Tightening polytopic constraint in MPC designs for mobile robot navigation

Ngoc Thinh Nguyen
Institute for Robotics and Cognitive Systems
University of Luebeck
Luebeck, Germany
nguyen@rob.uni-luebeck.de

Georg Schildbach
Institute for Electrical Engineering in Medicine
University of Luebeck
Luebeck, Germany
georg.schildbach@uni-luebeck.de

***Abstract*—We introduce a new implementation of a standard linear polytopic constraint which allows not only to validate the constraint itself but also to encourage a distance from the system to the boundary (i.e. the edges of the polytope), and hence, is called tightening polytopic constraint. We also present the MPC designs for mobile robots to navigate within a polytopic working zone towards the goal, which employ the proposed tightening constraint. Extensive simulations and comparison with the potential field approach show the effectiveness and advantages of the proposed methods.**

***Index Terms*—Polytopic constraint, Tightening, Model Predictive Control (MPC), Differential wheeled mobile robot.**

I. Introduction

Fundamental approaches for robotic path planning can be divided into four groups: *(a)* geometric approaches, based e.g. on meshes, grids, or a visibility graph [1]; *(b)* sampling-based approaches, using e.g. randomly exploring trees [2]; *(c)* curve fitting approaches, based e.g. splines or clothoids [3], [4], and *(d)* optimization-based approaches [5]. Regarding the latter category, Model Predictive Control (MPC) has been used in numerous recent works [6]–[9], including autonomous aircraft [10], drones [11], race cars [12], parking cars [13], and robots [14]. In this paper, we present an MPC formulation which provides an integrated approach for the navigation (path planning and control) of a mobile robot using real-time numerical optimization.

In many applications, the 'optimal' path between a start and a goal point is difficult to specify in mathematical terms. Important criteria include the length of the path, its smoothness and curvature, and keeping a certain distance from obstacles that are in its way. Potential fields have proven a useful tool that yields a good compromise between these conflicting objectives. In particular, they lead to smooth paths that keep a safe distance to obstacles whenever possible, without unnecessarily restricting the solution space. They work well in combination with most path planning approaches. However, they cause difficulties in optimization-based approaches, because they often render the resulting constraints non-convex [13]. This makes the optimization problem harder to solve and may cause the solver to converge to a local optimum [15].

Therefore, this paper proposes a new convex constraint formulation, whose effect is similar to that of a potential field. The main idea is that a distance of the planned path to the constraints is expressed with a slack variable, which is rewarded with a convex term in the objective function. The effectiveness of this approach is demonstrated in a small simulation example, for the navigation towards a desired goal of a generic differential wheeled mobile robot inside a convex polytopic region.

A. Contributions

1) We propose the tightening formulation of a standard polytopic constraint which allows to simultaneously validate the constraint and perform an inward offset to the original polytope. The offset distance is not specified but is encouraged to be large in order to keep the system far away from the boundary.

2) Two MPC designs for mobile robot navigation to the desired goal are formulated using the aforementioned tightening polytopic constraint. The first one aims to maximize the offset value while the latter one is to maintain a desired offset value by adding linear and quadratic terms in the cost function, respectively.

3) We present a comparison between the proposed MPC designs and the MPC controller using repulsive potential field [16] which shows various advantages of our MPC designs: keeping the system far away from the boundary, convergence without being stuck at local minima, clear interpretation of the parameters and low computation burden, in comparison with the potential field approach.

B. Outline and Notations

The remaining paper is organized as follows. Section II introduces the kinematics model of the mobile robot and the considered navigation problem. Next, Section III presents the main contributions of this paper, i.e. the tightening polytopic constraints, the normalization process and their usages in MPC designs. The simulation results and the comparisons with existing methods are given in Section IV. Finally, Section V draws the conclusions and future works.

Throughout the paper, $\mathbf{0}$ and $\mathbf{1}$ represent vectors containing all zeros and ones, respectively, whose dimensions are clear from the context. $\boldsymbol{I}_n$ is the $n \times n$ identity matrix. $\mathcal{P}(A, b)$ represents a 2D polytope: $Ap \leq b$, , where $A \in \mathbb{R}^{n\times 2}$ and $b \in \mathbb{R}^n$ for a point $p \in \mathbb{R}^2$. $\mathbb{R}_{\geq 0}$ is the set of non-negative real

978-1-6654-1496-8/21/$31.00 ©2021 IEEE

numbers while $\mathbb{R}^n_{\geq 0}$ represents the set of vectors consisting of n non-negative real elements.

II. Problem formulation

In this paper, we consider the navigation problem of a differential wheeled robot (DWR) whose kinematics model is given as follows [6]–[8]:

$$\dot{x} = f(x, u), \tag{1}$$

in which, the state vector $x \triangleq (\xi,\ \eta,\ \theta)^\top \in \mathbb{R}^3$ consists of the 2D position $(\xi,\ \eta)$ and the direction angle θ. The input vector is $u \triangleq (v,\ \omega)^\top \in \mathbb{R}^2$ with v and ω, the linear and angular speeds of the robot, respectively. The nonlinear function $f : \mathbb{R}^3 \times \mathbb{R}^2 \to \mathbb{R}^3$ is given by:

$$f(x, u) = (v\cos\theta,\ v\sin\theta,\ \omega)^\top. \tag{2}$$

We further denote the transformation from the state $x \triangleq (\xi,\ \eta,\ \theta)^\top$ to the position $p \triangleq (\xi,\ \eta)^\top$ by :

$$p = Mx,\ M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \tag{3}$$

The position p of the robot (1) is restricted in an obstacle-free zone, represented by a polytope in 2D plane:

$$p \in \mathcal{P}(A, b) \triangleq \{p \in \mathbb{R}^2 |\ Ap \leq b\}, \tag{4}$$

with $A \in \mathbb{R}^{m\times 2}$, $b \in \mathbb{R}^m$. Note that, a typical DWR system usually has constraints on the 2D pose p but not the orientation θ [6]–[8].
Also, the input u from (1) is bounded as follows:

$$u_{\min} \leq u \leq u_{\max}, \tag{5}$$

in which, $u_{\min}$ and $u_{\min}$, both in $\mathbb{R}^2$, gather the minimum and maximum values of the speeds (linear and angular), respectively. It is also necessary for the input constraint set to contain the zero input value $u_e = (0, 0)^\top$:

$$u_{\min} \leq u_e \leq u_{\max}. \tag{6}$$

Then, we consider a desired position $p_e \in \mathbb{R}^2$ within the state constraint set $\mathcal{P}(A, b)$:

$$Ap_e \leq b, \tag{7}$$

with (A, b) as in (4). We study the navigation problem of the system (1) to the desired position p_e from (7) while satisfying the state and input constraints (4), (5). Furthermore, as the main motivation of this work, while navigating to the desired position p_e, the robot should try to stay away from the edges of the operating zone defined as the polytope $\mathcal{P}(A, b)$ from (4). We show that the proposed novel methods provide the same effect of being pushed away from obstacles and boundaries when using potential fields and barrier functions [16] but with a simpler formulations consisting of only linear constraints and additional linear/ quadratic terms in the cost functions of the MPC controllers.

III. MPC designs for DWR navigation

This section presents the main contributions of this work, i.e. the tightening polytopic constraint and its uses in MPC design for DWR navigation. The core ideas can be briefly introduced as follows:

1) Deriving a formulation which allows to tighten the polytopic constraint $Ap \leq b$ (4) by an inward offset d (but not specifying any value to d).
2) Encourage a big value of the offset value d in the cost function of the MPC controller while ensuring that the state is still within the offset polytope.

We also define a process called "normalization of the half-space representation" in order to treat the offset d equally for all the edges of the polytopes.

A. Tightening polytopic constraint

Regarding the state constraint $p \in \mathcal{P}(A, b)$ (i.e. $Ap \leq b$ as in (4)), we propose a formulation which allows not only to check $Ap \leq b$ itself but also to rate the distance between the position p and its closest edge (among those of the polytope $\mathcal{P}(A, b)$). For doing this, we first represent the polytope $\mathcal{P}(A, b)$ with an inward offset $d \in \mathbb{R}_{\geq 0}$, i.e. $\mathcal{P}(A, b - d\mathbf{1})$ and then check if there exists any value of d such that $p \in \mathcal{P}(A, b - d\mathbf{1})$. The "so-called" tightening constraint is mathematically defined as follows:

$$\begin{cases} Ap \leq b - d\mathbf{1}, \\ 0 \leq d \leq d_{\max}, \end{cases} \tag{8}$$

where $d_{\max} \in \mathbb{R}_{\geq 0}$ is the maximum offset value and is chosen such that the maximally shrunk polytope still contains the desired position p_e from (7), i.e. $p_e \in \mathcal{P}(A, b - d_{\max}\mathbf{1})$:

$$Ap_e \leq b - d_{\max}\mathbf{1}. \tag{9}$$

Note that, the i^{th} element of the one vector $\mathbf{1}$ as in (8)–(9) can be set as 0 in order to completely remove the offset effect on the corresponding i^{th} element of the polytope $\mathcal{P}(A, b)$, i.e. $A_i x \leq b_i$ will be integrated as it is (with (A_i, b_i) the i^{th} columns of (A, b)). This is especially helpful for the navigation problem within sequential polytopes (e.g. using mix-integer programming MPC methods [5]) where several edges are not the boundary of the working zone but just separators between different polytopes. However, the problem with multiple polytopes is not within the scope of this paper as we focus on introducing the tightening polytopic constraint (8) and its use for the navigation problem within one single polytope.

Remark 1. *The tightening form $Ap \leq b - d\mathbf{1}$ as in (8) does not imply any more conservativeness than its original form $Ap \leq b$ since the inward offset value d can equal to zero. This also maintains the complexity level w.r.t. $Ap \leq b$, i.e. linear constraints as in (8). Moreover, it allows to encourage the position p to stay away from the edges by: i) maximizing the offset value d so that the robot will stay as far as possible from the edges of the polytope $\mathcal{P}(A, b)$ (4) or ii) by maintaining a desired offset value d_r so that the robot will attempt to move*

inside the shrunk polytope $\mathcal{P}(A, b - d_r\mathbf{1})$. Note that both the approaches do not enforce any specific offset values but only encourage the inward offset by using the cost function of the corresponding MPC controllers. The details will be presented in the forthcoming section. □

B. Normalization of the half-space representation

Let us consider an arbitrarily chosen polytope $\mathcal{P}(A, b)$ and the corresponding offset polytope $\mathcal{P}(A, b - d\mathbf{1})$ with an applicable offset value d as in (8). Then, the distances between all the original edges of $\mathcal{P}(A, b)$ and their corresponding inwardly offset edges of $\mathcal{P}(A, b - d\mathbf{1})$ do not equal to the chosen value d in general. This causes difficulties when using the tightening constraint proposed in (8) since the actual effects of the inward offset are not as intended by the value d and are different for each of the edges. We propose here the so-called, *normalization* operation to ensure that the same offset distance d will be applied to all the edges of the polytope $\mathcal{P}(A, b)$.

Definition 1 (Half-space normalization)**.** *Let us consider a polytope $\mathcal{P}(F, g) \subset \mathbb{R}^n$ in which, $F \in \mathbb{R}^{m\times n}$ and $g \in \mathbb{R}^m$ are indexed by:*

$$F = \begin{bmatrix} F_1 \\ \dots \\ F_m \end{bmatrix} = \begin{bmatrix} f_{11} & f_{12} & \dots & f_{1n} \\ \dots & \dots & \dots & \dots \\ f_{m1} & f_{m2} & \dots & f_{mn} \end{bmatrix}, \tag{10a}$$

$$g = \begin{bmatrix} g_1, & \dots, & g_m \end{bmatrix}^\top, \tag{10b}$$

with all f_{ij} and g_i scalar numbers. The normalization operation provides a new half-space representation of the same polytope $\mathcal{P}(F, g)$ defined as:

$$(\bar{F}, \bar{g}) = \overline{(F, g)} \tag{11}$$

with $\bar{F} \in \mathbb{R}^{m\times n}$ and $\bar{g} \in \mathbb{R}^m$ calculated as follows:

$$\bar{F} = \begin{bmatrix} f_{11}/\|F_1\| & f_{12}/\|F_1\| & \dots & f_{1n}/\|F_1\| \\ \dots & \dots & \dots & \dots \\ f_{m1}/\|F_m\| & f_{m2}/\|F_m\| & \dots & f_{mn}/\|F_m\| \end{bmatrix}, \tag{12}$$

$$\bar{g} = \begin{bmatrix} g_1/\|F_1\|, & \dots, & g_m/\|F_m\| \end{bmatrix}^\top, \tag{13}$$

with $\|F_i\|$ the Euclidean norm (2-norm) of the array F_i as defined in (10a). Note that, the polytope $\mathcal{P}(\bar{F}, \bar{g})$ with $(\bar{F}, \bar{g})$ resulted from the normalization (11) is the same as the original polytope $\mathcal{P}(F, g)$:

$$\mathcal{P}(\bar{F}, \bar{g}) \equiv \mathcal{P}(F, g). \tag{14}$$

In the following, we show that the normalized half-space representation ensures that the inward offset distance d as defined in (8) is applied for all of the edges.

Proposition 1. *Let us consider the normalized half-space representation of the polytope $\mathcal{P}(A, b)$ as in (4):*

$$(\bar{A}, \bar{b}) = \overline{(A, b)}, \tag{15}$$

with the normalization operation defined in (11). Then, the offset polytope $\mathcal{P}(\bar{A}, \bar{b} - d\mathbf{1})$ (i.e. with an applicable value of d from (8)) has all of its edges being inwardly translated for a distance d with respect to the corresponding edges of the original polytope $\mathcal{P}(A, b)$. □

Proof. At first, the normalization (11) implies that $\mathcal{P}(\bar{A}, \bar{b}) \equiv \mathcal{P}(A, b)$ as in (14). Next, let us consider the two i^{th} edges of the polytopes $\mathcal{P}(\bar{A}, \bar{b})$ and $\mathcal{P}(\bar{A}, \bar{b}-d\mathbf{1})$ (for all $i \in \{1, \dots, m\}$ from (4)), which are denoted as $\bar{A}_i p \le \bar{b}_i$ and $\bar{A}_i p \le \bar{b}_i - d$, respectively (with $\bar{A}_i$ the i^{th} row of the matrix $\bar{A}$, $\bar{b}_i$ the i^{th} element of the vector $\bar{b}$ and $p \in \mathbb{R}^2$ the position). Then, the distance between them is:

$$\frac{\bar{b}_i - (\bar{b}_i - d)}{\|\bar{A}_i\|} = \frac{d}{1} = d, \tag{16}$$

in which, the unit norm property $\|\bar{A}_i\| = 1$ results from the normalization operation (10a). This completes the proof. □

C. MPC designs with tightening constraints

We employ the discrete MPC formulation as in [6], [11] to navigate the DWR system to the desired goal p_e (7). Therefore, the first step is to discretize the continuous DWR model (1):

$$x_{i+1} = f_d(x_i, u_i), \tag{17}$$

in which, (x_i, u_i) gives the values of the state x and the input u at time step i and f_d can be obtained using various discretization methods in the literature, e.g. exact discretization [6], Euler method [9], Runge-Kutta method [11]. Note that, the discretization requires an appropriate sampling time which is specific for each individual DWR system. In the following, we provide an MPC framework for the system (17) to navigate to the desired position p_e using the proposed tightening polytopic constraints (8). The controller is formulated at a general step k with the current state feedback $x(k)$:

$$\{u_0^*, \dots\} = \arg\min \sum_{i=0}^{N_p} (\|\hat{p}_i - p_e\|_Q^2 + \|\hat{u}_i\|_R^2) + \sum_{i=0}^{N_s} \ell(d_i),$$

subject to:

$$\hat{x}_{i+1} = f_d(\hat{x}_i, \hat{u}_i), \ \forall i \in [0, N_p - 1], \tag{18a}$$

$$u_{\min} \le \hat{u}_i \le u_{\max}, \ \forall i \in [0, N_p - 1], \tag{18b}$$

$$\hat{p}_i = M\hat{x}_i, \ \forall i \in [0, N_p], \tag{18c}$$

$$\bar{A}\hat{p}_i \le \bar{b} - d_i\mathbf{1}, \ \forall i \in [0, N_p], \tag{18d}$$

$$0 \le d_i \le \bar{d}_{\max}, \ \forall i \in [0, N_p], \tag{18e}$$

$$\hat{x}_0 = x(k), \tag{18f}$$

with $(\hat{x}_i, \hat{u}_i)$ the predicted state and input at the prediction step i, $f_d(\cdot)$ the discrete system (17), $(u_{\min}, u_{\max})$ the input limits (5), M as in (3). The pair of $(\bar{A}, \bar{b})$ is obtained from the normalization (15) and d_i is the offset value at the prediction step i. The maximum offset value $\bar{d}_{\max}$ is chosen as similar to (9) but in terms of the pair $(\bar{A}, \bar{b})$:

$$\bar{A}x_e \le \bar{b} - \bar{d}_{\max}\mathbf{1}. \tag{19}$$

$N_p \in \mathbb{N}^*$ is the prediction horizon and $N_s \in \mathbb{N}^*$ is the shrinking horizon, $N_s \le N_p$ (c.f. Remark 2). The additional term $\ell(d_i)$ will be defined according to different purposes (e.g. maximizing or maintaining a desired distance to the edges)

hereinafter.
Then, the MPC input at step k applied to the system (17) is:

$$u_{\text{MPC}}(k) = u_0^*. \tag{20}$$

Remark 2. *The shrinking prediction horizon N_s as used in (18) appears to be an important parameter in our analysis. Within a prediction horizon, the positions at the latter predicted steps (e.g. $\hat{p}_{N_p-2}, \hat{p}_{N_p-1}, \hat{p}_{N_p}$) will be closer to the desired position p_e than those at several beginning steps (e.g. $\hat{p}_0, \hat{p}_1, \hat{p}_2$) due to the effect of minimizing $\|\hat{p}_i - p_e\|_Q^2$. This implies that the offset values at the latter predicted steps have higher chances to reach their maximum values d_{max}. However, due to the application of the first optimal input u_0^* to the real system as in (20) (i.e. MPC strategy [11]), the offset value d_1 is the most important element as it will be the actual offset value obtained at the next step with the MPC input (20) assuming no model mismatches and nominal application. Therefore, appropriate values of the shrinking horizon N_s are 3 or 4 in our analysis.* □

1) Maximizing the offset value: in order to maximize the distances from the robot to any edges of the polytope $\mathcal{P}(A, b) \equiv \mathcal{P}(\bar{A}, \bar{b})$ (15), we maximize the offset value d_i as in (18d) at all the steps i ($i \in [0, N_s]$ with N_s the shrinking horizon). In this case, the term $\ell(d_i)$ is given in a linear form:

$$\ell(d_i) = -Kd_i, \tag{21}$$

with $K > 0$ a positive coefficient.

2) Maintaining a desired offset value: Maintaining a desired offset value d_r can be done by modifying the term $\ell(d_i)$ as a standard quadratic function:

$$\ell(d_i) = K(d_i - d_r)^2, \tag{22}$$

with $K > 0$ a positive coefficient and d_r the desired offset value chosen such that $0 \leq d_r \leq \bar{d}_{\max}$ with $\bar{d}_{\max}$ as in (19).

Remark 3. *The tightening constraints (18d)–(18e) and the additional term $\ell(d_i)$ (21)–(22) in the cost function of the MPC controller (18) do not prevent the system (17) to reach the desired position p_e because the maximally offset polytope $\mathcal{P}(\bar{A}, \bar{b} - \bar{d}_{max})$ still contains the desired position p_e as required in (19). Through various simulations, these modifications even enhance several criterions such as safety (i.e. stay away from the edges of the operating zone), convergence time while do not increase much the computation time in comparison with a standard MPC design without terminal stabilizing ingredients (i.e. using a large enough prediction horizon N_p) [6]. A comprehensive study is undergoing to analyze more about the stability, feasibility and robustness of the proposed MPC designs.* □

IV. Simulation results and comparisons

We employ a simulation model of a DWR system characterized by the following numerical values:
- $u_{\min} = [0,\ -0.5]^\top$, $u_{\max} = [0.26,\ 0.5]^\top$.
- The working zone is limited by $0 \leq \xi \leq 1$ and $0 \leq \eta \leq 1$.

Therefore, the normalized half-space representation of the polytope $\mathcal{P}(\bar{A}, \bar{b})$ as in (15) is given with:

$$\bar{A} = \begin{bmatrix} 1 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix},\ \bar{B} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}. \tag{23}$$

The robot will navigate from its initial state $x_0 = (0.1,\ 0.1,\ \pi)^\top$ to the desired position $p_e = (0.6,\ 0.8)^\top$. Using the numerical values of $(p_e, \bar{A}, \bar{b})$, we obtain the maximum offset value $\bar{d}_{\max} = 0.2$ as in (19). Regarding the discretization model in (17), we employ the standard Euler discrete model [9] with the sampling time of 0.1 seconds. In the followings, we will present the comparison between the proposed MPC designs and different methods (i.e. standard MPC and potential field approach). For the comparison, we consider four different controllers:
- *Scenario 1*: A standard MPC controller is employed. I.e. the term $\ell(\cdot)$ as in (18) and the maximum offset value $\bar{d}_{\max}$ as in (18e) are set as zeros.
- *Scenario 2*: The proposed MPC controller (18) with the option on maximizing offset (21) is employed.
- *Scenario 3*: The proposed MPC controller (18) with the option on maintaining desired offset (22) is employed with $d_r = 0.1$.
- *Scenario 4*: An MPC with repulsive potential fields (PF) [16]. I.e. the maximum offset value $\bar{d}_{\max}$ as in (18e) is set as zero and the term $\sum \ell(\cdot)$ used in (18) is replaced with:

$$K \sum_{i=0}^{N_p} \left(\sum_{j=1}^{m} \frac{d_{\text{effect}}^2}{(\bar{A}_j \hat{p}_j - \bar{b}_j)^2 + d_{\text{effect}}^2} \right), \tag{24}$$

in which, $K > 0$ is the coefficient, $d_r > 0$ stands for the effective range of the potential fields and $(\bar{A}_j, \bar{b}_j)$ are the j^{th} columns of $(\bar{A}, \bar{b})$ as given in (23) ($j \in [1, \ldots, m]$). We will split this scenario into Scenario 4.1 and Scenario 4.2 corresponding to two different values of K, i.e. 0.1 and 0.8 since these two tuning values provides similar trajectories (regarding distances from the robot to the boundary) in comparison with those obtained from our proposed MPC designs. The parameters of the four controllers are summarized in Table I. The MPC controllers are implemented using the nonlinear solver IPOPT [15] in Python 3.

In Figures 1–4, we present the simulation results under five scenarios (i.e. having Scenario 4.1 and 4.2) plotted in red, green, blue, yellow and magenta, respectively. Five 2D trajectories $(\xi,\ \eta)$ are plotted in Figure 1 which all start from the initial position p_0 (represented by the black circle with an arrow pointing towards the initial direction). However, only the first four controllers succeed in driving the robot to the desired position p_e while the last MPC controller with stronger PF ($K = 0.8$) gets stuck in its local minima and cannot reach the goal. Furthermore, it can be observed that the trajectory resulted from the standard MPC controller under Scenario 1 (red line) stays closest to the boundary, wheres those obtained from the proposed MPC with maximal offset (21) (under Sce.

2, green line) and the MPC with strong PF (under Sce. 4.2, dashed magenta line) are furthest from the boundary. We can also see the similar trajectories obtained from the proposed MPC with desired offset $d_r = 0.1$ (under Sce. 3, blue line) and the MPC with weak PF ($K = 0.1$ as in (24), under Sce. 4.1, yellow line) which intuitively implies that the two controllers share the same effects on pushing the robot far away from the boundary. Same argument can be applied for the proposed MPC with maximal offset (green line) and the MPC with strong PF ($K = 0.8$ as in (24), under Sce. 4.2). They prove the effectiveness of the tightening polytopic constraint (8) and its use in the two proposed MPC designs (18)–(22). Furthermore, the MPC designs with tightening constraint (18)–(22) do not prevent the stability of the closed-loop system as long as the maximal offset $\bar{d}_{\max}$ satisfies the condition (19) which is a significant advantage over the PF approach under which, the system probably get stuck at the local minima as shown under Scenario 4.2, with $K = 0.8$.

TABLE I: Control parameters for all scenarios.

	Parameters	Values
Common paramters	N_p as in (18)	30
	Q as in (18)	$\boldsymbol{I}_2$
	R as in (18)	$0.01\boldsymbol{I}_2$
Scenario 1	$\ell(\cdot)$ as in (18)	0
	$d_{\max}$ as in (18e)	0
Scenario 2	N_s as in (18)	3
	$\ell(\cdot)$ as in (18)	as in (21)
	K as in (21)	100
	$d_{\max}$ as in (18e)	0.2
Scenario 3	N_s as in (18)	3
	$\ell(\cdot)$ as in (18)	as in (22)
	K as in (22)	100
	d_r as in (22)	0.1
	$d_{\max}$ as in (18e)	0.2
Scenario 4.1	K as in (24)	0.1
	d_{effect} as in (24)	0.2
Scenario 4.2	K as in (24)	0.8
	d_{effect} as in (24)	0.2

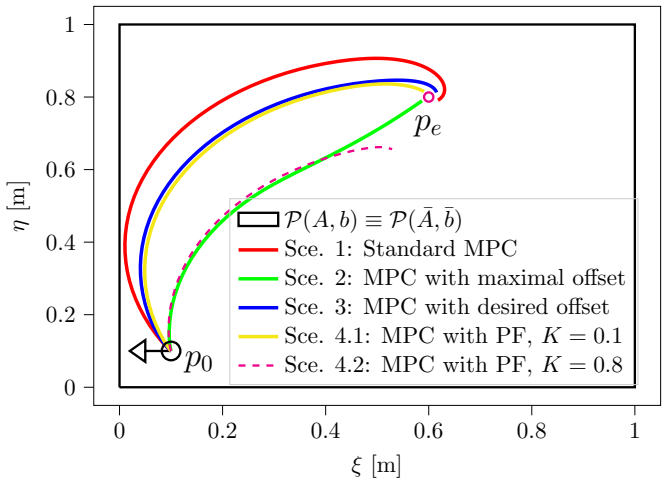

Fig. 1: Simulation trajectories under different scenarios.

The convergences of the state ξ and the corresponding input values under five scenarios are shown in Figures 2–4 in which, we can observe the same responses of all the simulated controllers at the beginning phase, i.e. rotating on spot in counter-clockwise representing by the negative values of the angular velocity in Figure 4, the zero linear velocity in Figure 3 and the constant $\xi = \xi_0 = 0.1$ in Figure 2 within twenty first steps. These responses are reasonable since the initial orientation is π (black arrow pointing to the left in Figure 1) and the robot cannot move backward, $u_{\min} = (0, -0.5)^\top$ as used in 5. The robot starts to move at different moments under different scenarios: the standard MPC (red lines, Sce. 1) drives the robot earliest (around step 18 as can be seen in Figure 3) when the direction is still heading much to the left) which leads to the closest trajectory to the boundary (red line in Figure 1) as being mentioned before. In contrast, the MPC with maximal offset (green lines, Sce. 2) and the MPC with strong PF (dashed magenta line, Sce. 5) keep rotating on spot and only move forward after step 27 (Figure 3) when they already obtain clearer heading angles and hence, the resulted trajectories are the furthest from the boundary as can be seen in Figures 1–2. Furthermore, regarding Sce. 4.2 in which, the robot gets stuck at the local minima of the strong PF ($K = 0.8$), it can be seen in Figure 3 that the linear velocity reduces to zero at step 72 being the earliest to stop (dashed magenta line).

For the two proposed MPC designs (18)–(22), we also report the offset values of d_0 as used in (18d) at every simulation step (the offset distance achieved with the actual state) in Figure 5 in which, the values resulted from the MPC with maximal offset (green line, Sce. 2) reaches the maximal offset value $d_{\max} = 0.2$ while those resulted from the MPC with desired offset (blue line, Sce. 3) stay at $d_r = 0.1$ after a reduction happening from step 20 to step 40. In general, it can be concluded that our proposed MPC designs (18)–(22) work as intended, i.e. the MPC with maximal offset (21) and the MPC with desired offset (22) all drive the system to the desired position while staying as far as possible from the boundary (Sce. 2, green lines in all figures) and allowing a desired distance from the boundary (Sce. 3, blue lines in all figures), respectively.

TABLE II: Computing time of five MPC controllers
Unit: millisecond. Std: standard deviation.

Name	Mean	Std	Min	Max
Standard MPC	66.76	8.20	57.88	86.71
MPC with maximal offset	68.49	4.60	61.83	80.73
MPC with desired offset	72.33	6.18	61.52	90.63
MPC with PF ($K = 0.1$)	82.85	4.43	74.82	99.78
MPC with PF ($K = 0.8$)	87.90	3.58	78.77	103.86

Finally, we report the computing time per step for all five scenarios in Table II (data collected within the simulation horizon of 200 steps) in which, the mean values increase from Sce. 1 to Sce. 5. The increasing order of computation burden among five controllers is:
Standard MPC $<$ MPC with maximal offset (i.e. linear additional term as in (21)) $<$ MPC with desired offset (i.e. quaratic

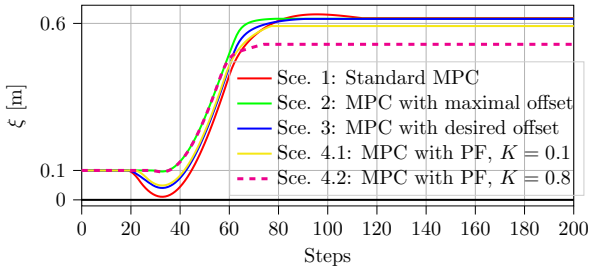

Fig. 2: Convergences of state ξ under different scenarios.

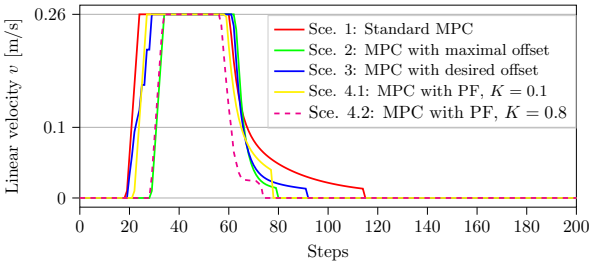

Fig. 3: Input v under different scenarios.

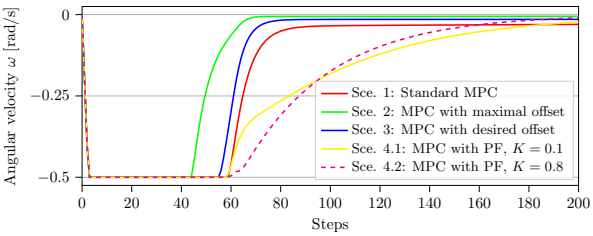

Fig. 4: Input w under different scenarios.

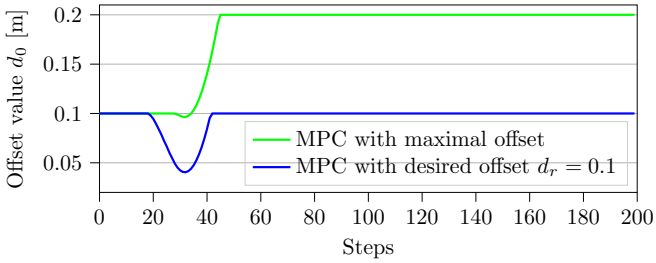

Fig. 5: Values of offset distance under different scenarios.

additional term as in (22)) < MPC with medium PF ($K = 0.1$ as in (24)) < MPC with strong PF ($K = 0.8$ as in (24)). Hence, this again confirms the advantages of our proposed MPC designs with tightening constraints (18)–(22) over the potential field approach, i.e. providing better control performances (e.g. convergence without stucking in local minima, pushing away from boundary) with less computation burden.

V. Conclusions

This paper presented the formulation of the "so-called" tightening polytopic constraint and its uses in MPC designs for DWR system to navigate to the goal while being pushed away from the boundary of the polytopic working zone. The tightening polytopic constraint introduces one more scalar variable representing an inward offset to the original polytope while maintaining the linear form. Then, the offset value will be encouraged to have a large value, or a desired value (according to interest) by adding a linear, or quadratic terms to the standard cost function in the MPC designs. Through extensive simulations, the proposed methods with tightening polytopic constraint succeed to drive the DWR system to the goal and furthermore, report significant advantages over the repulsive potential field approach (convergence without being stuck at local minima and less computation time).

Future work will analyze more about the effects of the tightening constraint and of the parameters (such as the desired offset value or the shrinking horizon N_s as in (18)) on the stability, feasibility and robustness of the MPC controllers.

References

[1] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Practical search techniques in path planning for autonomous driving," *Ann Arbor*, vol. 1001, no. 48105, pp. 18–80, 2008.

[2] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *The international journal of robotics research*, vol. 30, no. 7, pp. 846–894, 2011.

[3] N. T. Nguyen, L. Schilling, M. Angern, H. Hamann, F. Ernst, and G. Schildbach, "B-spline path planner for safe navigation of mobile robots," *accepted for publication at 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2021)*.

[4] N. T. Nguyen, I. Prodan, and L. Lefèvre, "Flat trajectory design and tracking with saturation guarantees: a nano-drone application," *International Journal of Control*, vol. 93, no. 6, pp. 1266–1279, 2020.

[5] D. Ioan, I. Prodan, S. Olaru, F. Stoican, and S.-I. Niculescu, "Mixed-integer programming in motion planning," *Annual Reviews in Control*, 2020.

[6] K. Worthmann, M. W. Mehrez, M. Zanon, G. K. Mann, R. G. Gosine, and M. Diehl, "Model predictive control of nonholonomic mobile robots without stabilizing constraints and costs," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 4, pp. 1394–1406, 2015.

[7] J. J. Park, C. Johnson, and B. Kuipers, "Robot navigation with model predictive equilibrium point control," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 4945–4952.

[8] A. De Luca, G. Oriolo, and M. Vendittelli, "Control of wheeled mobile robots: An experimental overview," *Ramsete*, pp. 181–226, 2001.

[9] F. Künhe, J. Gomes, and W. Fetter, "Mobile robot trajectory tracking using model predictive control," in *II IEEE latin-american robotics symposium*, vol. 51. Citeseer, 2005.

[10] R. B. Patel and P. J. Goulart, "Trajectory generation for aircraft avoidance maneuvers using online optimization," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 1, pp. 218–230, 2011.

[11] N. T. Nguyen, I. Prodan, and L. Lefèvre, "Stability guarantees for translational thrust-propelled vehicles dynamics through nmpc designs," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 1, pp. 207–219, 2020.

[12] A. Liniger, A. Domahidi, and M. Morari, "Optimization based autonomous racing of 1:43 scale rc cars," *Optimal Control Applications and Methods*, vol. 36, no. 5, pp. 628–647, 2013.

[13] X. Zhang, A. Liniger, A. Sakai, and F. Borrelli, "Autonomous parking using optimization-based collision avoidance," in *57th IEEE Conference on Decision and Control*, 2018, pp. 4327–4332.

[14] C. Liu and M. Tomizuka, "Real time trajectory optimization for nonlinear robotic systems: Relaxation and convexification," *Systems & Control Letters*, vol. 108, pp. 56–63, 2017.

[15] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

[16] N. Q. H. Tran, I. Prodan, and L. Lefevre, "Nonlinear optimization for multi-agent motion planning in a multi-obstacle environment," in *2017 IEEE International Conference on System Theory, Control and Computing (ICSTCC)*, 2017, pp. 488–493.