Quadratic Programming Algorithm of Approximate Nonlinear Model Predictive Control for Tracking Trajectory of Wheeled Mobile Robots

Ramdane HEDJAR, Senior member, IEEE
Department of Computer Engineering, College of Computer and Information Sciences (CCIS), P. O. Box 5117, Riyadh 11543, King Saud University, Saudi Arabia.

Corresponding author: Ramdane Hedjar (e-mail: hedjar@ksu.edu.sa).

This work was supported by the Research Center of the College of Computer and Information Sciences, King Saud University. The author is grateful for this support.

ABSTRACT Nonlinear model predictive control (NMPC) has proved its competency in controlling constrained nonlinear processes. Although NMPC can achieve exemplary tracking performance, the related computation effort as well as guaranteeing tracking convergence are its main drawbacks. Indeed, constrained NMPC is a nonlinear and nonconvex optimization problem where it is difficult to find a feasible solution within a reasonable time. Motivated by these difficulties, this paper proposes a procedure, using Euler approximation, to transform the nonlinear optimization problem of NMPC into a constrained quadratic optimization problem. The proposed tracking controller is applied to the autonomous navigation problem of a wheeled mobile robot (WMR) in a constrained space. Under certain assumptions, we prove the closed-loop system stability and the boundedness of the tracking error. Further, we show the recursive feasibility of the solution. Simulations are performed, first to determine the adequate control parameters, and secondly to show the effectiveness of the proposed algorithm, while its real-time implementation is experimentally verified using a differential drive mobile robot.

Index Terms— Nonlinear predictive control algorithm, state and control constraints, convex optimization, stability, and feasibility analysis, wheeled mobile robot.

I. INTRODUCTION

Over the last few decades, mobile robots have been widely studied due to their inherent applications in different fields. Indeed, mobile robots are being used for a simple exploration of the unknown region for search and rescue military applications and surveillance [1]. Hence, many researchers have focused on the autonomous navigation of mobile robots in an unknown environment with static and dynamic obstacles. Therefore, the desired trajectory is generated online by a motion planner, based on the information acquired from the environment, in order to avoid obstacles and minimize the distance to the target. Control methods utilized to control the under-actuated systems are mainly nonlinear. Therefore, many methods based on nonlinear techniques, have been developed to solve the tracking problem and allow the mobile robot to track the desired trajectory [2,3,4,5]. While the authors in [6,7] have focused on a more complex environment (outdoor) where the wheels are subject to deformation. However, mobile robots have physical limits, and space constraints that affect the achievable closed-performance in the indoor and outdoor environment.

It is a known fact that model predictive control is the best technique used to cope with hard constraints on controls and states. For this reason, nonlinear model predictive control has been widely used in academic research and industry due to its simplicity to include different constraints in the optimization problem [8,9]. To solve the NMPC problem variety of approaches, based on sequential quadratic programming (SQP) algorithms, have been developed. This direct method is based on converting the NMPC problem to an optimal control problem. This nonlinear programming problem (NLP) requires an online solution to a receding horizon optimization problem in order to be implementable in practice. Thus, this is a nonlinearly constrained optimization problem with the following challenges:

1- The processing time of the algorithm.
2- Nonconvexity of the optimization problem.
3- Stability of the closed-loop system and feasibility of the solution.

For this reason, most of the NMPC applications that have been reported in the literature have been applied to systems with slow dynamics like, for instance, chemical processes. For instance, in [10], the authors have designed the NMPC to a continuous stirred tank reactor (CSTR) using a multi-stage control approach. While the authors in [11], presented a linear mismatched model-based offset-free model-predictive control approach for CSTR nonlinear system. In [12], the authors have applied the robust nonlinear predictive control to the CSTR process using the radial basis function network (RBF-ARX) model. In [13], the authors have formulated the nonlinear optimization problem as a quadratic cost function with inequality constraints and applied it to the CSTR plant.

In recent years significant effort has been made to cope with the real-time implementation of the NMPC to be applied to high dynamic systems with a fast-sampling rate. The real-time iteration (RTI) scheme has been proposed in [14], and the authors have combined ideas of the RTI with Multi-Level Iteration (MLI) that ensures high computational efficiency for real-time implementation of the NMPC. The authors in [15] have proposed a homotopy-based nonlinear interior-point method for an efficient real-time implementation of NMPC. In [16], the authors have developed a highly parallelizable Newton-type method to solve efficiently the NMPC. In [17], the authors have presented a collection of embedded optimization algorithms under a modular structure written in C language for better real-time optimization for nonlinear optimization. A parallel optimization toolkit for a real-time implementation of NMPC has been proposed in [18]. Indeed, the authors have used a highly parallelizable method for an efficient real-time implementation of NMPC. The authors in [19] have used the RPROP algorithm based on a faster backpropagation approach for the gradient-descent optimizer. To this end, the authors in [20] have used the high-performance framework ACADO software for a computation-efficient implementation of the NMPC to reluctance synchronous machines. Consequently, to overcome the computational burden of the NMPC, a huge amount of NMPC schemes have been developed during the last years (see the excellent reviews [39]).

Despite previous challenges, several applications of NMPC to a mobile robot can be found in the literature. First applications have focused on tracking the desired trajectory [19,21,22]. In [23], the authors have proposed NMPC with a nonlinear (nonconvex) optimization algorithm to track the centerline of the roadway while avoiding obstacles. However, the iterative method used to determine the optimal solution is a high time-consuming algorithm. The authors in [24] have combined path following controller with obstacle avoidance using nonlinear model predictive control. Even though the authors have provided the stability and feasibility analysis of the algorithm, the optimization problem is nonlinear and nonconvex with a high computational burden. The nonlinear model predictive control has been also used in [25] to steer the WMR to the desired pose. Despite of the rigorous asymptotic stability analysis, the proposed solution needs to solve a nonlinear optimization problem. To avoid the local minimum of the optimization problem, using receding horizon control (RHC), the authors in [26] have proposed a local-minima-free navigation function with the global minimum at the target state. The proposed algorithm solves the navigation problem with control constraints. The authors in [27] have used a genetic algorithm (GA) to solve the NMPC algorithm to avoid the computational difficulties of the nonlinear programming-based MPC. The algorithm has been successfully applied to solve the steering problem of an autonomous vehicle. To reduce the processing time due to the online optimization, the authors in [28] proposed a convex quadratic programming-based model predictive scheme for collision-free collision navigation of autonomous vehicles.

Consequently, the processing time of the NMPC algorithm is one of the most commonly encountered problems when NMPC is used to control a highly dynamic nonlinear system. Indeed, the time required to reach the optimal solution is high with regard to the dynamic of the process. This is due to the nonlinear optimization problem that is solved at each sample time (online) [29]. Further, the optimization problem is, in general, nonconvex and the solution can converge to the local minimum [30, 31]. Hence, the convexity of the optimization problem plays a central role in the practical implementation of the constrained NMPC.

In this paper, we propose a simple algorithm, based on approximated model predictive control, for safe autonomous navigation of the WMR in a constrained space. This algorithm permits the WMR to track accurately the feasible predefined trajectory. To avoid the computational burden of the NMPC, the Euler approximation is utilized to convert the NLP to constrained quadratic programming (QP). Figure 1 describes the constrained QP implementation of the proposed algorithm. The stability analysis of the closed-loop system is investigated using the Lyapunov method and the positive invariance principle in constrained cases. Therefore, the contribution of this work resides in the transformation of the tracking problem, which is a nonconvex optimization problem, to a constrained QP problem. The restricted space and constraints on the control input are written as convex constraints (inequality constraints). Thus, the heavy computation burden of the original optimization problem can be reduced considerably using the fast-available algorithms [32,33].

The proposed algorithm can be used efficiently in applications where the navigation space is limited, like for instance, robot vacuum cleaners and landmine detection robots. Further, undetectable obstacles, like for instance downstairs in an indoor environment, can also be included as space constraints to avoid the robot falling downstairs. Therefore, the proposed control algorithm is simple, practical, and efficient with the feasible solution and asymptotic stability of the equilibrium point of the closed-loop system.

The paper is organized as follows. In section 2, the kinematic model of the unicycle mobile robot is derived with some considered assumptions and properties. Section 3 deals with the trajectory tracking algorithm based on approximated nonlinear model predictive control (ANMPC). The stability issue of the closed-loop system and the feasibility of the solution are discussed in section 4. To see the effectiveness of the proposed
scheme, simulations and experimental tests have been performed. The obtained results are shown in section 5. The paper ends with a conclusion and future work.

II. PRELIMINARIES

This section deals with some preliminaries required for this work.

A. MATHEMATICAL MODELING

In this work we consider a differential drive mobile robot. It contains two independent rear wheels and a front caster wheel. The linear speed $v$ of the WMR is given by the combination of the left speed $v_l$ and right speed $v_r$ of each DC motor: $v = \frac{v_l + v_r}{2}$ and the angular speed is given by: $\omega = \frac{v_l - v_r}{L}$, where $L$ is the wheelbase. The transformation between these variables is given in (Fig.1):

$$
\begin{bmatrix}
v_l \\
v_r
\end{bmatrix} = \begin{bmatrix}
\frac{L}{2} & 0 \\
0 & \frac{L}{2}
\end{bmatrix} \begin{bmatrix}
v \\
\omega
\end{bmatrix} = T \begin{bmatrix}
v \\
\omega
\end{bmatrix}.
$$

The Kinematic model of the mobile robot is given by [1]

$$
\begin{align*}
x_c &= v \cos(\theta) \\
y_c &= v \sin(\theta) \\
\dot{\theta} &= \omega
\end{align*}
$$

(1)

This kinematic model is valid under the nonhOLONomical constraint [1]. To overcome the controllability problem of the kinematic model (1), the coordinates of the WMR are shifted from the mass center $C(x_c, y_c)$ to the point $C(x, y)$ along the sagittal axis by the distance $|d| \neq 0$ (see figure 2). Thus, new coordinates of the WMR are given by

$$
\begin{align*}
x &= x_c + d \cos(\theta) \\
y &= y_c + d \sin(\theta)
\end{align*}
$$

Under the matrix form, the nonlinear kinematic model of WMR is

$$
\dot{z}(t) = f(Z, U, t) = G(\theta) \ u(t)
$$

(2)

where $G(\theta) = \begin{bmatrix} \cos(\theta) & -d \sin(\theta) \\ \sin(\theta) & d \cos(\theta) \end{bmatrix}$.

$Z(t) = [x(t), y(t), \theta(t)]^T$ and $u(t) = [v(t), \omega(t)]^T$. The discrete-time version of the model (2) is

$$
Z(k+1) = Z(k) + T_s \ G(\theta(k)) \ u(k)
$$

where $T_s$ is the sampling time.

The tracking error dynamic can be written as

$$
\dot{e}(t) = f(Z, U, t) - \dot{Z}_{ref} = G(\theta) \ u(t) - \dot{Z}_{ref}
$$

(3)

B. ASSUMPTIONS

In order to design the ANMPC algorithm, some assumptions are required.

1. Both reference position $Z_{ref}(t)$ and its time derivative $\dot{Z}_{ref}(t)$ are bounded. So, $\exists \ Z_{max} > 0, \dot{Z}_{max} > 0$ such that $|Z_{ref}(t)| < Z_{max}$; $|\dot{Z}_{ref}(t)| < \dot{Z}_{max}$. $\forall \ t \in \Re^+$ where $Z_{ref}(t) = [x_{ref}(t), y_{ref}(t), \theta_{ref}(t)]^T \in \Re^3$.

2. The first-order Taylor approximation (Euler method) is used to convert the constrained nonconvex optimization problem to a constrained quadratic optimization problem. It is assumed that the approximation is valid for a small step-time $h$.

3. The state vector $Z(t)$ is available.

C. DEFINITIONS

Here, we define some properties that will be used in the stability and feasibility analysis.

1. The column vectors of $G(\theta)$ are linearly independent. Consequently, $G(\theta) \ u(t) = 0 \Rightarrow u(t) = 0$. Accordingly, for any positive definite (or negative definite) matrix $\Pi$, the matrix $G^T(\theta) \Pi G(\theta)$ is also a definite positive (or negative definite) matrix.

2. The minimal and maximal eigenvalue of a matrix $Q$ is denoted by $\lambda_{min}(Q)$ and $\lambda_{max}(Q)$ respectively.

3. The identity matrix is denoted by $I_n$.

4. The Euclidean norm in $\Re^n$ is written as $|.|$ while for a matrix $P \in \Re^{n \times m}$, $\|P\| = \lambda_{max}(P^T P)^{1/2}$.

D. NOTATION

In addition to the nonholonomic constraint, the position of the WMR $P_{xy}(t) = [x(t), y(t)]^T \in \Re^2$ and the input vector $u(t) = [v(t), \omega(t)]^T \in \Re^2$ are constrained inside a given compact set. For the control signal: $U = [y_{min}, y_{max}] \times [\omega_{min}, \omega_{max}] \subset \Re^2$.

The sequences of the control signals

$$
U(t) = [u(t), u(t+h), \ldots, u(t+(m-1)h)]^T
$$
is feasible, if \( U(t) \in \mathcal{U}_T \), where
\[
\mathcal{U}_T = \{ U(t) \in \mathbb{R}^{2 \times m} / u(t+i+1) \in \mathcal{U} \text{ for } i = 0, \ldots, m-1 \}.
\]
while the restricted cartesian space is defined as compact (nonempty) set by:
\[
\mathcal{Z} = \{ [x_{min} - x_{max}] \times [y_{min} - y_{max}] \times \mathbb{R} \subset \mathbb{R}^3 \},
\]
\[
\mathcal{Z}_T = \{ Z(t) \in \mathbb{R}^{3 \times m} / Z(t+i-h) \in \mathcal{Z}, \text{ for } i = 0, \ldots, m \},
\]
\[
Z(t) = [Z(t+h) Z(t+2h) \ldots Z(t+mh)]^T.
\]
We can also define the constrained compact set related to the predicted tracking error as
\[
\mathcal{E} = \{ [x_{min} - x_{ref}(t), x_{max} - x_{ref}(t)] \times [y_{min} - y_{ref}(t), y_{max} - y_{ref}(t)] \times \mathbb{R} \subset \mathbb{R}^3 \},
\]
\[
\mathcal{E}_T = \{ E(t) \in \mathbb{R}^{3 \times m} / e(t+i) \in \mathcal{E} \text{ for } i = 1, \ldots, m \},
\]
where
\[
E(t) = [e(t+h) e(t+2h) \ldots e(t+mh)]^T.
\]

Then, for an initial feasible state error \( e(t) \), the sequence of the control vector is
\[
U(t) = [u(t) u(t+h) \ldots u(t+(m-1)h)]^T \in \mathcal{U}_T,
\]
an admissible control vector, if the state error trajectories \( E(t) \), generated by the dynamic system (3), satisfies \( E(t) \in \mathcal{E}_T \).

We also define a compact set \( \mathcal{Z}_{eq} = \{ U(t) \in \mathbb{R}^{2 \times m}, Z(t) \in \mathbb{R}^{3 \times m}, U(t) \in \mathcal{U}_T, Z(t) \in \mathcal{Z}_T \} \), which will be converted to a polyhedral set using the Euler approximation.

### III. RECEDING HORIZON CONTROL OF A WMR

The formulation of the NMPC problem for trajectory tracking is to find the control vector \( u(t) \) such that the predicted tracking error \( e(t+\tau) = Z(t+\tau) - Z_{ref}(t+\tau) \) converges to zero along the interval \( \tau \in [t, t+T] \). Note that \( Z_{ref}(t) \) is the safe trajectory generated by the planner (Figure 1) in order to move the WMR to the desired target without colliding with obstacles. The above problem is reformulated as an optimization problem \([8, 9]\):
\[
\text{Min}_{u \in \mathcal{U}_T} J(t, e, u, T) = \text{Min}_{u \in \mathcal{U}} \frac{1}{2} \int_t^{t+T} L_C(\tau) d\tau,
\]
subject to:
- \( \dot{e} = f(e, u, t) = G(\theta) u(t) - Z_{ref}(t) \),
- \( u(\tau) \in \mathbb{R}^2 \), \( e(\tau) \in \mathcal{E} \), \( L(t) = e(\tau)^T Q e(\tau) + u(t)^T R u(t) \),
- \( e(t) = Z(t) - Z_{ref}(t) \), \( \tau \in [t, t+T] \),
where \( T > 0 \) is the prediction horizon, \( Q > 0 \) is a positive definite matrix (dim(\( Q \)) = 3 \times 3), and \( R \geq 0 \) is a positive semi-definite matrix dim(\( R \)) = 2 \times 2).

Problem (4) is a nonconvex optimization problem and must be solved online, with computational complexity, at each sampling time. Further, most of the existing algorithms may converge to a local solution. In this work, the optimization problem (4) will be converted to a convex quadratic optimization problem using the first-order Taylor approximation to solve it efficiently with a global minimum. Therefore, to obtain an explicit solution of NMPC, the prediction horizon \( T \) is divided into \( m \) subintervals (\( m \geq 2 \)) of equal width \( h \) such that \( T = mh \). The discrete-time version of the cost function (4) is given by
\[
f(\epsilon(t)) = \sum_{i=1}^{m-1} L(t+i+h) + L_T(t+mh),
\]
where \( e(t) \) is the initial tracking error and \( L(t+jh) = e(t+jh)^T Q e(t+jh) + u(t+jh)^T R u(t+jh) \).

To ensure the stability of the closed-loop system, the terminal cost should be included in the objective function (5) as follows
\[
L_T(t+mh) = (e(t+mh)^T P e(t+mh))
\]
where the adequate selection of the symmetric definite-positive matrix \( P \in \mathbb{R}^{3 \times 3} \) will be given later. So, the cost function (5) can be written as
\[
J(t, e(t), U(t), h) = \sum_{i=1}^{m-1} e(t+i+h)^T Q e(t+i+h) + e(t+mh)^T P e(t+mh).
\]

Note that, to avoid the cross product of control signals in (6), the following approximation is used:
\[
\begin{align*}
\sin(\theta(t+jh)) & \approx \sin(\theta(t)) \\
\cos(\theta(t+jh)) & \approx \cos(\theta(t)) \\
\Rightarrow & \quad G(\theta(t+jh)) \approx G(\theta(t))
\end{align*}
\]
Hence, the cost function (6) becomes
\[
J(t, e(t), U(t), h) = \{ \varphi(e(t), Z_{ref}(t)) + \sum_{i=0}^{m-1} e(t+i+h)^T Q e(t+i+h) + e(t+mh)^T P e(t+mh) + U(t)^T \tilde{R} U(t) \},
\]
where \( \varphi(e(t), Z_{ref}(t)) \) represents terms that are independent of the control vector \( U(t) \), \( \tilde{R} = \text{diag}(R, R, R, \ldots, R) \).

To evaluate the effect of the control effort on the predicted tracking error, the Euler approximation is used as follows
\[
e(t+jh) = e(t) + h \quad G(\theta) \sum_{i=0}^{m-1} u(t+jh) - h \quad \sum_{i=0}^{m-1} Z_{ref}(t+jh)
\]
where \( U(t)^T = [u(t)^T u(t+h)^T \ldots u(t+(m-1)h)^T] \), and
\[
\tilde{Z}_{ref}^T(t) = [Z_{ref}(t) Z_{ref}(t+h) \ldots Z_{ref}(t+(m-1)h)]^T
\]
Using (7), the cost function can be rewritten under the quadratic form
\[
J(e(t), U(t)) = \{ \varphi(e(t), \tilde{Z}_{ref}(t)) + 2h U^T(t) \tilde{G} \tilde{G}^T \tilde{G}^T \tilde{G} \tilde{G}^T \}
\]
where \( \tilde{G} = \text{diag}(G(\theta), G(\theta), G(\theta), \ldots, G(\theta)); \)
\[
\tilde{G}^T \tilde{G} = \text{diag}(Q, Q, \ldots, P);
\]
\[
\tilde{G}^T \tilde{G} = \text{diag}(G(\theta), G(\theta), G(\theta), \ldots, G(\theta)); \text{dim}(\tilde{G}) = 3 \times 2.
\]
Since \( \tilde{Q} > 0, P > 0, \) and \( \tilde{R} \geq 0, \) so the matrix \( \tilde{R} + h^2 \Pi \) is a positive definite matrix. The optimization problem (8) can be written under the following quadratic form
Min $J(t,e,U(t)) = \text{Min}_{\bar{y}(t) \in \mathbb{R}^p} \left\{ \varphi + 2h \bar{U}(t)\bar{G}^T\bar{Q} \left( \bar{I}_s \ e(t) - h \bar{I}_{3m} \bar{\hat{\theta}}_{ref} \right) + \bar{U}(t)(\bar{R} + h^2\Pi)(U(t)) \right\}.$ \hfill (9)

such that

- $e(t+jh) = e(t) + h G(\theta) L_2 U(t) - h L_3 \bar{\hat{\theta}}_{ref},$ for $j = 1, \cdots, m$
- $U(t) \in \mathbb{U}_T,$ and $E(t) \in \mathbb{E}_T.$

**CONSTRAINTS HANDLING**

Constrained spaces are being utilized mainly in the motion planning of robot manipulators. Hence, the contribution of this work is to formulate the motion planning problem of the mobile robot in the constrained environment as a constrained quadratic optimization problem. So, two kinds of constraints are considered:

**A. CONSTRAINTS ON CONTROL VECTOR**

All real-world control problems are subject to actuator constraints. Thus, the control signals $v(t)$ and $\omega(t)$ of the mobile robot (the speed and the steering angle) are bounded and cannot exceed some threshold levels [2]. In model predictive control, the predicted control signals should be constrained inside the admissible set. Thus, the constraint $U(t) \in \mathbb{U}_T$ can be expressed as linear constraints

$$A_1 \ U(t) \leq B_1$$

where $A_1 = \text{diag}(\bar{A}, \bar{A}, \cdots, \bar{A}), \ \bar{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, \ \ B_1 = \begin{bmatrix} \bar{B} \\ \vdots \\ \bar{B} \end{bmatrix},$ and

$$\bar{B} = \begin{bmatrix} v_{\max} \\ \omega_{\max} \\ -v_{\min} \\ -\omega_{\min} \end{bmatrix}.$$
IV. STABILITY AND FEASIBILITY ANALYSIS

The stability of the constrained model predictive control of nonlinear systems has been a main subject of research for many years. The closed-loop stability of the plant under the cost function with state and input constraints (12), can be guaranteed independently of the nonlinear mathematical model of the plant [36, 37]. This fact is realized by adding an equality constraint to the optimization problem that forces the tracking error to lay inside a terminal region at the end of the finite horizon. This terminal region should be a region of attraction for the nonlinear system controlled by a local controller.

A. STABILITY ANALYSIS

In this work, the Lyapunov method is used to prove the stability, and feasibility of the system closed by the model predictive control. The proposed controller will drive the state error to a terminal set. Inside this terminal set, a local controller will be used to drive the error state to the equilibrium point.

Lemma For the system (3), the compact set \( \Lambda_F = \{ e(t) \in \mathbb{R}^3, V_F(e) = e(t)^T P e(t) \leq a, \text{ for } \tau \geq T_p = t + mh \} \) is a positively invariant set with \( a = \delta_{df}^T \), under the local controller:

\[
u_F(\tau) = -\frac{1}{h}(G(\theta)^T P G(\theta))^{-1} G(\theta)^T P e(\tau), \tag{13}\]

Proof: In order to find the current control \( u_F(\tau) \) that reduces tracking error along with the interval \( \tau \in [T_p, t + h], \) we consider a quadratic performance index, that penalizes the tracking error under the form

\[
J_{ind}(e, u_F) = e(T_p + h)^T P e(T_p + h),
\]

where \( T_p = t + T \). The minimization of \( J_{ind}(e, u_F) \) with respect to \( u_F(\tau) \), using the first-order Taylor approximation of the one-step-ahead predicted tracking error, and by setting \( \frac{\partial J_{ind}}{\partial u_F} = 0 \) yields to:

\[
u_F(\tau) = -\frac{1}{h}(G(\theta)^T P G(\theta))^{-1} G(\theta)^T P e(\tau),
\]

where \( (G(\theta)^T P G(\theta))^{-1} \) is a positive-definite matrix.

Let’s take the following Lyapunov function

\[
V(e) = \frac{1}{2} e(t)^T P e(t),
\]

Given that the planned trajectory can be considered as a path or a set of points with \( \tilde{Z}_{ref} = 0 \), the time derivative of the Lyapunov function using the local controller (13) yields,

\[
\dot{V}(e) = e(t)^T P \dot{e}(t)
= -\frac{1}{h} e(t)^T P G(\theta)(G(\theta)^T P G(\theta))^{-1} G(\theta)^T P e(t),
\]

where \( \dot{V}(e) = P(\theta)(G(\theta)^T P G(\theta))^{-1} G(\theta)^T P \) and \( \dot{e}(\theta) = 0 \) are also positive definite. Indeed, the function \( s^T(G(\theta)^T P G(\theta))^{-1} G(\theta)^T P = 0 \) provided that \( e \neq 0 \). Thus, the set \( \Lambda_F = \{ e(t) \in \mathbb{R}^3, V_F(e) = e(t)^T P e(t) \leq h, \text{ for } \tau \geq T_p = t + mh \} \) is a positively invariant set under the local controller (13). The tracking error entering the set \( \Lambda_F \), then it will belong to this terminal set for \( \tau \geq T_p \). Then, the tracking error converges asymptotically to the origin.

Theorem 1 Assume that the initial solution of the optimization problem is feasible, \( e(t + h) \subseteq \mathbb{R} \), and \( u(t + (t + 1)h) \subseteq U \) for \( i = 1, m \). Suppose that the following inequality holds:

\[
G^T(\theta) P G(\theta) > \frac{1}{h} R. \tag{15}\]

Then, under the terminal controller (13), the optimal cost function \( J(t, e(t), h) \) is a control Lyapunov function (CLF) for the system (3). Consequently, the equilibrium point is asymptotically stable.

Proof: See the appendix.

B. FEASIBILITY ANALYSIS

To guarantee the closed-loop stability and convergence derived previously, we assumed that the predictions by the tail lie inside the constrained space. That is, the optimal solution at the time \( t \) is feasible:

\[
U^*(t) = [u^*(t) \ u^*(t + h) \ \ldots \ u^*(t + (m - 1)h)]^T,
\]

where \( u(t + ih) \subseteq U, i = 1, \ldots, m \).

\[
\hat{P}_{xy}(t + h) = \begin{bmatrix} P_{xy}(t + h) & P_{xy}(t + 2h) & \ldots & P_{xy}(t + mh) \end{bmatrix}^T,
\]

where \( \hat{P}_{xy}(t) \) is the predicted cartesian position of the mobile robot using the optimal solution \( U^*(t) \) with \( P_{xy}(t + ih) \in \mathbb{R}_+, i = 1, \ldots, m \).

We have to ensure the feasibility of the solution at the next step, i.e at the time \( t + h \) [36,37]. Hence, the next solution to the optimization problem has to be feasible (not necessarily optimal). The solution at the time \( t + h \) is:

\[
\bar{U}(t + h) = [u^*(t + h) \ \ldots \ u^*(t + (m - 1)h) \ u_F(t + mh)]^T,
\]

\[
\bar{P}_{xy}(t + h) = \begin{bmatrix} P_{xy}(t + h) & P_{xy}(t + 2h) & \ldots & P_{xy}(t + mh + h) \end{bmatrix}^T.
\]

Therefore, necessary and sufficient conditions for the predictions generated by \( \bar{U}(t + h) \) and \( \bar{P}_{xy}(t + h) \) to be feasible at time \( t + h \), whenever the solution of the optimization problem at time \( t \) is feasible, are:

i. \( P_{min} \leq \bar{P}_{xy}(t + mh + h) \leq P_{max} \) or \( \bar{P}_{xy}(t + mh + h) \in \mathbb{Z} \)

ii. \( u_{min} \leq u_F(t + mh) \leq u_{max} \) or \( u_F(t + mh) \subseteq U \).

For the solution given by (i), it is assumed that the predicted tracking error at time \( t + mh \) is feasible, then there exists a feasible subset of \( \mathbb{E} \), named \( \Lambda_F \) such that \( e^*(t + mh) \subseteq \Lambda_F \subseteq \mathbb{E} \). Hence, using the Lemma, the tracking error at the next time is also feasible \( e^*(t + mh + h) \subseteq \Lambda_F \subseteq \mathbb{E} \). Thus, the optimal solution will drive the state error from the set \( \mathbb{E} \) to the terminal set \( \Lambda_F \). To constraint the control signal inside its limits (ii) when the tracking error lies inside the terminal region \( \Lambda_F \), the following inequality should be satisfied

\[
|u_F(\tau)| \leq \frac{1}{h} \left\| (G(\theta)^T P G(\theta))^{-1} G(\theta)^T P \right\| e(\tau) \leq \bar{u},
\]

using assumption 6, the inequality becomes

\[
|e(\tau)| \leq \frac{h \bar{u}}{\lambda_{max}(G(\theta)^T P G(\theta))}, \tag{16}\]
where
\[ \Psi = (G(\theta)^T P G(\theta))^{-1} G(\theta)^T P, \hat{u} = \text{Max}(u_{\text{max}}, |u_{\text{min}}|). \]

Hence, the condition (16) is satisfied if the terminal region is chosen as
\[ \Lambda_F = \{e(t) \in \mathbb{E} \subset \mathbb{R}^3 \mid \forall \varepsilon(e) \leq \alpha, \text{such that } \alpha = \frac{\lambda_{\text{max}}(P) h^2 \hat{u}^2}{\beta_{\text{max}}(\Psi P^T)^{1/2}} \}. \]

Consequently, the feasibility can be achieved provided that the control problem admits a feasible solution at the initial time \( t \), in order to maintain the feasibility of all solutions. Indeed, it is a well-known fact that the feasibility at one time instant leads to a feasible solution at the next instant, and the value of the cost function decreases [36, 37].

**Theorem 2** Suppose that the assumptions 1-3 hold, the matrix \( P \) is chosen such that the inequality (15) holds. Starting from a feasible initial position \( Z(t) \in \mathbb{Z} \) and a feasible control prediction sequence \( U(t) \in U_T \), then the solution \( U^*(t) \) of the quadratic problem (12) steers any initial tracking error from \( \mathbb{E} \) to \( \Lambda_F \). Further, the terminal control signal \( u_F(t) \) will steer the tracking error to the equilibrium point \( e^* \) for \( t \geq (t + T + h) \).

**Proof:** Assume that the initial control sequence \( U(t) \) and the initial state are feasible at time \( t \). Using the constrained QP problem (12), the optimal solution \( U^*(t) \) is feasible. From theorem 1, the control Lyapunov function is a decreasing function, then the polyhedron set \( Z_T \) is a positively invariant set and the tracking error will converge to the terminal region within the interval \([t, t + T]\). For \( t \geq (t + T + h) \), the terminal controller \( u_F(t) \) will steer the tracking error to the equilibrium point \( e^* \). Thus, the equilibrium point of the closed-loop system is asymptotically stable.

According to the feasibility or unfeasibility of the reference trajectory \( Z_{\text{ref}}(t) \), we have two cases:

**Case 1:** \( Z_{\text{ref}}(t) \in \mathbb{Z} \) (feasible trajectory)
In this case, the terminal region is an attractive region of the equilibrium point which is the origin. Indeed, the solution provided by the QP algorithm (12) is feasible, since \( Z(t) \in \mathbb{Z} \) and \( Z_{\text{ref}}(t) \in \mathbb{Z} \), the equilibrium point is \( e^* = \lim_{t \to \infty} Z(t) - Z_{\text{ref}}(t) = 0 \). The dynamic of the tracking error inside the terminal region is given by
\[ \dot{e}(t) = -\frac{1}{h} H e(t) \text{ for } t \geq t + mh \implies e(t) = e(t + mh) e^{-\frac{1}{h}t}, \]
where \( H = (G(\theta)^T P G(\theta))^{-1} G(\theta)^T P \). Hence, we have
\[ \lim_{t \to \infty} e(t) = e(t + mh) e^{-\frac{1}{h}t} = e^* \to 0 \]

**Case 2:** \( Z_{\text{ref}}(t) \notin \mathbb{Z} \) (unfeasible trajectory)
In this case, \( Z(t) \in \mathbb{Z} \) and \( Z_{\text{ref}}(t) \notin \mathbb{Z} \), the equilibrium point is \( e^* = Z(t + T) - Z_{\text{ref}}(t + T) = Z_s - Z_{\text{ref}} = e_s \neq 0 \), where \( Z_s \) is the value of \( Z(t) \) on the border of the feasible set \( \mathbb{Z} \). Thus, the solution will converge to an equilibrium point defined by \( e_s \neq 0 \).

In conclusion, the solution to the convex optimization problem (12) steers asymptotically the tracking error to the equilibrium point \( e^* \) defined by
\[ e^* = \begin{cases} 0 & \text{if } Z_{\text{ref}}(t) \in \mathbb{Z} \\ e_s & \text{if } Z_{\text{ref}}(t) \notin \mathbb{Z} \end{cases} \]

**V. SIMULATION AND EXPERIMENTAL RESULTS**
This part illustrates the effectiveness of the proposed control algorithm, given by the QP (12), through simulation using MATLAB software (201b) and a real-time implementation using the Khepera III platform [39]. This section deals with the different performances achieved by the proposed algorithm in terms of:
1. Achieved control performances (tracking performance and feasibility of the solution).
2. Processing time comparison.
3. Real-time implementation of the approximated NMPC.

Note that, for the simulation part, the control parameters \( Q, R, \) and \( h \) have been tuned in order to determine the optimal values that achieve reasonable tracking performances. It is noticed that the step time \( h \) is a key parameter that should be chosen carefully to validate the Euler approximation used in this work. Therefore, many simulations have been performed to find the adequate control parameters that achieve admissible tracking performances.

**A. CONTROL PERFORMANCE RESULTS**
For this part, we have stated two different scenarios [19]:

I. Tracking eight-shaped trajectory. In this case, constraints on \( y \)-axis limits and control signals are included in the optimization problem.

II. Tracking a circular-shaped trajectory where the admissible region is a square-shaped space. In this case, both state constraints and input constraints are included in the optimization problem.

For simplicity, we assume that the matrices \( Q \) and \( R \) are diagonals. After some tuning, the control parameters are chosen as
\[ h = 0.1 s; Q = \begin{bmatrix} 10^4 & 0 & 0 \\ 0 & 10^4 & 0 \\ 0 & 0 & 1 \end{bmatrix}; Ts = 0.01 s; P = 2 Q; d = 0.2; T = 0.4 s; m = 4; R = \begin{bmatrix} 10^{-4} & 0 \\ 0 & 10^{-2} \end{bmatrix}. \]
Note that, with these control parameters, the condition (15) is satisfied.

a. **First scenario:** The eight-shaped desired trajectory defined by
\[ \begin{align*}
\{x_{\text{ref}}(t) &= x_c + R_1 \sin(\omega_1 t) \\
y_{\text{ref}}(t) &= y_c + R_2 \sin(\omega_2 t) \}
\end{align*} \quad t \in [0, t_f]; \]
where \( (x_c = 3, y_c = 3) \) represent the center of the eight-shaped trajectory. The reference trajectory of the orientation angle is deduced from
\[ \theta_{\text{ref}}(t) = \tan^{-1} \left( \frac{y_{\text{ref}}}{x_{\text{ref}}} \right), \quad t \in [0, t_f] \]
Trajectory parameters are
\[ R_1 = R_2 = 2m; \quad \omega_r = 0.05 \text{ rad/s}; \quad \omega_1 = \omega_r; \quad \omega_2 = 2 \omega_r; \quad t_f = 42ns. \]

Because of the bounded velocity of each wheel, constraints on the control signals should be included in the optimization problem. In this simulation part, the same bounds provided by the authors in [2] are used, which are
\[ \Omega = \{ 0 \leq v(t) \leq 0.3 \text{ m/s}; -0.5 \text{ rad/s} \leq \omega(t) \leq 0.5 \text{ rad/s} \}. \]

The constrained space is delimited over the y-axis, where
\[ F_R = \{ y(t) \in \mathbb{R}; \quad 1.5m \leq y(t) \leq 4.5m, \quad t \in [t, t + T] \}. \]

Figure 3 shows the good tracking performance achieved by the proposed controller. In this scenario, the mobile robot is represented by the black triangle and the moving target by a green ball. One can see that the desired eight-shaped trajectory is precisely tracked inside the constrained space. The applied control signals, which are inside the saturation limits, are depicted in Figure 4 for \( t \in [0, 75s] \). Small oscillations of the steering angle are noticed during the start-up.

**FIGURE 3.** Tracking performance for the first scenario.

**FIGURE 4.** The time variation of the applied control signals.

b. Second scenario: In this case, the WMR is required to follow a circle-shaped trajectory generated by the planner. The circle-shaped trajectory is defined as follows
\[
\begin{align*}
    x_{ref}(t) &= x_c + R_o \sin(\omega_o t) \\
    y_{ref}(t) &= y_c - R_o \cos(\omega_o t) \\
    \theta_{ref}(t) &= \omega_o t
\end{align*}
\]
where \( R_o = 2m; \quad \omega_o = 0.05 \text{ rad/s} \).

The mobile robot has to navigate inside the restrained region; therefore, it is required to find the control vector \( U(t) \) to track the desired trajectory only inside the free space. Thus, the following constraints have been included in the optimization problem
\[ F_R = \{ R_{xy}(t) \in \mathbb{R}^2; \quad 1.2m \leq x(t) \leq 4.8m \text{ and } 1.5m \leq y(t) \leq 4.5m, \quad t \in [t, t + 0.4s] \}. \]

The initial position of the WMR is \( [2m \quad 4m \quad 0m]^T \) and the desired trajectory starts at \( [x_{ref}(0) = 3m \quad y_{ref}(0) = 1m]^T \).

The tracking performance of the WMR under the control algorithm (12) is shown in Figure 5.

It is observed that when the desired trajectory is inside the feasible region, the robot follows it precisely. However, when the reference trajectory is outside the desired region (unfeasible), the mobile robot navigates close to the border of the admissible region.

Figure 6 illustrates the applied control signals. So, the linear speed and the steering angle are inside the saturation limits. Here also oscillations are noticed at the beginning of the movement. It is shown in Figure 7 the tracking error converges to zero when the reference trajectory is inside the desired region and is different from zero when it is outside the desired region. Further, to show the effect of the feasibility of the desired trajectory, Figure 8 shows the tracking performance for two different cases: The desired trajectory is feasible, completely inside the restricted region (\( R=1m \)) and for the second case the desired trajectory is unfeasible, completely outside the restricted region (\( R=3m \)), but the state constraints are satisfied.

**FIGURE 5.** Tracking performance for the second scenario.

**FIGURE 6.** Optimal control signals.

B. EXECUTION TIME COMPARISON

The proposed NMPC formulation based on the approximation of NMPC (ANMPC) is compared against the processing time
of the NLP formulation (4) and the algorithm that uses the CasADI tool to fasten the operation of NMPC [40]. The NLP problem, termed by NMPC in this comparison part, is solved using MATLAB “fmincon” solver with SQP. To perform a valid comparison, we have used these three algorithms, with the same sampling time, prediction horizon, and control parameters. Further, regulation to the same target in free space has been conducted for \( t \in [0,2s] \). During simulation, we have noticed that the initial guess of the algorithm affects the processing time for the first steps of the simulation. To this end, Monte Carlo simulation is utilized in these simulations with ten different runs. The initial vector guess is sampled from the normal distribution \( \mathbb{N}(0,1) \).

Observing this comparison demonstrates that the ANMPC-based approach is an attractive alternative algorithm to solve the NMPC problem efficiently.

### Table 1: The Mean and Standard Deviation of the Execution Time per Iteration

| Method  | Processing time per iteration | Standard deviation per iteration |
|---------|-------------------------------|--------------------------------|
| NMPC    | 0.1062 s                      | 0.1086                         |
| CASADI  | 0.0290 s                      | 4.2564e-18                     |
| ANMP    | 0.0029 s                      | 0.0148                         |

### C. Experimental Results

In this second part, we consider the real-time implementation of the proposed algorithm on a real mobile robot platform. The experiment has been conducted using a TOSHIBA laptop with Intel Core i7 processors. The platform is the mobile robot Khepera III (Figure 10), which is a differential drive mobile robot [39]. Khepera platform is an automated differential drive guided mobile robot designed and equipped for autonomous tasks. It is provided with infrared, and Ultrasonic sensors for navigation, and encoders for localization. The distance between the left and right wheels is equal to \( L = 88.4\text{mm} \). The diameter of the wheel is 41mm. The connection between the platform and the Laptop is established via Bluetooth communication technology. Since MATLAB functions support communication with devices via Bluetooth, the MATLAB software has been also used in this work to implement the proposed algorithm.

The scenario of tracking a circular-shaped trajectory in constrained space is adopted in this experimental part. The reference trajectory is the same as the one used in the simulation part, with the following parameters

\[
x_c = 0; y_c = 0; R_o = 650 \text{ mm}, \omega_o = 2 \times \pi/35; t \in [0s,146s].
\]
Constraints on control signals are:

\[ v_{\text{max}} = 300 \text{ mm/s}; v_{\text{min}} = 0; \omega_{\text{max}} = 5 \text{ rad/s}; \omega_{\text{min}} = -5 \text{ rad/s}. \]

Space constraints are formulated as follows:

\[ F_R = \{ x(t) \in \mathbb{R}^2; -580 \text{ mm} \leq x(t) \leq 580 \text{ mm and } -580 \text{ mm} \leq y(t) \leq 580 \text{ mm}, \ t \in [0s,146s] \} \]

Because of the presence of noisy measurements, and also due to the inaccuracies that come from residual errors (quantization of the velocity controls), the constrained space used by the control algorithm is diminished with regards to the real constrained region (hard constraints). Consequently, for more safety, the soft constraints used in the algorithm are as follows:

\[ F_S = \{ x(t) \in \mathbb{R}^2; -550 \text{ mm} \leq x(t) \leq 550 \text{ mm and } -550 \text{ mm} \leq y(t) \leq 550 \text{ mm}, \ t \in [0s,146s] \} \] where \( F_S \subset F_R \).

The robot starts from the feasible region, which is given by the position: \([x(0) \ y(0) \ \theta(0)] = [0 \ 0 \ 0]^T\) and the desired trajectory (also feasible) starts at \([x_{\text{ref}}(0) = 650 \text{ mm; } y_{\text{ref}}(0) = 0 \text{ mm; } \theta_{\text{ref}}(0) = 0^\circ]\). The numerical values of control parameters for this experiment are:

\[ Q = 10^3 I_3; R = 10^{-3} I_2; h = 0.1 \text{s; } d = 80 \text{ mm; } m = 4; \ T = 0.4 \text{s} \]

Figures 11-13 show the experimental results obtained by solving the constrained quadratic optimization problem given by (12).

![FIGURE 11. Trajectory tracking performance in the constrained square.](image1)

Figures 11 displays the desired trajectory (circle), the restricted space region (square), and the trajectory of the mobile robot. The tracking of the reference trajectory inside the restricted space is accurate and when the desired trajectory is outside the restricted space, the mobile robot stays close to the border of the square.

Figure 11 shows the obtained solution from the constrained optimization problem. Both controls signal, linear velocity, and steering velocity are inside the saturation limits.

The command signals sent to WMR are the velocities of the right and left wheels \((v_R, v_L)\), which are deduced from the following equation:

\[
\begin{bmatrix}
\dot{v}(t) \\
\dot{\omega}(t)
\end{bmatrix} = \begin{bmatrix}
\frac{1}{2}(v_R(t) + v_L(t)) \\
\frac{1}{2}(v_R(t) - v_L(t))
\end{bmatrix} \Rightarrow
\begin{bmatrix}
v_R(t) \\
v_L(t)
\end{bmatrix} = T \begin{bmatrix}
\dot{v}(t) \\
\dot{\omega}(t)
\end{bmatrix}
\]

Figure 12 illustrates the Cartesian positions of the mobile robot along the \(x\)-axis and \(y\)-axis. One can see clearly that the mobile robot navigates practically inside the restricted region and that the constrained optimization problem has forced the mobile robot to move inside the tolerated space, which is a square for this case.

Finally, we note that the main drawback of the proposed control algorithm is to find adequate tuning control parameters \((h, Q, R, \text{etc.})\) that meets the desired tracking performance of different tracking trajectories.

![FIGURE 12. Linear velocity and angular velocity.](image2)

![FIGURE 13. The current evolution of the WMR along x-y axes.](image3)

VI. CONCLUSION

In this work, the approximated nonlinear model predictive control has been applied to the autonomous navigation of a differential drive mobile robot in a constrained region. Euler approximation was used to overcome the nonconvex and nonlinear optimization problem in the control design. The stability of the closed-loop system has been provided using the Lyapunov method. The autonomous navigation of the mobile robot in the constrained region was formulated as a quadratic optimization problem or QP where both physical constraints (control vector) and space constraints were included in the design of the controller. Therefore, all constraints have been explicitly included in the optimization problem. Thus, the obtained constrained optimization problem is convex with the fast update and many numerical solutions exist that solve this kind of optimization problem efficiently. It was proven that the solution is feasible and the trajectory tracking error was...
bounded and may converge to the origin when the desired trajectory lies inside the feasible region.

The proposed method offered many advantages over existent nonlinear methods: simplicity, fast computation, and its ability to include different constraints in the controller design. Simulations and experimental results demonstrated the effectiveness of the proposed control algorithm and good tracking performances were achieved.

Future work will focus on the application of the proposed approach to three-dimensional vehicles for instance aerial unmanned vehicles.

APPENDIX

Proof of Theorem 1

Let the initial solution \( U'(t) \) be feasible, and the equivalent optimal cost function (9) is given under the form of

\[
J(t, e(t), U'(t), h) = J(t, e(t), U'(t), h)
\]

where

\[
U'(t) = [u'(t) \quad u'(t + h) \quad u'(t + (m - 1)h)]^T
\]

\[
\bar{U}(t + h) = [u'(t + h) \quad u'(t + (m - 1)h)]
\]

\[
J(t + h, e(t + h), \bar{U}(t + h), h) - J(t, e(t), U'(t), h)
\]

\[
= -L(e(t), u'(t)) + P_r(t, m)
\]

\[
P_r(t, m) = L(e(t + mh), u_r(t + mh)) + e(t + mh)^T P e(t + mh + h) - e(t + mh)^T P e(t + mh)
\]

which can be transformed into the quadratic form using the terminal controller

\[
P_r(t, m) = e(t + mh)^T \left\{ G - 2Q \begin{bmatrix} G^T Q & -G^T P \end{bmatrix} + Q \begin{bmatrix} G^T Q & -G^T P \end{bmatrix} + G^T P G \begin{bmatrix} G^T Q & -G^T P \end{bmatrix} \right\} e(t + mh)
\]

The following inequality is obtained

\[
G^T \prod G = G^T Q^2 G - G^T P G + \frac{R}{h^2} < 0
\]

It follows from the definition (1), the following inequality is obtained

\[
\prod < 0 \Leftrightarrow G^T Q^2 G - G^T P G + \frac{R}{h^2} < 0
\]

Consequently, the function \( P_r(t, m) \) is negative. This inequality can be simplified if we take diagonal matrices as \( P = p I_2; Q = q I_2 \), and \( R = r I_2 \), then the previous inequality becomes

\[
p > q + \frac{r}{h^2 (2^q + 1)}
\]

The cost \( J(t + h, e(t + h), \bar{U}(t + h), h) \) is feasible and not necessarily optimal, and we have

\[
J(t + h, e(t + h), U'(t + h), h) < J(t + h, e(t + h), \bar{U}(t + h), h)
\]

Thus, the optimal cost (6) is decreasing since

\[
f(t + h, e(t + h), U'(t + h), h) - f(t, e(t), U'(t), h) < 0
\]

Further, all sublevel sets of the optimal cost are bounded. Consequently, the optimal cost function is a Lyapunov function [41].

REFERENCES

[1] B. Siciliano and O. Khatib (Eds.), Springer handbook of robotics, 2nd ed., Springer-Verlag, 2016.
[2] G. Oriolo, A. De Luca, and M. Vendittelli, “WMR control via dynamic feedback linearization: design, implementation, and experimental validation,” IEEE Trans. On Control Systems Technology, vol. 10, no. 6, pp. 835-852, 2002.
[3] M. Bloch and S. Drakunov, “Tracking in the nonholonomic dynamic system via sliding modes,” in Proc. 34th CDC, New Orleans, L.A., pp. 2103-2106, 2002.
[4] R. Fierro and P. L. Lewis, “Control of a nonholonomic mobile robot: Backstepping kinematics into dynamics,” in Proc. 34th CDC, New Orleans, L.A., pp. 3805-3810, 1995.
[5] S. M. Ahmadi, M. B. Taghadosi, and A. Haqshenas “A state augmented adaptive backstepping control of wheeled mobile robots,” Transactions of the Institute of Measurement and Control, vol. 43, no.2, p.434-450, 2021.
[6] Shu Li, Liang Ding, Haibo Gao, Chao Chen, Zhen Liu, and Zongquan Deng, “Adaptive neural network tracking control-based reinforcement learning control for wheeled mobile robots with skidding and slipping,” Neurocomputing vol.283, pp. 20-30, 2018.
[7] Liang Ding, Lan Huang, Shu-Li, Haibo Gao, Huichao Deng, Yuankai Li, and Guanqiu Liu, “Definition and application of variable resistance coefficient for wheeled mobile robots on deformable terrain,” IEEE Transactions on Robotics, vol. 36, no.3, pp. 894 – 909, May 2020.
[8] R. Findelen, F. Allgöwer and L. T. Biegler (Eds) “Assessment and future directions of nonlinear model predictive control,” Lecture Notes in Control and Information Sciences 358, Heidelberg, Springer, 2007.
[9] L. Magni, D. M. Raimondo and F. Allgöwer (Eds), “Nonlinear model predictive control: towards new challenging applications,” Lecture Notes in Control and Information Sciences 384, Heidelberg, Springer, 2009.
[10] S. Subramanian, Y. Abdessalam, S. Lucia, and S. Engell, “Robust tube-enhanced multi-stage NMPC with stability guarantees,” IEEE Control systems letters, vol. 6, 2022.
[11] L. Xie, H. Su, S. Lu, and L. Xie, “Linear mismatched model-based offset-free MPC for nonlinear constrained systems with both stochastic and deterministic disturbances and its application to CSTR,” IEEE Access, vol. 6, pp.69172-69184, 2018.
[12] F. Zhou, H. Peng, G. Zhang, X. Zeng, and X. Peng, “Robust predictive control algorithm based on parameter variation rate information of functional-coefficient ARX model,” IEEE Access, vol.7, pp.27231-27243, 2019.
[13] Nilay Saraf, Mario Zanon, Alberto Bemporad, “A Fast NMPC Approach based on Bounded- Variable Nonlinear Least Squares,” 6th IFAC Conference on Nonlinear Model Predictive Control, Madison, WI, USA, August 19-22, 2018.
[14] A. Nurkanovié, A. Zanelli, S. Albrecht, M. Diehl, “The Advanced Step Real-Time Iteration for NMPC,” in Proceedings of 58th IEEE Conference on Decision and Control (CDC), Nice, France, 11-13 Dec. 2019.
[15] A. Zanelli, R. Quirynen, J. Jerez, and M. Diehl, “A homotopy-based nonlinear interior-point method for NMPC,” in Proceedings of the 20th IFAC World Congress,2017, pp. 13188–13193, Toulouse, France.
[16] H. Deng, and T. Ohtsuka, “A parallel Newton-type method for nonlinear model predictive control,” Automatica, vol.109, 2019.
[17] R. Verschueren, G. Frison, D. Kouzoupis, N. Dajkjer, A. Zanelli, R. Quirynen, and M. Moritz, “Towards a modular software package for embedded optimization,” 6th IFAC Conference on Nonlinear Model Predictive Control NMPC, vol.51, Issue 20, pp. 374-380, Madison, Wisconsin, USA, 19–22 August 2018.
[18] H. Deng & T. Ohtsuka, “ParNMPC – a parallel optimization toolkit for real-time nonlinear model predictive control,” International Journal of Control, 2020.
[19] T. P. Nascimento1, C. E. T. Dorea, and L. M. G. Gonçalves, “Nonlinear model predictive control for trajectory tracking of nonholonomic mobile robots: A modified approach,” International Journal of Advanced Robotic Systems, pp.1-14, Jan-Feb 2018.
[20] A. Zanelli, J. Kallick, H. M. Eldeeb, G. Frison, C. M. Hackl, and M. Diehl, “Continuous control set nonlinear model predictive control of reluctance synchronous machines,” IEEE Trans. On Control Systems Technology, pp.1-12, 2021.

[21] A. S. Conceicao, A. P. Moreira, and P. J. Costa, “A nonlinear model-predictive control strategy for trajectory tracking of a four-wheeled omnidirectional mobile robot,” Optimal Control Applications and Methods, vol.29, pp.335-352, 2008.

[22] H. N. Huyash, O. Verlinden, A. V. Wouwer, “Comparative application of model predictive control strategies to a wheeled mobile robot,” J. Intell. Robot Syst. vol.87, pp.81–95, 2017.

[23] U. Rosolia, S. De Bruyne, and A. G. Alleyne, “Autonomous vehicle control: A nonconvex approach for obstacle avoidance,” IEEE on Control Systems Technology, vol.25, no.2, pp.469-484, 2017.

[24] I. Sanchez, A. D’Jorge, G.V. Raffo, A.H. Gonzalez, and A. Ferramosca, “Nonlinear model predictive path following controller with obstacle avoidance,” Journal of Intelligent & Robotic Systems, vol.16, no.102, April 2021.

[25] K. Worthman, M.W. Mehrez, M. Zanon, G. I. Mann, R.G. Gosine, and M. Diehl, “Model predictive control of nonholonomic mobile robots without stabilizing constraints and costs,” IEEE Trans. On Control Systems Technology, vol. 24, no. 4, pp.1394-1406, July 2016.

[26] M. Seder, M. Baotic, and I. Petrivic, “Receding horizon control for convergent navigation of a differential drive mobile robot,” IEEE Trans. On Control Systems Technology, vol. 25, no. 2, pp.653-660, March 2017.

[27] X. Du, K. K. K. Htet, and K.K. Tan, “Development of a genetic-algorithm based nonlinear model predictive control scheme on velocity and steering of autonomous vehicles,” IEEE Trans. On Industrial Electronics, vol.63, no.11, pp.6970-6977, Nov. 2016.

[28] Z. Wang, G. Li, H. Jiang, Q. Chen, and H. Zhang, “Collision-free navigation of autonomous vehicles using convex quadratic programming-based model predictive control,” IEEE/ASME Transactions on Mechatronics, vol.23, no. 3, pp.1103-1113, June 2018.

[29] M.N. Zeilinger, C. N. Jones, and M. Morari, “Real-time suboptimal model predictive control using a combination of explicit MPC and online optimization,” IEEE Trans. On Automatic Control, vol. 56, no.3, pp.1524-1534, 2011.

[30] V. Valls, and D. J. Leith, “A convex optimization approach to discrete optimal control,” IEEE Trans. On Automatic Control, vol.64, no.1, pp.35-50, Jan. 2019.

[31] S. Boyd, and L. Vandenberghe, Convex optimization, Cambridge university press, 2004.

[32] Y. Wang and S. Boyd, “Fast model predictive control using online optimization,” IEEE Trans. On Control Systems Technology, vol.18, no.2, pp.267-278, March 2010.

[33] R. Hedjar, “Application of nonlinear rescaling method to model predictive control,” International Journal of Control, Automation and Systems (IJCAS), vol. 8, no.4, pp.762-768, 2010.

[34] Maury and J. Vend, “Handling of contacts in crowd motion simulations,” in Traffic and Granular Flow 07. Berlin, Heidelberg, Springer, vol.1, pp.171-180, 2007.

[35] R. Hedjar and M. Bounkhel, “Real-time obstacle avoidance for a swarm of autonomous mobile robots,” International Journal of Advanced Robotic Systems (IJARS), vol.11, no.67, pp.1-12, 2014.

[36] D. Q. Mayne, and P. Falugi, “Stabilizing conditions for model predictive control,” International Journal of robust and nonlinear control, vol.29, issue 4, pp.894-903, March 2019.

[37] B. Rawlings James, D. Q. Mayne, M. Diehl, Model predictive control: theory computation and design, 2nd Ed., by Nob Hill Publishing, LLC, Madison, Wisconsin, 2018.

[38] Inga J. Wolf, Wolfgang Marquardt, “Fast NMPC schemes for regulatory and economic NMPC – A review,” Journal of Process Control, vol. 44, pp.162-183, 2016.

[39] www.k-team.com

[40] https://github.com/MMehrez/MPC-and-MHE-implementation-in-MATLAB-using-Casadi.git

[41] H. K. Khalil, Nonlinear control, 1st edition, Pearson, 2014.