Smooth approximation of polyhedral potential field in NMPC for obstacle avoidance

Theodor-Gabriel Nicu, Florin Stoican, Ionela Prodan

Smooth approximation of polyhedral potential field in NMPC for obstacle avoidance

Preprint · February 2023

CITATIONS
0

READS
118

3 authors:

Theodor-Gabriel Nicu
Universitatea Națională de Știință și Tehnologie Politehnica București
8 PUBLICATIONS **14** CITATIONS
SEE PROFILE

Florin Stoican
Universitatea Națională de Știință și Tehnologie Politehnica București
145 PUBLICATIONS **1,772** CITATIONS
SEE PROFILE

Ionela Prodan
Supélec
111 PUBLICATIONS **1,125** CITATIONS
SEE PROFILE

Smooth approximation of polyhedral potential field in NMPC for obstacle avoidance

Theodor-Gabriel Nicu[1], Florin Stoican[1], Ionela Prodan[2]

***Abstract*— The sum function notion allows to define a piecewise affine (over a polyhedral support) surface which accurately penalizes the closeness to polyhedral obstacles. This, in turn, leads to a piecewise description of the potential field surface further used in a NMPC (Nonlinear Model Predictive Control) motion planning problem.**

We introduce and analyze here their smooth equivalents to show significant computational speedup. We analyze the links between the piecewise and smooth surfaces (magnitude and location of critical points). The results are validated in simulation and shown to compare favourably with previous mixed-integer based formulations.

I. Introduction

The last decades have seen an exponential increase in the analysis and design of motion planning in both academia and industry. The people's need and interaction with mobile platforms is constantly increasing, and as such, a wide variety of applications have been (and still are) developed (see, e.g., [1]). One of the main issues in the field is the obstacle avoidance problem under the restriction of constraints imposed by the agent's internal dynamics and computational limitations [2].

One popular approach is based on *artificial potential field methods* (see, e.g., [3]). The idea is to "push" the complex part into the offline (assuming the environment known) design of a surface (the potential field) which incorporates information about the environment and then, at runtime, provide a relatively simple control action (e.g., take the surface's gradient as control action). Hence, the design stage should provide a surface with "hills" that repel from the obstacles's location and "valleys" to direct towards the goal.

Arguably, the two main problems affecting potential field methods are:

i) composing a surface which accounts for all elements of interest;
ii) finding a control action which does not get bogged down in the inevitable local minima.

For the first, we make use and extend the concept of the "sum function", employed in [4], [5], to accurately describe polyhedral obstacles. For the second, we consider a model predictive control (MPC) implementation to compute the control action. Without being a panacea, MPC has the double benefit of taking into account constraints and of predicted the system's evolution along a prediction horizon (which largely avoids the pitfalls of greedy methods such as the steepest gradient selection). Note as well that the local minima issue is well known in the state of the art with multiple methods employed with varying degrees of success, e.g., *the virtual hill concept* [6], local minima avoidance using an *input to state stability property* [7].

When using MPC one should be careful not to negate the advantages which led to selecting a potential field method in the first place. Specifically, MPC is usually implemented as an online constrained optimization problem which requires solving at each simulation step. Especially when the cost is nonlinear (the potential field in our case), the computation time may become significant. E.g., the piecewise affine descriptions given in [5], [4] require mixed integer modelling and subsequent use of specialized solvers. This shortcoming is the main justification for the current paper. We aim to use the shape-description capabilities of the sum function but also to avoid the computational overload associated with piecewise descriptions. The paper's contributions are:

i) We relax the discontinuity-inducing terms by smooth near-equivalents which are well known in the literature. The resulting surfaces (the sum function and potential field) gain smoothness at the price of a reasonable loss in the precision of representation;
ii) This in turn allows to express the MPC cost as a smooth, albeit strongly nonlinear, term. Due to the cost gradient's continuity, a significant computational speedup is observed;
iii) Lastly, we analyze the links between the piecewise surfaces and their smooth counterparts from the viewpoints of approximation error and location of local minima.

The rest of the paper is organized as follows. Section II revisits the 'sum function' definition and its use in constructing a potential field. Section III introduces its smoothing approximation and IV enumerates some of the links between the piecewise and smoothed cases. Simulation results are shown in Section V and the conclusions are drawn in Section VI.

[1] Politehnica University of Bucharest, Department of Automatic Control and Systems Engineering, Romania {theodor.nicu,florin.stoican}@upb.ro

[2] Univ. Grenoble Alpes, Grenoble INP†, LCIS, F-26000, Valence, France, ionela.prodan@lcis.grenoble-inp.fr. † Institute of Engineering and Management Univ. Grenoble Alpes.

II. Problem statement

Arguably, the two main problems affecting potential field methods are, first, composing a surface which accounts for all elements of interest and second, finding a control action which does not get bogged down in the inevitable local minima of the afore-obtained surface.

A. The overall sum function surface

We start by recapitulating the 'sum function' idea [8] which takes into account the shape of polyhedral obstacles to provide a piecewise affine surface, further employed in the design of the repulsive component of the potential field.

As prerequisite, let us define the pairs $(a_k^\top, b_k) \in \mathbb{R}^n \times \mathbb{R}$ for $k = 1 \dots N$ which describe hyperplanes,

$$H_k^0 = \{x : \mathbb{R}^n : \ a_k^\top x = b_k\}, \tag{1}$$

cutting the space $\mathbb{R}^n$ into disjoint (up to their border) half-spaces

$$H_k^\pm = \{x : \mathbb{R}^n : \ \mp a_k^\top x \leq \mp b_k\}. \tag{2}$$

Taking all these cuts gives a hyperplane arrangement [9], a union of disjoint cells,

$$\mathcal{A}(\sigma) = \bigcap_{k=1}^{N} H_k^{\sigma_k}, \tag{3}$$

each of them defined by a unique combination of signs $\sigma = \{\sigma_1 \dots \sigma_N\} \in \Sigma \subset \{\pm\}^N$. This notation provides a coherent framework in which to define obstacles

$$\mathbb{P} = \bigcup_{\sigma^{\bullet,i} \in \Sigma^\bullet} \mathcal{A}(\sigma^{\bullet,i}), \quad \mathbb{R}^n \setminus \mathbb{P} = \bigcup_{\sigma^{\circ,j} \in \Sigma \setminus \Sigma^\bullet} \mathcal{A}(\sigma^{\circ,j}), \tag{4}$$

by a list of *interdicted* sign tuples $\Sigma^\bullet \subset \Sigma$.

We may now introduce $\gamma^{\bullet,i}(x)$, the sum function associated to the i-th obstacle (defined by $\mathcal{A}(\sigma^{\bullet,i})$):

$$\gamma^{\bullet,i}(x) = \sum_{k=1}^{N} \omega_k \Big[\sigma_k^{\bullet,i} \left(a_k^\top x - b_k\right) + \left|\sigma_k^{\bullet,i} \left(a_k^\top x - b_k\right)\right| \Big], \tag{5}$$

with $\omega_k \geq 0$ a positive scalar weight[1] and $\sigma_k^{\bullet,i}$, the k-th element of the $\sigma^{\bullet,i}$ sign tuple. Since the sign of the absolute operator appearing in (5) changes only when x switches between half-spaces (2), the rationale for introducing (1)–(4) becomes clear: the hyperplane arrangement induced by (1) serves as polyhedral support for (5) and allows to define it piecewise affine as

$$\gamma^{\bullet,i}(x) = 2 \sum_{k:\ \sigma_k^{\bullet,i} \neq \sigma_k} \sigma_k^{\bullet,i} \left(a_k^\top x - b_k\right), \tag{6}$$

for all $x \in \mathcal{A}(\sigma)$ with $\sigma \in \Sigma$.

The piecewise affine relation (6) is defined for a single obstacle. The extension to the multi-obstacle case, as per [4], is done by applying the 'min' operator:

$$\gamma^\bullet(x) = \min_{\sigma^{\bullet,i} \in \Sigma^\bullet} \gamma^{\bullet,i}(x). \tag{7}$$

[1]Since ω_k can be 'pushed' into $(a_k^\top, b_k)$ by a suitable scaling, we ignore it hereafter.

The result is that (7) will be zero whenever x is inside any of the obstacles defined by $\sigma^{\bullet,i} \in \Sigma^\bullet$ and greater than zero otherwise [4], [5]. This makes (7) into a 'seed' to which standard repulsive functions are applied to construct a repulsive potential surface taking into account the polyhedral shape of the obstacles. This is in contrast with the standard approach where only the distance to the obstacle's center is considered (and hence information about its shape is lost/disregarded).

B. Motion planning problem

Consider the linear time invariant dynamics

$$x_{k+1} = Ax_k + Bu_k, \quad y_k = Cx_k, \tag{8}$$

with $u \in \mathbb{R}^m$, $x \in \mathbb{R}^n$, $y \in \mathbb{R}^p$ the system's input, state and output vectors. Matrices A, B, C define the state-space representation and are of appropriate dimensions.

The goal of a typical motion problem is to give a sequence of inputs such that a cost minimized (e.g., total path length) and various constraints are respected (e.g. obstacle avoidance and target reaching). The ethos of potential field formulations is to relax hard constraints into penalties added to the cost, i.e., to construct a potential field surface which repels the agent from obstacles and attracts it toward the goal.

The danger is that a simple update rule like taking the steepest descent gradient along the potential surface is too "greedy" and risks to get bogged into local minima. Hence, a popular approach is to apply a receding horizon technique, such that the input action is taken by analyzing the behavior of the system along a prediction horizon (and not looking only at the current time instant). A typical formulation [10] is given by the model predictive control (MPC) approach

$$\bar{\mathbf{u}}_N^\star = \arg\min_{\bar{\mathbf{u}}_N} \sum_{\ell=1}^{N} P^A(\bar{x}_\ell - r) + P^R\left(\gamma^\bullet(\bar{y}_\ell)\right), \tag{9a}$$

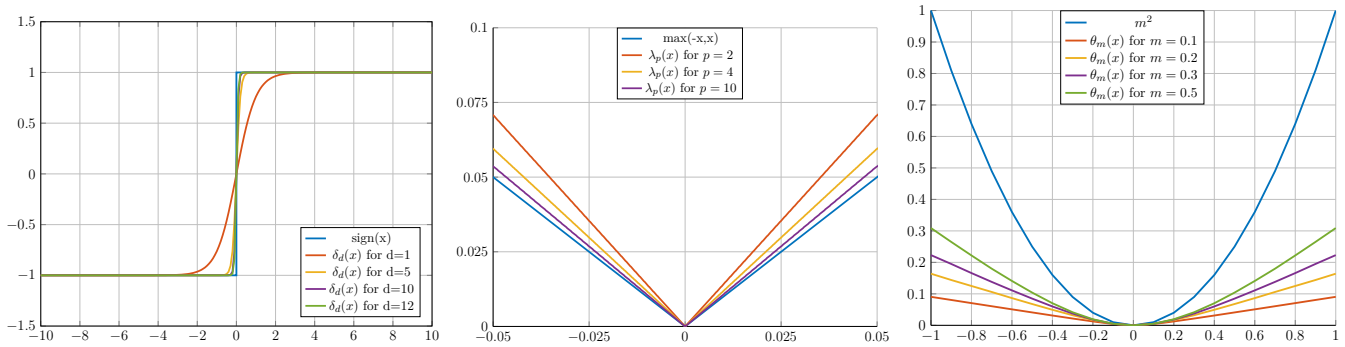
$$\text{s.t. } \bar{x}_{\ell+1} = A\bar{x}_\ell + B\bar{u}_\ell, \ \bar{y}_\ell = C\bar{x}_\ell, \tag{9b}$$

$$\bar{u}_\ell \in \mathcal{U}, \bar{y}_\ell \in \mathcal{Y}, \ \forall \ell = 0 \dots N-1. \tag{9c}$$

The cost penalizes the distance to a target r via the attractive potential $P^A(\bar{x}_\ell - r)$ and the closeness to obstacles, via the repulsive potential $P^R(\gamma^\bullet(\bar{y}_\ell))$. $P^A(\cdot) : \mathbb{R} \mapsto [0, \infty)$ and $P^R(\cdot) : \mathbb{R} \mapsto [0, \infty)$ are the attractive, and respectively repulsive functions and may be of any form as long as they are monotonously increasing, and respectively decreasing. In $\mathcal{U}, \mathcal{Y}$ are kept the 'simple' constraints such as those describing input/output magnitude bounds. The loop is closed by choosing as starting predicted state the current state of the dynamics ($\bar{x}_0 \leftarrow\!\shortmid x_k$) and by applying the first input from the optimal sequence $\bar{\mathbf{u}}_N^\star$ to the dynamics ($\bar{u}_0^\star \mapsto u_k$).

Several remarks are in order.

Remark 1. *Note the use of y_ℓ as argument in the repulsive potential component. In motion planning, usually the output measures the agent's position and the obstacles are in the same subspace.* ■



(a) smooth signum (b) softmax (c) pseudo Huber loss

Fig. 1: Smooth approximations of discontinuous functions

Remark 2. *Exploiting the monotonicity assumption for the repulsive function $P^R(\cdot)$, we have the equivalence*

$$P^R\left(\gamma^\bullet(x)\right) = P^R\left(\min_{\sigma^{\bullet,i}\in\Sigma^\bullet} \gamma^{\bullet,i}(x)\right), \tag{10a}$$

$$= \max_{\sigma^{\bullet,i}\in\Sigma^\bullet} P^R\left(\gamma^{\bullet,i}(x)\right). \tag{10b}$$

■

Remark 3. *Lastly, while MPC is by no means a panacea, in practice it still shows good performance (local minima avoidance, target reaching) for reasonable computational complexity. This is still dependent on length of the prediction horizon and on adequate tuning of the various parameters appearing in the cost.* ■

III. Main idea

The sum function (7) is an interesting idea since it allows to consider the obstacles' shape. Furthermore, its complexity depends not on the number of obstacles (as given by the size of sign tuple collection $\Sigma^\bullet$) but on the number of hyperplanes (1) defining the underlying hyperplane arrangement. Still, the piecewise description shown in (6), which becomes even more convoluted by applying the 'min' operator in (7), see [5], leads to mixed-integer formulations which are difficult to model and solve, despite powerful dedicated solvers and recent advances [11].

As previously noted, the piecewise nature of (7) impedes a fast resolution of the repeated receding horizon (9). The solution we propose hereinafter is to relax the discontinuity-inducing terms by smooth near-equivalents which are well known in the literature. The resulting surfaces (i.e., the sum function and potential field) gain smoothness at the price of a reasonable loss in the precision of representation. This in turn allows to express the cost in (9) as a smooth, albeit strongly nonlinear, term.

A. Smooth relaxations

As seen in (5) the apparition of the absolute operator induces a discontinuity coming from the switch in the sign of the argument, i.e., $|x| = \text{sign}(x)\cdot x$. This issue is well-known in the literature where various sigmoid-like functions are given as candidates to model $\text{sign}(x)$. In what follows we consider the hyperbolic tangent

$$\delta_d(x) : \mathbb{R} \mapsto [-1,1], \quad \delta_d(x) = \frac{e^{dx} - e^{-dx}}{e^{dx} + e^{-dx}}, \tag{11}$$

parameterized after a scalar $d \geq 0$. For further use, note that (11)'s derivative is given by

$$\frac{d}{dx}\delta_d(x) = d\frac{\left(e^{dx}+e^{-dx}\right)^2 - \left(e^{dx}-e^{-dx}\right)^2}{\left(e^{dx}+e^{-dx}\right)^2} = d(1-\delta_d^2(x)). \tag{12}$$

Fig. 1a illustrates the discontinuous $\text{sign}(x)$ and $\delta_d(x)$, defined as in (11), for $d \in \{1,5,10,12\}$. As expected, larger values of d correspond to a faster variation of the function $\delta_d(x)$ (hence, a better approximation of the discontinuity).

The other source of discontinuity is the max operator, applied in (10b). To smooth it we consider one of the 'softmax' variations [12]. Specifically, we exploit the fact that the sublevels induced by p-norms respect an inclusion property. That is, for any $p \geq q$ we have that $\{x \in \mathbb{R}^n : \|x\|_p = 1\} \subseteq \{x \in \mathbb{R}^n : \|x\|_q = 1\}$. Hence, taking $p \to \infty$ leads to an approximation of the ∞-norm (which norm in fact describes the max operator). To this end, we define

$$\lambda_p(x) : \mathbb{R}^n \mapsto [0,\infty), \quad \lambda_p(x) = \left(\sum_k x_k^p\right)^{\frac{1}{p}}. \tag{13}$$

We disregard the $|\cdot|$ operator in (13) by choosing p even. For further use, note that (13)'s gradient is given by

$$\nabla\lambda_p(x) = \left(\sum_k x_k^p\right)^{-\frac{p-1}{p}} \cdot x^{p-1} = \lambda_p^{1-p}(x)\cdot x^{p-1}. \tag{14}$$

The power operation applies elementwise.

Fig. 1b illustrates the behavior of $\max(-x,x)$ in both exact form and approximated via (13), for $p \in \{2,4,10\}$.

Lastly, to provide a smooth description of the attractive potential component, we introduce

$$\theta_m(x) : \mathbb{R} \mapsto [0, \infty),\ \theta_m(x) = m^2\left[\sqrt{1 + (x/m)^2} - 1\right], \tag{15}$$

the pseudo-Huber loss function [13]. Its behavior is quadratic in the vicinity of zero (it approximates $x^2/2$) and linear (with slope $\approx m$) when farther away. For further use, note that (15)'s derivative is given by

$$\frac{d}{dx}\theta_m(x) = \frac{2x}{\sqrt{1 + (x/m)^2}}. \tag{16}$$

Fig. 1c illustrates the behavior of (15) for values $m \in \{0.1, 0.2, 0.3, 0.5\}$.

B. Smooth reformulation of the problem

The notions presented at the beginning of this section will be now introduced into the piecewise affine surfaces of Section II to smooth them, i.e., to make the surface gradients continuous and endow them with a unique description (not a piecewise one).

First, let us introduce (11) in (5), to arrive at the smooth[2] form[3]

$$\gamma_\epsilon^{\bullet,i}(x) = \sum_{k=1}^{N} \sigma_k^{\bullet,i}\left(a_k^\top x - b_k\right)\left[1 + \delta_d(\cdot)\right]. \tag{17}$$

Next, recall that the overall repulsive potential is the maximum from each obstacle's repulsive potential, as shown in (10b). A first relaxation is to replace the non-smooth $\gamma^{\bullet,i}(x)$ from (5) with $\gamma_\epsilon^{\bullet,i}(x)$ from (17) in (10b):

$$\mathbb{P}^R(x) = \max_{\sigma^{\bullet,i} \in \Sigma^\bullet} P^R\left(\gamma_\epsilon^{\bullet,i}(x)\right). \tag{18}$$

Another step is required to avoid the discontinuities induced by the max operator. Making the shorthand notation $\Gamma(x) = \begin{bmatrix}\dots & P^R\left(\gamma^{\bullet,i}(x)\right) & \dots\end{bmatrix}$ for the vector of repulsive potentials induced by each obstacle, we make use of (13) to arrive at

$$\begin{aligned}\mathbb{P}_\epsilon^R(x) =& \lambda_p\left(\Gamma(x)\right) = \|\Gamma(x)\|_p = \left(\sum_i \left[P^R\left(\gamma_\epsilon^{\bullet,i}(x)\right)\right]^p\right)^{\frac{1}{p}}, \\ =& \left(\sum_i P^R\left(\sum_{k=1}^{N} \sigma_k^{\bullet,i}\left(a_k^\top x - b_k\right)\cdot\left[1 + \delta_d\left(\cdot\right)\right]\right)^p\right)^{\frac{1}{p}}.\end{aligned} \tag{19}$$

Remark 4. *Note that we have a hierarchy of successive approximations: $P^R\left(\gamma^\bullet(x)\right)$ is the exact, piecewise form, which is then relaxed (by smoothing the sum function $\gamma^{\bullet,i} \mapsto \gamma_\epsilon^{\bullet,i}$) into $\mathbb{P}^R(x)$ from (18), from which, by smoothing the* 'max'*, we arrive at $\mathbb{P}_\epsilon^R(x)$ in (19).* ■

[2]Hereinafter, we attach the 'ϵ' subscript to the smoothed variants of the exact, piececewise forms, as a stand-in for the various smoothing parameters (d, p and m).

[3]Whenever the argument is clear, and in the interest of compactness, we do not write it.

Lastly, the attractive potential is given via (15):

$$\begin{aligned}\mathbb{P}_\epsilon^A(x) &= \theta_m\left(\|x - r\|_Q\right) \\ &= m^2\left[\sqrt{1 + \left(\left(x - r\right)^\top Q\left(x - r\right)/m\right)^2} - 1\right],\end{aligned} \tag{20}$$

with r the goal to be reached and $Q \succ 0$ a positive definite weight matrix of appropriate dimensions.

To conclude, the total potential field $P(x) = \mathbb{P}_\epsilon^R(x) + \mathbb{P}_\epsilon^A(x)$ has been reformulated as a smooth (continuous in its gradient) surface which may now be integrated into the MPC problem[4] (9). While the cost is strongly nonlinear, its smoothness actually proves beneficial. Tools such as CasADi [14] which implement algorithmic differentiation provide fast and robust resolution of the problem (as will be illustrated in Section V).

IV. Relation to the piecewise affine case

We are interested in the links between the exact (given in piecewise form) and approximated surfaces discussed in the previous section.

Let us first compare $\gamma^{\bullet,i}(x)$ and $\gamma_\epsilon^{\bullet,i}(x)$. While (17) defines a smooth surface we may choose to look at it from the perspective of the polyhedral support characterizing the original sum function (as given in (6))

$$\begin{aligned}\gamma_\epsilon^{\bullet,i}(x) = &\sum_{k:\ \sigma_k^{\bullet,i} \neq \sigma_k} \sigma_k^{\bullet,i}\left(a_k^\top x - b_k\right)\left[1 + \delta_d(\cdot)\right] \\ &+ \sum_{k:\ \sigma_k^{\bullet,i} = \sigma_k} \sigma_k^{\bullet,i}\left(a_k^\top x - b_k\right)\left[1 + \delta_d(\cdot)\right],\end{aligned} \tag{21}$$

for all $x \in \mathcal{A}(\sigma)$ with $\sigma \in \Sigma$. Comparing with (6) and using that in the first sum $1 + \delta_d(x) \leq 2$ and that the elements of the second sum are negative we conclude that

$$\gamma_\epsilon^{\bullet,i}(x) \leq \gamma^{\bullet,i}(x). \tag{22}$$

Next, we assess the distance between the exact potential (10b) and its smooth approximation from (19) by recalling that for any $x \in \mathbb{R}^n$ and $p \geq 1$, inequality $\|x\|_\infty \leq \|x\|_p \leq n^{\frac{1}{p}} \cdot \|x\|_\infty$ holds. Hence, we have that

$$\begin{aligned}\max_{\sigma^{\bullet,i} \in \Sigma^\bullet} P^R(\gamma_\epsilon^{\bullet,i}(x)) \leq \mathbb{P}_\epsilon^R(x) \\ \leq n^{\frac{1}{p}} \cdot \max_{\sigma^{\bullet,i} \in \Sigma^\bullet} P^R(\gamma_\epsilon^{\bullet,i}(x)).\end{aligned} \tag{23}$$

For $p \to \infty$ the left and right bounds converge to the same value, hence $\mathbb{P}^R(x) \to \max_{\sigma^{\bullet,i} \in \Sigma^\bullet} P^R(\gamma_\epsilon^{\bullet,i}(x))$.

On the other hand, using (22), coupled with the fact that $P^R(x)$ is monotonously decreasing, we have that $P^R(\gamma_\epsilon^{\bullet,i}(x)) \geq P^R(\gamma^{\bullet,i}(x))$. Introducing this in (23) then leads to

$$\mathbb{P}^R(x) \geq \max_{\sigma^{\bullet,i} \in \Sigma^\bullet} P^R(\gamma^{\bullet,i}(x)). \tag{24}$$

[4]Note that for simplicity of notation we have used 'x' in (17)–(20). When introducing these relations in (9) we change the argument to '$\bar{y}_\ell$'.

The second element of interest is the presence and position of local minima/maxima in the potential surface. Needless to say, locating critical points in a complex, non-convex, surface is not straightforward. The idea is simple, compute the gradient and find the values where it vanishes, but the formulas quickly become unwieldy. Taking (19) and applying the chain rules of derivation, gives:

$$\nabla \mathbb{P}^R_\epsilon(x) = \lambda_p^{1-p}(\cdot) \sum_i \Big[(P^R(\cdot))^{p-1} \cdot (P^R(\cdot))' \cdot \tag{25a}$$

$$\sum_{k=1}^{N} \left[1 + \delta_n(\cdot) + n(1-\delta_n^2(\cdot))\right] \sigma_k^{\bullet,i} a_k \Big], \tag{25b}$$

where we made use of (12) to get the gradient of (17),

$$\nabla \gamma_\epsilon^{\bullet,i}(x) = \sum_{k=1}^{N} \sigma_k^{\bullet,i} \cdot \left[1 + \delta_n(\cdot) + n(1-\delta_n^2(\cdot))\right] a_k. \tag{26}$$

While an analytic description of the gradient has been written in (25), it is clearly unsuitable for further calculations. On the other hand, when far away from the points of gradient discontinuity, we may approximate $\nabla \gamma_\epsilon^{\bullet,i}(x) \approx \sum_{k=1}^{N} \sigma_k^{\bullet,i} a_k = \nabla \gamma^{\bullet,i}(x)$. A simple, constant within a cell of the polyhedral support, gradient lends itself to further manipulations. Specifically, points of local minima/maxima may be easily deduced for $\gamma^\bullet(x)$. That is, for each of the surface vertices (call it v) we take its neighboring cells (call them $\mathcal{A}(\sigma^j)$) and see whether all products $v^\top \sum_{k=1}^{N} \sigma_k^j a_k$ are positive (local maximum) or negative (local minimum). Furthermore, by construction, to each minimum/maximum of the exact overall sum function $\gamma^\bullet(x)$ corresponds a maximum/minimum of the potential field $P^R(\gamma^\bullet(x))$. In other words, we may, with some hope of success, analyze the behavior of the smooth $\mathbb{P}^r(x)$ by analyzing, the much easier to handle, $\gamma^\bullet(x)$.

Remark 5. *We have shown in this section that the sum function and repulsive potential smooth approximations are in some sense 'close' to the initial piecewise description and that, equally important, the latter may be used to infer the behavior of the former.* ■

V. Illustrative example

We consider a double integrator dynamic, discretized with a sampling time of 0.25sec, and apply the MPC controller (9) with the smoothed repulsive (19) and attractive (20) potentials over the three-obstacle (illustrated in Fig. 3) example from [5].

Applying (19) with parameters $d = 15$, $p = 6$, $m = 0.1$ and with $P^R(x) = 1/(1+x)$ leads to the surface shown in Fig. 2. The behavior is the desired one: the closeness to obstacles is penalized and their shape is clearly visible in the surface contours.

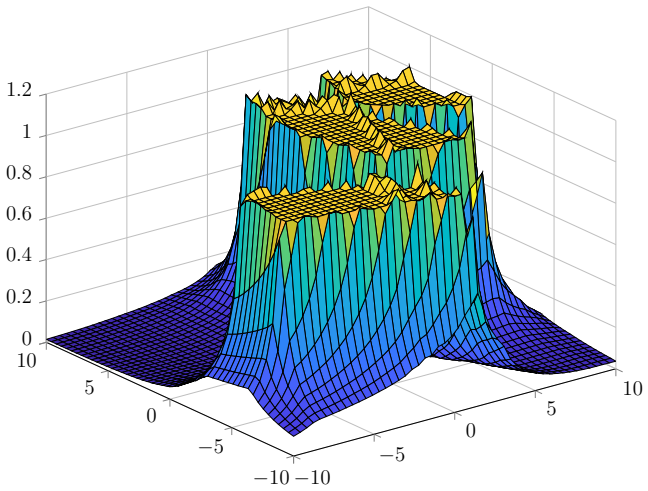


Fig. 2: The smoothed repulsive potential surface (with parameters $d = 15$, $p = 6$, $m = 0.1$)

The parameter choice is at this point rather empirical, hence we consider multiple combinations of parameters

$$S_i \in \{(d,p,m) : \{10,12\} \times \{4,6\} \times \{0.1,0.3\}\}, \tag{27}$$

to asses the scheme's behavior for various parameter values. The scenario indexing is done in lexicographical order, i.e., $S_{\{4i_d+2i_p+i_m+1\}}$ corresponds to the i_d-th 'd' value, i_p-th 'p' value and i_m-th 'm' value. In addition we consider two mixed-integer representations of the exact piecewise affine potential field from [5]: R2) describes the piecewise overall sum function by its hyperplane arrangement and R3) linearized the repulsive potential function to arrive at vertex-based piecewise affine representation.

Running (9) for the same initial point/target for the 8+2 methods we obtain the trajectories illustrated in Fig. 3. To avoid cluttering it, we only depict a subset of the trajectories (the remaining ones,

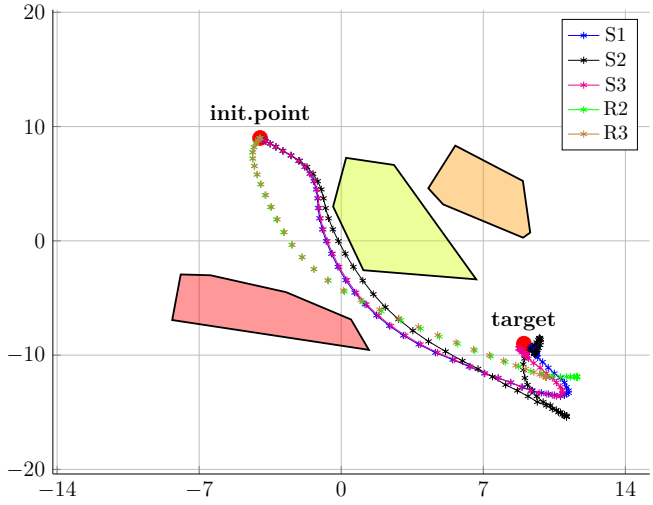


Fig. 3: Trajectories obtained for methods $\{S1, S2, S3, R2, R3\}$

$\{S4, S5, S6, S7, S8\}$, are not included due to their similarity compared to the other variations). In all cases, we obtain obstacle avoidance and target tracking.

The control actions are depicted in Fig. 4. Noteworthy, the smooth variants perform similarly in the output space even with tighter input magnitude bounds.

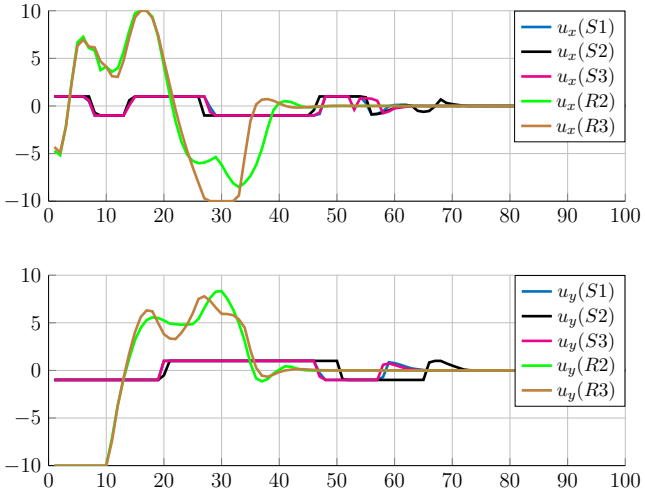


Fig. 4: Control inputs associated to the methods $\{S1, S2, S3, R2, R3\}$

Not least, in Fig. 5 we depict the path lengths (red curve, left axis) and computation times (blue curve, right axis).

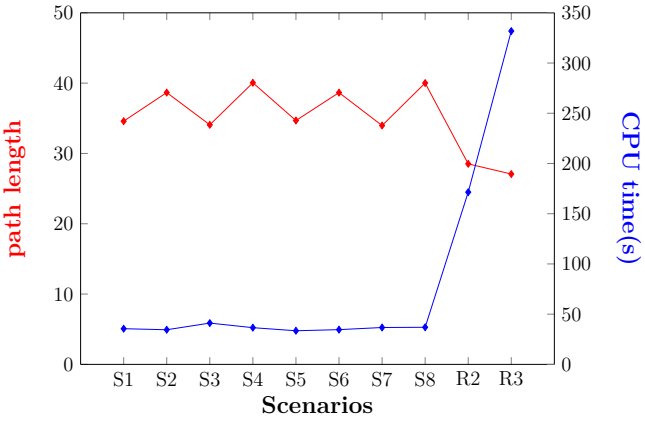


Fig. 5: Overall simulation results for methods $\{S1, S2, ..., S8, R2, R3\}$

As it can be clearly observed, all of the scenarios $\{S1, ..., S8\}$ which imply smoothing relaxations offer a reduced computational time (averaging 36.16 sec.), whereas the mixed-integer programming methods from [5] are significantly slower (171.45 sec. and 331.82 sec.), which is to be expected, due to the presence of binary variables. Nonetheless, the path lengths are similar and denote a similar performance.

Lastly, we depict the quiver plot (gradient map) of the exact overall sum function (7) and identify the points of local minimum (red circle) and maximum (blue square).

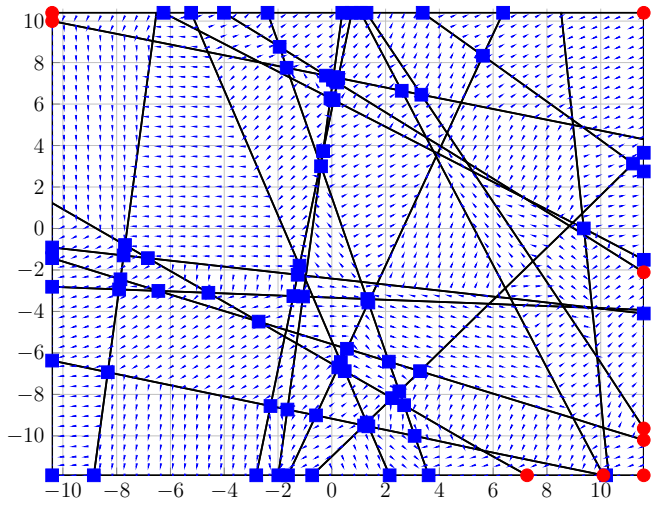


Fig. 6: Gradient orientation for a polyhedral environment alongside the associated local minima and maxima points

VI. Conclusion

Applying smooth approximations for the discontinuities appearing in the standard sum function construction we arrived at a nonlinear MPC implementation which proved computationally robust and fast. We analyzed the similarities between the exact (piecewise form) and smoothed approximations of the sum function and potential field surfaces.

As future work, we will consider ways to better analyze the critical points of the potential field surface: how to reduce their number by exploiting the degrees of freedom in the construction and how guarantee local minima escape with receding horizon control implementations.

References

[1] P. Falcone, F. Borrelli, J. Asgari, H. E. Tseng, and D. Hrovat, "Predictive active steering control for autonomous vehicle systems," *IEEE Transactions on control systems technology*, vol. 15, no. 3, pp. 566–580, 2007.

[2] S. Berkane, A. Bisoffi, and D. V. Dimarogonas, "Obstacle avoidance via hybrid feedback," *IEEE Transactions on Automatic Control*, vol. 67, no. 1, pp. 512–519, 2021.

[3] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Autonomous robot vehicles*. Springer, 1986, pp. 396–404.

[4] F. Stoican, T.-G. Nicu, and I. Prodan, "A mixed-integer mpc with polyhedral potential field cost for obstacle avoidance," in *2022 American Control Conference (ACC)*. IEEE, 2022, pp. 2039–2044.

[5] T.-G. Nicu, F. Stoican, and I. Prodan, "Polyhedral potential field constructions for obstacle avoidance in a receding horizon formulation," *IFAC-PapersOnLine*, vol. 55, no. 16, pp. 254–259, 2022.

[6] M. G. Park and M. C. Lee, "Real-time path planning in unknown environment and a virtual hill concept to escape local minima," in *30th Annual Conference of IEEE Industrial Electronics Society, 2004. IECON 2004*, vol. 3. IEEE, 2004, pp. 2223–2228.

[7] M. Guerra, D. Efimov, G. Zheng, and W. Perruquetti, "Avoiding local minima in the potential field method using input-to-state stability," *Control Engineering Practice*, vol. 55, pp. 174–184, 2016.

[8] C. Bordons and E. Camacho, "Model predictive control (advanced textbooks in control and signal processing)," 2004.

[9] I. Prodan, F. Stoican, S. Olaru, and S.-I. Niculescu, *Mixed-integer representations in control design: Mathematical foundations and applications*. Springer, 2016.
[10] N. Q. H. Tran, I. Prodan, E. I. Grøtli, L. Lefèvre, *et al.*, "Safe navigation in a coastal environment of multiple surface vehicles under uncertainties: A combined use of potential field constructions and NMPC," *Ocean Engineering*, vol. 216, p. 107706, 2020.
[11] C. Bliek1ú, P. Bonami, and A. Lodi, "Solving mixed-integer quadratic programming problems with ibm-cplex: a progress report," in *Proceedings of the twenty-sixth RAMP symposium*, 2014, pp. 16–17.
[12] B. Gao and L. Pavel, "On the properties of the softmax function with application in game theory and reinforcement learning," *arXiv preprint arXiv:1704.00805*, 2017.
[13] J. T. Barron, "A general and adaptive robust loss function," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4331–4339.
[14] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.