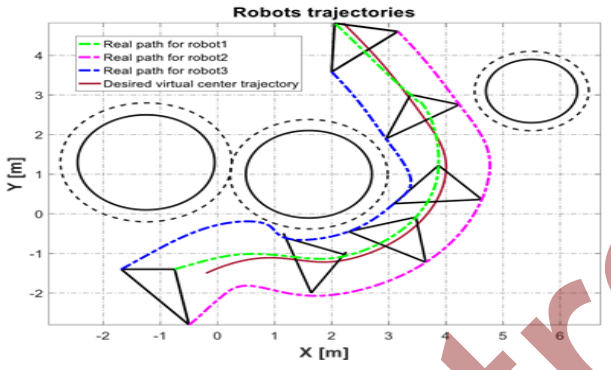
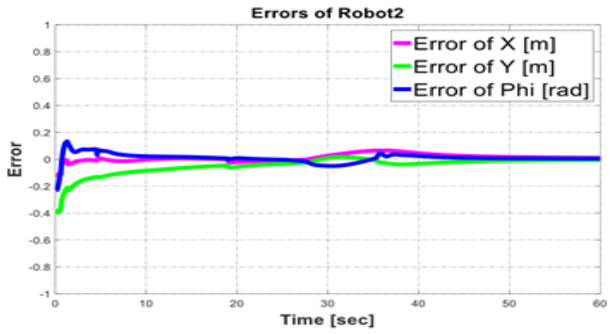



**Figure 9)** *Virtual center and robot's trajectories and obstacles using MPC with the presence of disturbance on the first robot with formation control.*

It is obvious that the robots can follow their trajectories while maintaining the desired formation when the formation cost function is considered. Figures 10-a, 10-b and 10-c illustrate the errors of the three robots. It can be seen that when the first robot has disturbance, the other two robot try to compensate the formation error leading to a tracking error, i.e. the robots change their paths related to the occurred disturbance.

Figure 11 indicate the formation error between the three robots, which is reduced in comparison with case (a).

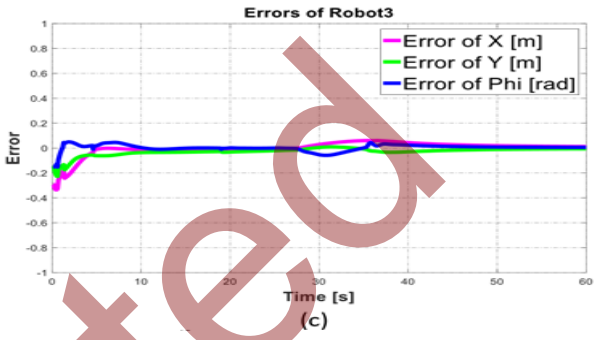




**Figure 10)** *Error signals of X, Y and φ. (a) errors of robot 1. (b) errors of robot 2. (c) errors of robot 3.*

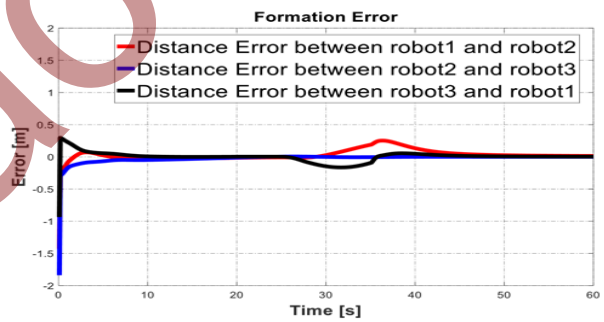



**Figure 11)** *Distance error between the robots.*

The required wheels torque for robot 1 is demonstrated in figure 12.

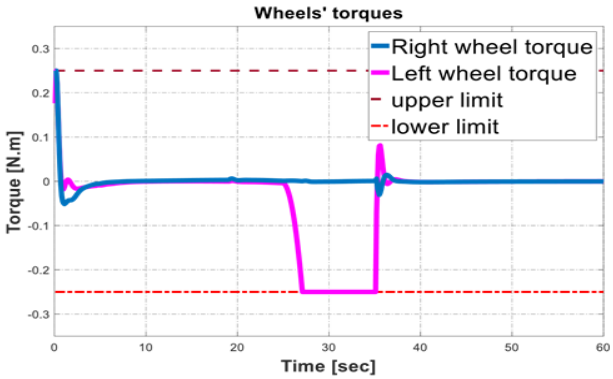
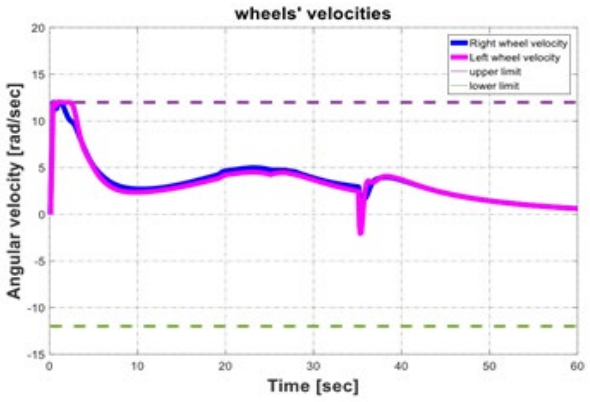



**Figure 12)** *Wheels' torques for robot 1.*

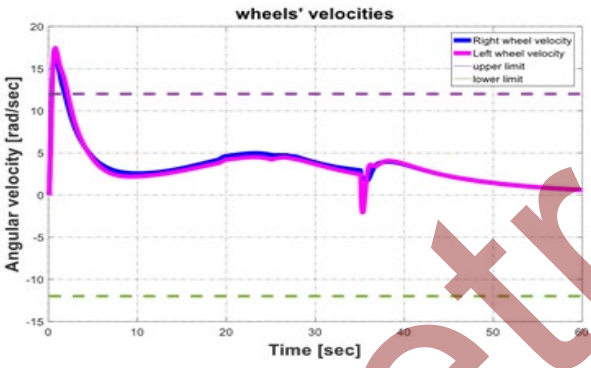
In the two cases discussed above the predictive horizon is $N_p$=10. Comparing results from case (a) and (b), we can conclude that the control law is effective and it drives the formation error as near as possible to zero while keeping the robots on their own trajectories. In

the case of formation control absence, the robots don not interact with each other, so that when one of them is suffering from disturbance or uncertainties the formation shape will be changed.

Since in practice the velocity of the motor is limited, two cases are studied, control with constraints of velocity and control without such constraints. Figure 13 indicates the angular velocities of the right and left wheels in both situations. It is seen that the constraints are satisfies and the velocities cannot exceed the limits defined as desirable.




(a)




(b)

**Figure 13)** *Angular velocities of right and left wheels. (**a**) Control with constrains. (**b**) Control without constraints.*

Figures 14 illustrate the actual and the desired angular velocities for each wheel using nonlinear predictive control without and with constraints, respectively. When comparing each pair of figures, we can conclude that here is an improvement in the performance where the over-shoot in the output becomes slightly small.

When Compare simulation result it can be easily realized that nonlinear MPC results in better performance than the other two controllers. Using NMPC, the error signals of each robot are more stable and converge to zero faster than the other two methods.

The control effort and the error resulted in the three control approaches are illustrated in table 5.

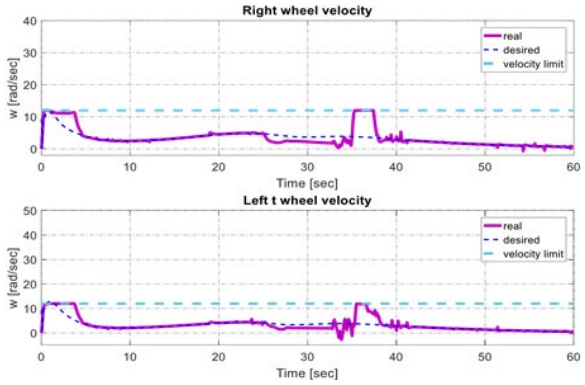



**Figure 14)** *Actual and desired velocities in the case of control without constraints.*

**TABLE 5**
**CE and TE for both controllers**

| | Robot 1 | Robot 3 | Robot 3 |
|---|---|---|---|
| | MPC | MPC | MPC |
| U1 | 0.2353 | 0.2264 | 0.2951 |
| U2 | 2.4262 | 0.2058 | 0.2178 |
| E1 ($e_x$) | 6.7297 | 1.7169 | 3.2488 |
| E2 ($e_y$) | 3.4615 | 3.8744 | 2.8263 |
| E3 ($e_{theta}$) | 3.6174 | 1.7414 | 1.4548 |

Given the above, the nonlinear model predictive control gives better results and the performance obtained with this method is more stable. This is reasonable since the system to be controlled is nonlinear and has constraints on the control signal and MPC takes into account both current values and future desired values.

## CONCLUSION

This paper addresses the design and implementation of formation control of a multi-robot system in an environment with obstacles. The problem of formation control of a multi-agent system can be defined as the problem of trajectory tracking while maintaining a desired geometric formation to achieve specific tasks. To this end, the objectives of our controllers will be divided into two parts, trajectory tracking and formation keeping. The nonlinear dynamics of the mobile robot is used to design different controllers. Artificial potential field is used to generate the desired trajectory for the formation then each robot trajectory is determined from the virtual structure approach.

Three other controllers are obtained using three different methods of model predictive control, traditional MPC, Laguerre based MPC and nonlinear MPC. The idea behind the formation control using MPC strategies is to find the optimal solution that achieve control objective of trajectory tracking and formation maintaining. To do so, the cost function to be minimize includes a term representing the trajectory tracking error and another term related to the formation error. From the simulation results, applying these three methods on a disturbance system emphasizes the effectiveness of the control law. These results also reveal that the nonlinear MPC results in better performance than the other two controllers. One of the advantages of using MPC approaches over feedback linearization is finding the optimal solution of the problem with respect to constraints on input, output and

control variables. In fact, from the path and error of robot 1 to 3, according to table 5, it is possible to see the values of the control inputs as well as the planar and angular control fields, which show that these values show better control stability compared to the first category of robots, so that the robot Compared to the first robot, the third one has almost 50% less error than the first robot and also spends less optimal energy than avoiding obstacles for routing.

To conclude, this thesis concerned with the problem of formation control using virtual center approach and several controllers. The simulation results showed the effectiveness of the proposed methods and their ability in maintaining the desired formation. Although the thesis mainly focused on the simulation analysis, implementation on a real platform was accomplish to an extent. However, exterminations are open for future research.

The future researches also can be extended to further objectives such as formation control in a dynamic obstacle environment. Additional suggestions for future work is to implement to real platform.

## ACKNOWLEDGMENT

The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University, Abha, Saudi Arabia for funding this work through Large Groups Project under grant number RGP.2/347/44.

## REFERENCES

1. JiangC,ChenZ,GuoY.Multirobotformationcontrol:acomparisonbetweenmodelbasedandlearningbasedmethods.JControlDecis.2020;7(1):90-108.
2. NascimentoTP,MoreiraAP,ConceiçãoAGS.Multi-robotnonlinearmodelpredictiveformationcontrol:Movingtargetandtargetabsence.RobotAutonSyst.2013;61(12):1502-1515.
3. Alonso-MoraJ.Distributedmulti robotformationcontrolindynamicenvironments.AutonRobots.2019;43:1079-1100.
4. QinH.Reviewofautonomouspathplanningalgorithmsformobilerobots.Drones.2023;7(3):211.
5. ParkerLE,RusD,SukhatmeGS.Multiplemobilerobotsystems.SpringerHandbookofRobotics;2016.1335-1384.
6. MoghaddamBM,ChhabraR.Ontheguidance,navigationandcontrolofinorbitspaceroboticmissions:Asurveyandprospectivevision.ActaAstronaut.2021;184:70-100.
7. SampedroC.Afullyautonomousaerialrobotforsearchandrescueapplicationsinindoorenvironmentsusinglearningbasedtechniques.JIntellRobotSyst.2019;95:601-627.
8. MackayJ,MunozA,PepperM.Conceptualisingredundancyandflexibilitytowardssupplychainrobustnessandresilience.JRiskRes.2020;23(12):1541-1561.
9. DorriA,KanhereSS,JurdakR.Multiagentsystems:Asurvey.IEEEAccess.2018;6:28573-28593.[
10. KorayemMH,AdrianiHR,LademakhiNY.Intelligenttime-delayreductionofnonlinearmodelpredictivecontrol(NMPC)forwheeledmobilerobotsinthepresenceofobstacles.ISATrans.2023.
11. AframA,Janabi-SharifiF.TheoryandapplicationsofHVACcontrolsystems–Areviewofmodelpredictivecontrol(MPC).BuildEnviron.2014;72:343-355.
12. MillerM.Repetitivebehaviorwithobjectsininfantsdevelopingautismpredictsdiagnosisandlatersocialbehaviorasearlyas9months.JAbnormPsychol.2021;130(6):665.
13. Alonso-MoraJ.Distributedmulti-robotformationcontrolamongobstacles:Ageometricandoptimizationapproachwithconsensus.IEEEinternationalconferenceonroboticsandautomation(ICRA).IEEE;2016.
14. TranVP,GarrattMA,andPetersenIR.Switchingformationstrategywiththedirecteddynamictopologyforcollisionavoidanceofamultirobotsysteminuncertainenvironments.IETControlTheoryAppl.2020;14(18):2948-2959.
15. LeeG,ChwaD.Decentralizedbehavior-basedformationcontrolofmultiplerobotsconsideringobstacleavoidance.IntellServRobot.2018;11:127-138.
16. MachadoT.Attractordynamicsapproachtojointtransportationbyautonomousrobots:theory,implementationandvalidationonthefactoryfloor.AutonRobots.2019;43:589-610.
17. SoorkiMN,TalebiHA,NikraveshSKY.Arobustdynamicleadrfollowerformationcontrolwithactiveobstacleavoidance.IEEEInternationalConferenceonSystems,Man,andCybernetics.IEEE;2011.
18. WangY.Apracticalleader–followertrackingcontrolschemeformultiplenonholonomicmobilerobotsinunknownobstacleenvironments.IEEETransControlSystTechnol.2018;27(4):1685-1693.
19. XiaoH,LiZ,ChenCLP.Formationcontrolofleader–followermobilerobots'systemsusingmodelpredictivecontrolbasedonneuraldynamicoptimization.IEEETransIndElectron.2016;63(9):5752-5762.
20. Drŏğa J.Allyouneedtoknowaboutmodelpredictivecontrolforbuildings.AnnuRevControl.2020;50:190-232.
21. LiH,ShiY.Distributedmodelpredictivecontrolofconstrainednonlinearsystemswithcommunicationdelays.SystControlLett.2013;62(10):819-826.
22. KamelMA,ZhangY.Linearmodelpredictivecontrolviafeedbacklinearizationforformationcontrolofmultiplewheeledmobi

lerobots.IEEEInternationalConferenceonInformationand Automation.IEEE;2015.

23. LiSE.Robustlongitudinalcontrolofmulti-vehiclesystems—AdistributedH-infinitymethod.IEEETransIntellTranspSyst.2017;19(9):2779-2788.
24. WenB,HuangJ.Leader-FollowingFormationTrackingControlofNonholonomicMobileRobotsConsideringCollisionAvoidance:ASystemTransformationApproach.ApplSci.2022;12(24):12579.
25. XuZ,YangSX,GadsdenSA.EnhancedbioinspiredbacksteppingcontrolforamobilerobotwithunscentedKalmanfilter.IEEEAccess.2020;8:125899-125908.
26. GeSS,CuiYJ.Newpotentialfunctionsformobilerobotpathplanning.IEEETransRobotAutom.2000;16(5):615-620.