Bounding the uncertainity in nonlinear robust model predictive control using sphere covering

Shridhar K. Shah and Herbert G. Tanner

***Abstract*— We consider nonlinear continuous-time systems with additive model uncertainty. We design controllers based on a receding horizon optimization strategy, and we propose a new method to bound the uncertainty along the predicted trajectories. The bounds derived here are less conservative compared to existing methods, because the proposed method limits the exponential growth of the invariant cones around the nominal predicted trajectories. This is achieved by applying results from computational geometry, which allows us to cut and reset the width of the mouth of these cones through tunable control parameters. The method does not impose specific constraints on the structure of the uncertain term in the equations, other than assuming that it is locally Lipschitz and upper bounded.**

***Index Terms*— Robust model predictive control, nonlinear systems, computational geometry**

I. INTRODUCTION

Model predictive control has been found to be a reasonable and practical relaxation to the problem of infinite (time) horizon optimal control. Developed within control engineering practice, it has found applications in a wide range of problems, from chemical process control, to automotive [1] and robotics [2]. For linear dynamical systems, the theoretical framework of model predictive control is well developed and closed form solutions for the optimal control policies exist [3]. For nonlinear systems, however, due to the inherent complexity of the optimal control problem, model predictive control is still an active area of research [4].

Having an accurate system model is not always possible. Robustness to model uncertainties is therefore an important issue in control design, and model predictive control is no exception. There are different approaches in the literature to provide robustness within a (nonlinear) model predictive control framework. These can broadly be categorized as follows: (i) min-max optimization based approaches [5], [6]; (ii) approaches based on adaptive control, for uncertainties represented as unknown constant parameters [1]; (iii) auxiliary-controller based methods, where a robustifying feedback control action is pre-computed based on an offline characterization of uncertainity [7]; and (iv) using worst-case bounds around nominal trajectory computed based on bounds on the uncertain terms in the dynamics and Lipschitz constants for the nominal system. [8]. For a paper of this length, the literature coverage is necessarily incomplete, and we refer readers to [9], [10].

This work is supported by ARL MAST CTA # W911NF-08-2-0004

Shridhar Shah and Herbert Tanner are with Department of Mechanical Engineering, University of Delaware, Newark, DE, USA {shridhar, btanner}@udel.edu

Naturally, existing methods have limitations. Min-max approaches [5], [6] tend to be quite computationally demanding for online implementation. The use of auxiliary robust feedback [7], [8] to restrict the actual trajectories within invariant tubes around the nominal trajectory under the predicted optimal feedforward control is a reasonable solution, however, for general nonlinear systems —especially those subject to nonholonomic constraints— one may not have existence guarantees for these auxiliary inputs. In addition, as time evolves, worst-case bounds derived using Lipschitz constants [8] can become overly conservative to be practically useful. Finally, if the receding horizon control policy for the nominal system is not known before hand (e.g., in the form of fixed state feedback control gains) it is not clear how it can be computed on-line with some of the existing robust MPC methodologies. That is because in a real-time implementation, the system cannot wait until the next update time to compute the control input that is to be applied then; it should have the control law computed within the previous time step. If however it cannot be sure of where it is going to be at the end of this time step, planning of the trajectory beyond that point becomes problematic.

The contribution of this paper is the use of results from computational geometry, in conjunction with Lipschitz constant based methods, to (i) offer a clear way to compute the nominal receding horizon policies in real time, and (ii) bound the uncertainty around nominal model predictive control trajectories in a less conservative, and adjustable way. By adjustable, we mean that the size of the invariant sets (or cones) can not only be decreased by decreasing the control horizon, but can also be *reset* on-line at the end of each control horizon to a user-defined upper bound. To achieve this, we use results on variants of the sphere covering problem, specifically solutions to the *lattice covering* problem, to decompose the invariant set at the end of each control horizon into much smaller sets, out of which a different nominal solution can be computed. In that way, at the end of each control horizon, the uncertainty bound is at most as big as the the ball which the actual solution has landed at.

The sphere covering problem is a well studied problem in computational geometry [11]. Solutions to this problem are extensively used in computer science, in the context of coding theory, and numerical results are available [12]. The sphere covering problem can be defined as “finding the minimum number of spheres of specific size to cover a given larger sphere.” The lattice covering problem, on the other hand, is a special case of the sphere covering problem, in which the spheres are constrained to lay on the vertices of some lattice, and the problem is to find the optimal lattice structure [13]. Details on the solution can be found in [13] and are beyond the scope of this paper. There exist solutions for spaces of upto twenty four dimensions, and a list of the best known solutions can be found [14]. More details on these problems are given in [15].

The approach to bounding uncertainty that this paper outlines, starts by constructing invariant sets for future states around the trajectory of a nominal model under a given control input, using Lipschitz constants of the terms that describe the nominal system and the uncertainty model in the expression of the perturbed dynamics. Up to this point, the method is similar to that of [8]. We expand these sets to the full length of the *control* horizon, at which point we uniformly divide the set of all possible states at the end of the control horizon into finite smaller balls, based on the known (and off-line computed) solution of the lattice covering problem for the specific dimension, applied in scale to the specific case. The idea now is that we can use the control horizon to compute nominal (sub-optimal) trajectories originating from the center of each one of the balls covering the invariant set at the end of the control horizon. At that moment, the actual (perturbed) trajectory is bound to land in one of these smaller sets, and we will already have a (closest) pre-computed sub-optimal policy to use during the next step of the receding horizon step. What is different compared to [8], is that as soon as the ball containing the state is identified, the size of the invariant cone automatically shrinks to that of the small ball, and in subsequent time, the Lipschitz constant based estimates build on the small covering sphere rather than the full mouth of the invariant funnel. With successive application of this algorithm, we achieve a *constant* worst-case upper bound on the perturbed trajectories, without having to assume the existence of any auxiliary input. We demonstrate the method using a two-dimensional example and employ the best known 2D lattice covering (hexagonal packing).

The rest of the paper is organized as follows: Section II presents the model predictive control design problem formulation and introduces the basic common assumptions in robust model predictive control. Section III describes the proposed approach, and the coverage of the wide "mouth" of the invariant cones using the solution to the lattice covering problem. Section IV presents a new definition for optimality in a robust control framework, and shows formally why this definition is well posed. Our computational example follows in section V, and conclude the paper with section VI that summarizes the results and outlines ongoing research directions.

II. Preliminaries

A. Nominal model predictive control

Let the nominal (unperturbed) system dynamics be of the form

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

where $\hat{x} \in \mathbb{R}^n$, is the state vector and $u \in \mathbb{R}^m$ is the control input, and $m, n \in \mathbb{N}_+$. The function $f : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ is assumed to be locally Lipschitz:

Assumption 1: Function $f(x, u)$ is Lipschitz in x, uniformly in u *on the domain* $\mathbb{X}\times\mathbb{U}$, satisfying that $\forall x_1, x_2 \in \mathbb{X}$, and $u \in \mathbb{U}$

$$\|f(x_1, u) - f(x_2, u)\|_p \le L\|x_1 - x_2\|_p \;,$$

where L is the Lipschitz constant for f on $\mathbb{X}$, with respect to some p-norm.

Given an initial condition x_0 and a control input $u(\cdot)$, the solution of system (1) at time t of (1) under input $u(t)$, which passes through x_0 for $t = 0$ is defined as

$$\hat{x}^u(t; x_0) = x_0 + \int_0^t f\big(\hat{x}^u(\tau; x_0), u(\tau)\big)\, d\tau \;.$$

We use the ˆ notation to denote the trajectories of the nominal system (1) and distinguish them from those of the perturbed dynamics, which will be introduced shortly.

From an (initial) state x and for a specific input function $u(t)$, we define an (infinite horizon) cost as an integral over the trajectories $\hat{x}^u(t; x)$ of (1), given by a functional

$$J(x, u(\cdot)) \triangleq \int_0^\infty q\big(\hat{x}^u(\tau; x_0), u(\tau)\big)\, d\tau \;, \tag{2}$$

where $q(\cdot,\cdot)$ is a positive semi-definite function of both of its arguments, referred to as the *incremental cost*. The cost (2), computed over an infinite time horizon, can be approximated by the *finite horizon cost*, expressed as

$$J_T(x, u(\cdot)) \triangleq \int_0^T q(\hat{x}^u(\tau; x), u(\tau))d\tau + V\big(\hat{x}^u(T; x)\big), \tag{3}$$

where $V : \mathbb{R}^n \to \mathbb{R}_+$ is a function which approximates the tail of the infinite horizon integral in (2), and to ensure stability in the infinite horizon [16], it can be chosen as a control Lyapunov function which is also *compatible* with incremental cost $q(\cdot,\cdot)$ in the sense that

$$\min_u(\dot{V} + q)(x, u) \le 0 \;. \tag{4}$$

It can then be shown that under some additional reasonable assumptions on the existence of quadratic upper and lower bounds for the optimal and finite horizon costs [16], the system (1) is exponentially stable at the origin. In this context, the *optimal* finite horizon cost is be defined as

$$J_T^*(x) \triangleq \inf_{u(\cdot)} J_T\big(x, u(\cdot)\big) \;.$$

In what follows, we assume that states should remain inside an invariant set $\mathcal{W} \subset \mathbb{R}^n$, and satisfy the constraint $x(t) \notin \mathcal{O} \subset \mathbb{R}^n$ for all $t > 0$. In view of this, the *admissible* subset of the state space can be expressed as $\mathcal{W} \setminus \mathcal{O}$. It has been shown [2] that state constraints of this type can be incorporated in the framework of [16] if V is constructed as a navigation function modeling $\mathcal{O}$ as an "obstacle" region, and if in addition, either certain neighborhoods of the saddle configurations of V are excluded, or the stability condition (4) is relaxed into an integral version [2]. For the sake of simplicity, and trying to keep in this paper self-contained, we will assume that appropriate neighborhoods of the isolated points where $\nabla V(x) = 0$, $x \neq 0$, are excluded.

B. Robust model predictive navigation

Let us now consider the perturbed version of (1) in the form of the following continuous-time nonlinear system with additive uncertainty,

$$\dot{x} = f(x, u) + g(x, u) \ , \tag{5}$$

where, as before, $x \in \mathbb{X} \subseteq \mathbb{R}^n$ is the state and $u \in \mathbb{U} \subseteq \mathbb{R}^m$ is the control input. The vector-valued function $g(\cdot)$ can represent unmodeled dynamics. Obviously, $g(\cdot)$ is unknown; what we assume that we know about $g(\cdot)$ is that its norm is upper bounded for all time and states by some known positive constant

$$\|g(x, u)\| \leq \mu, \quad \forall x \in \mathbb{R}^n, \quad \mu > 0 \ . \tag{6}$$

For the sake of having well defined trajectories for (5) in the classical sense, we will assume that $g(\cdot)$ is locally Lipschitz in x, uniformly in u.

For an initial condition x_0 and a control input $u(\cdot)$, the trajectory of (5) can now be similarly to (1), defined as

$$x^u(t; x_0) = x_0 + \int_0^t f\big(x^u(\tau; x_0), u(\tau)\big)\, d\tau + \int_0^t g\big(x^u(\tau; x_0), u(\tau)\big)\, d\tau \ . \tag{7}$$

The finite horizon cost for the system (5) is defined similarly as (3)

$$J_T\big(x, u(\cdot)\big) \triangleq \int_0^T q\big(x^u(\tau; x), u(\tau)\big)\, d\tau + V\big(x^u(T; x)\big) \ .$$

However, the state trajectories $x^u(\tau; u)$, $\tau \in [0, T]$ cannot be known *a priori*, due to the effect of uncertainty. For a given time instant t, the set of possible states given any locally Lipschitz function $g(\cdot)$ satisfying (6), can be expressed as

$$\mathbb{X}_f(t; x_0) = \bigcup_{g(\tau); \tau \in [0,t]} x^u(t; x_0) \ .$$

C. Continuous dependence on initial state

It is straightforward to prove analytically the intuitive thought that for "small" uncertainties, the trajectories of (1) and (5) stay "close" to each other.

Theorem 1 ([17]): Let $f(t, x)$ be piecewise continuous in t and Lipchitz in x on $[t_0, t_1] \times W$ with a Lipchitz constant L, where $W \subset \mathbb{R}^n$ is an open connected set. Let $x(t)$ and $\hat{x}(t)$ be the solutions of

$$\dot{\hat{x}} = f(t, \hat{x}) \ , \quad \hat{x}(t_0) = \hat{x}_0 \quad \text{and}$$
$$\dot{x} = f(t, x) + g(t, x) \ , \quad x(t_0) = x_0 \ ,$$

respectively, such that $x(t), \hat{x}(t) \in W$, for all $t \in [t_0, t_1]$. Suppose that

$$\|g(t, x)\|_p \leq \mu, \quad \forall (t, x) \in [t_0, t_1] \times W \ ,$$

for some $\mu > 0$, and that

$$\|\hat{x}(t_0) - x(t_0)\|_p \leq \varepsilon \ .$$

Then $\forall t \in [t_0, t_1]$,

$$\|\hat{x}(t) - x(t)\|_p \leq \varepsilon \exp[L(t - t_0)] + \frac{\mu(\exp[L(t - t_0)] - 1)}{L} .$$

III. Invariant Sets over the Control Horizon

To facilitate the discussion that follows, let us introduce a set operation.

Definition 1 (Minkowski sum): Given $\mathcal{A} \subset \mathbb{R}^n$ and $\mathcal{B} \subset \mathbb{R}^n$, their Minkowski sum is defined as

$$\mathcal{A} \oplus \mathcal{B} \triangleq \{z \in \mathbb{R}^n \mid \exists\, x \in \mathcal{A}, y \in \mathcal{B} : z = x + y\} \ .$$

The Minkowski sum is probably a more formal and general way of expressing what is usually denoted in literature as $x + \mathcal{B}$ where x is a vector and $\mathcal{B}$ is a set (typically a ball).

A. Invariant sets

As illustrated in figure 1, under the influence of the uncertain term in (5), actual trajectories $x^u(t; x_0)$ diverge from the predicted nominal trajectory $\hat{x}^u(t; x_0)$ originating from a given point x_0. The Lipschitz constant L is used to construct bounds on how far away from $\hat{x}^u(t; x_0)$, perturbed trajectories $x^u(t; x_0)$ can be.

Let us assume that the initial state at time t_0, $x_0 = x(t_0)$ is not known exactly, but rather $x_0 \in \{\hat{x}_0\} \oplus \mathcal{B}_{\varepsilon_0}$, where $\mathcal{B}_{\varepsilon_0}$ denotes a ball of radius $\varepsilon_0 \leq \varepsilon$ and $\{\hat{x}_i\}$ is the center of closest small ball from current state at i^{th} control horizon and was used as initial points for computation of selected nominal trajectory. Based on theorem 1, the state $x^u(\tau; x_0)$ at time $\tau = t$ is inside the set

$$\mathbb{X}_f(t; x_0) = \{\hat{x}^u(t; x_0)\} \oplus \Lambda(t) \ ,$$

where,

$$\Lambda(t) = \left\{ z \in \mathbb{R}^n : \|z\| \leq \varepsilon e^{L(t - t_0)} + \frac{\mu(e^{L(t - t_0)} - 1)}{L} \right\} .$$

For example, the leftmost circle in figure 1 would correspond to $\{\hat{x}^u(\delta, x_0)\} \oplus \Lambda(\delta)$, whereas the middle cone would be expressed as $\mathbb{X}_f(\delta; x_1) = \{\hat{x}^u(\delta, \hat{x}_1)\} \oplus \Lambda(t)$ where $\|\hat{x}_1 - x(\delta)\| \leq \varepsilon$.

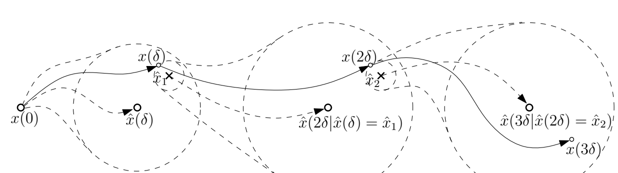

Fig. 1. Invariant sets at the end of each control horizon of length δ. Nominal trajectories are shown in dashed pointed lines, whereas actual trajectories are depicted solid. Around each terminal point $\hat{x}(\delta)$ of a nominal trajectory for the length of a control horizon, a ball is centered and shown in dashed outline, containing all possible perturbed solutions $x(\delta)$ after time δ. The dashed curves that surround each ball mark the boundaries of the invariant set along each control horizon.

As figure 1 hints at, the growth of sets $\mathbb{X}_f(\delta; x_0)$ with time can be restricted, if the initial state at the beginning of the next control horizon is known to an accuracy ε_0. Then the diameter of the set $\mathbb{X}_f(\delta+\Delta t; x_0)$, for a small $\Delta t > 0$, does not need to be larger than that of $\mathbb{X}_f(\delta; x_0)$; it can actually shrink back to $k\varepsilon$, where k is some pre-determined constant. The availability of state feedback at regular periods of (at most as big as) the control horizon, makes this feasible.

The challenge presented by on-line computation of receding horizon policies is that the "new" policy for the control horizon $(t+\delta, t+2\delta)$ needs to be precomputed in the time interval $(t, t+\delta)$ and be available for implementation at $t = (t+\delta)$. But if the "initial" state for the control horizon $(t+\delta, t+2\delta)$ is not known due to uncertainty, it is not clear how the receding horizon policy can be computed. Both issues, keeping the uncertainty bounds constant and enabling the application of (nominal) receding horizon policies computed on-line, can be addressed if the set $\mathbb{X}_f(t+\delta, x_0)$ is decomposed into balls $\mathbb{B}_i$, $i = 1, \ldots, N$, and control policies for the receded prediction horizon $(t+\delta, t+\delta+T)$ starting at each ball center, are computed during the time period $(t, t+\delta)$.

To ensure that the starting point of each precomputed policy is not more than ε away from the actual state of the perturbed trajectory $x^u(t+\delta; x_0)$, the balls need to be chosen so that any point in $\mathbb{X}_f(t+\delta, x_0)$ is not more than ε away than a ball center. If x_i denotes such the center of a ball $\mathbb{B}_i$. Then for a given set $\mathbb{X}_f$, the problem is formulated as

$$\begin{array}{ll} \text{Find} & x_i \in \mathbb{X}_f,\ i = 1, 2, .., N, \\ \text{s.t.} & \forall\, x \in \mathbb{X}_f,\ \inf_{x_i} \|x - x_i\|_p \leq \varepsilon\ . \end{array}$$

The following section presents a solution to this problem, using existing results from computational geometry.

B. Getting a constant bound on uncertainty

The problem stated in section III-A, is an instance of the well known *sphere covering problem*, for which the optimal solution is known for certain dimensions [11]. Unfortunately, computing these optimal solutions —even when such a solution is known to exist— is quite computationally demanding to find. There is a variation of the sphere covering problem, known as the *lattice covering problem* [13], restricts the centers of the spheres to form a lattice. Optimal solutions to the lattice covering problem are known for spaces of dimension up to 24. What is more, the optimal lattices can be computed off-line for a specific dimension, and then scaled and fitted in place at every $\mathbb{X}_f$.

Here, we are interested in lattice covering solutions. For the two-dimension cases, as the one treated in section V, the best known lattice covering is the hexagonal lattice [14] as shown in figure 2. Figure 2 shows the solution $N = 7$ obtained for $r = \frac{R}{2}$.

Any perturbed trajectory $x^u(\tau; x^{\{k\}})$, $\tau \in [0, \delta]$ of system (5) will land within the ball $\mathbb{X}_f(\delta; x^{\{k\}})$ at the end of a control horizon $\tau = \delta$. If it is given that for some $x_i \in$

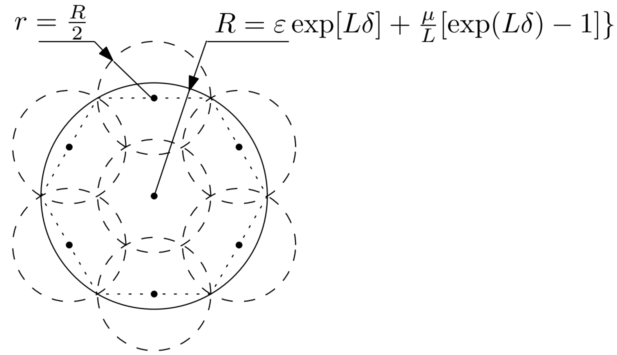

Fig. 2. Hexagonal lattice covering in two dimensions.

$\mathbb{X}_f(0; x^{\{k-1\}})$, it is $\|x_i - x^{\{k-1\}}\|$, then setting

$$r := \varepsilon\ , \qquad R := \varepsilon\, e^{L\delta} + \frac{\mu(e^{L\delta} - 1)}{L}\ , \tag{8}$$

yields the following expression for the minimum number of spheres[1] needed to cover the ball $\mathbb{X}_f(\delta; x^{\{k\}})$:

$$N \geq \frac{2\pi}{3\sqrt{3}} \left(e^{L\delta} + \frac{\mu(e^{L\delta} - 1)}{L\varepsilon} \right)^2\ . \tag{9}$$

The right hand side of (9) is actually a lower bound on the spheres of radius ε that are needed in general case not necessarily a lattice solution. It should be noted that actual number of balls required will be larger for a lattice solution. Also depending on the dimension of the space in which the balls are embedded, the optimal lattice structure differs. Hence, the calculation of number of balls will differ for different cases. If, however, (lattice) optimality is not required, one can pick their favorite lattice structure, and just scale it appropriately so that they fit at least as many as N small balls in the bigger ball of radius R.

C. The interplay between δ and ε

If the system (5) is subject to constraints of the form $x(t) \notin \mathcal{O} \subset \mathbb{R}^n$, then it may be of interest to be able to ensure that perturbed trajectories remain feasible, at least over the control horizon. It follows that for a control horizon δ, and from an initial state for the predicted trajectories x_0, one needs to verify that

$$\bigcup_{t \in [0,\delta]} \mathbb{X}_f(t; x_0) \notin \mathcal{O}\ .$$

With the proposed approach this is possible, either by adjusting the length of the control horizon δ or the uncertainty bound ε over the initial state at the beginning of a predicted nominal trajectory. In view of (8), a decrease of either δ or

[1]For the two-dimension cases, the minimum number of disks required to cover a disk of radius R, over all possible radii $r \leq R$ in limit is given by [18]

$$N \geq \frac{2\pi}{3\sqrt{3}} \left(\frac{R}{r} \right)^2 .$$

which represents the unavoidable overlapping of disks. This arrangement does not necessarily form a lattice but gives a lower bound on the minimum number of disks in general case.

ε results in a decrease of the radius R of $\mathbb{X}_f$. The control horizon δ influences R more strongly than ε, but in general, for a given value for R, a smaller value for one parameter allows a larger value for the other. Any reduction, however, comes at a cost: reducing δ means that the system has less time to compute the policy for the next control horizon; reducing ε means that there are more spheres covering the next ball $\mathbb{X}_f$ and thus, more predicted trajectories to compute. There are thus trade-offs, and the appropriate choice of δ and ε depends on the particular implementation.

IV. Feasibility and Optimality

Stability of a model predictive control scheme with (5), cannot be ensured based on (4) alone. In addition, the presence of state constraints raise the issue of trajectory feasibility. We formalize trajectory feasibility as follows.

Definition 2 (Feasibility): A trajectory $x(t)$ given in (7), is feasible for interval $[t_0, t_1]$ if $\forall t \in [t_0, t_1], x(t) \in \mathcal{W} \setminus \mathcal{O}$.

Stability and convergence for the closed-loop (5) under a model predictive control strategy is beyond the scope of this paper. The remaining of the section outlines briefly our intended approach, which involves strengthening (4) as follows.

Assumption 2: For (5) there exists a robust-control Lyapunov function and $\forall g(\cdot)$ and $x(t) \in \mathbb{X}_f$, $\exists u \in \mathbb{R}^n$ such that

$$\min_{u(\cdot)} \left(\frac{\partial V}{\partial x}(f+g) + q \right)(x(t), u(t)) \leq 0, \quad \forall t > 0 \ . \quad (10)$$

It has been noted [9], that in the presence of uncertainty, the principle of optimality cannot be invoked in general. Motivated by this fact, and as part of an ongoing robust stability analysis of a CLF-based model predictive control strategy for (5), we here introduce an alternative notion of optimality for the cost.

Definition 3 (Optimality): A trajectory $x(t)$ given in (7), is optimal for interval $[t_0, t_1]$ if $\forall t \in [t_0, t_1]$ and $\forall u(\cdot) \in \mathbb{R}^m$, $\exists u^*(\cdot)$ *and* $\exists g^*(\cdot)$ such that

$$\tilde{J}_T^*(x, u(\cdot)) \triangleq \inf_{u(\cdot), g(\cdot)} J_T(x, u(\cdot)) \ . \quad (11)$$

What (11) suggests when considering optimizing a finite horizon cost in the presence of uncertainty, is to treat $g(\cdot)$ as if it were an *input*. Instead of planning for the worst, (11) takes into account the best possible scenario for the uncertainty to define the optimal cost. This way, the optimal finite horizon cost will always be the lowest for any choice of control and occurrence of $g(\cdot)$. In this setting, we can prove that the use of navigation functions in the role of control Lyapunov functions as suggested in [2] results in optimal trajectories which are well defined.

Lemma 1: Given (10), if V is a navigation function attaining a maximum uniformly on the boundary of $\mathcal{O}$, then the optimal trajectory defined implicitly through (11) is always feasible.

Proof: Suppose that at initial time t_0 the state is in the feasible workspace $\mathcal{W} \setminus \mathcal{O}$, but optimal trajectory is infeasible. This means that $x(t_0) \in \mathcal{W} \setminus \mathcal{O}$, but $x(t) \in \mathcal{O}$ for some time $t \in [t_0, t_0 + \delta]$. If V is a navigation function, there must be $V(x(t_0)) < V(x(t))$. However, V is a robust Lyapunov function, which means that $\exists\, u(\cdot)$, such that $\forall g(\cdot)$, $\left(\frac{\partial V}{\partial x}(f+g) + q\right)(x(t), u(t)) \leq 0$. But if this condition holds for all $g(\cdot)$, it must also hold for the best-case $g^*(\cdot)$ for which the optimal trajectory is defined. Thus,

$$\left(\frac{\partial V}{\partial x}(f+g^*) + q \right)(x, u) \leq 0 \ . \quad (12)$$

Now since for every x, (12) must hold for some u, and given that whenever V increases, J_T increases too (because the incremental cost is always increasing), the optimal control input $u^*(\cdot)$ should be such that $\forall\, x$, $\left(\frac{\partial V}{\partial x}(f+g^*) + q\right)(x, u^*) \leq 0$. But then, if V always decreases along an optimal trajectory, it is impossible for $V(x(t)) > V(x_0)$, for any $t > 0$, which is a contradiction. ∎

It should be noted, however, that unless the model predictive control policy is designed to account for the worst possible case of $g(\cdot)$, as in the min-max based approaches, the perturbed closed-loop trajectories cannot be guaranteed to be feasible.

V. Example

Consider a planar mobile robot, which needs to be stabilized at the origin of $\mathbb{R}^2$, while avoiding a circular region $\mathcal{O}_1$ of radius 0.1 around $(0.2, 0.3)$ and staying always within a disc of radius 1, centered at the origin, the complement of which in $\mathbb{R}^2$ is denoted $\mathcal{O}_0$. The dynamics of the robot is that of a single integrator, perturbed by an uncertain term Δ

$$\dot{\mathbf{x}} = u(\mathbf{x}, t) + \Delta, \quad \mathbf{x}(0) = \mathbf{x}_0 \ ,$$

where $\mathbf{x}$ represents the position of the robot on the plane, and $u(\mathbf{x}, t)$ is the control law. The term Δ is considered unknown, but for the purposes of this example it is taken as $\Delta = 0.1 \sin(\mathbf{x}_0 - \mathbf{x})$; thus, $\mu = |\Delta| \leq 0.1$. Define a navigation function [19] $\phi(\mathbf{x})$ on $\mathbb{R}^2$, having the form

$$\phi(\mathbf{x}) = \left(\frac{\|\mathbf{x}\|^{2\kappa}}{\|\mathbf{x}\|^{2\kappa} + \beta_0 \beta_1} \right)^{\frac{1}{\kappa}} \ ,$$

where $\kappa = 2.3$, and $\beta_0 = 1 - \|\mathbf{x}\|^2$, $\beta_1 = \|\mathbf{x} - \mathbf{x}_1\|^2 - 0.1^2$ with $\mathbf{x}_1 = (0.2, 0.3)$. The robot is controlled using $u(\mathbf{x}) = -K \nabla \phi(\mathbf{x})$. Constant K is a gain taken 0.01, which defines the speed at which the robot is traveling. From that, the Lipschitz constant of the nominal system dynamics on $\mathbb{R}^2 \setminus (\mathcal{O}_0 \cup \mathcal{O}_1)$ is estimated at $L = 3$.

The nominal and perturbed trajectories are calculated over the period of two control horizons each of size $\delta = 0.1$. With the initial condition known exactly, the radius of $\mathbb{X}_f(\delta; \mathbf{x}_0)$ is found as $R^{\{1\}} = \frac{\mu}{L}(e^{L\delta} - 1) = 0.0117$. Let $\varepsilon = 0.0117$, so that the first ball $\mathbb{X}_f$ does not need to be covered. En route, the robot computes its control input for the next control horizon; without being able to predict exactly where it is going to land in $\mathbb{X}_f(\delta; \mathbf{x}_0)$, it computes this input based on the nominal dynamics and assuming that it starts its second control horizon starts at the center of $\mathbb{X}_f(\delta; \mathbf{x}_0)$.

At the end of the first control horizon, after time δ, figure 4 shows that the robot slightly overshot its prediction, but still

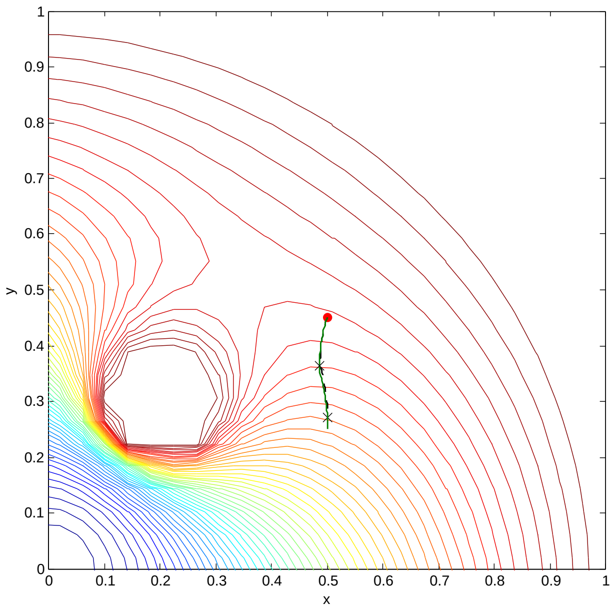

Fig. 3. Contour lines of a navigation function with one spherical obstacle at $(0.2, 0.3)$ with radius 0.1. The boundary of the workspace is a circle of radius 1.0, while the desired position is the origin. The initial position for the system is marked with a dot, while the nominal and perturbed trajectories are shown by dotted and solid lines respectively; the $\times$ marker indicate the end of a control horizon.

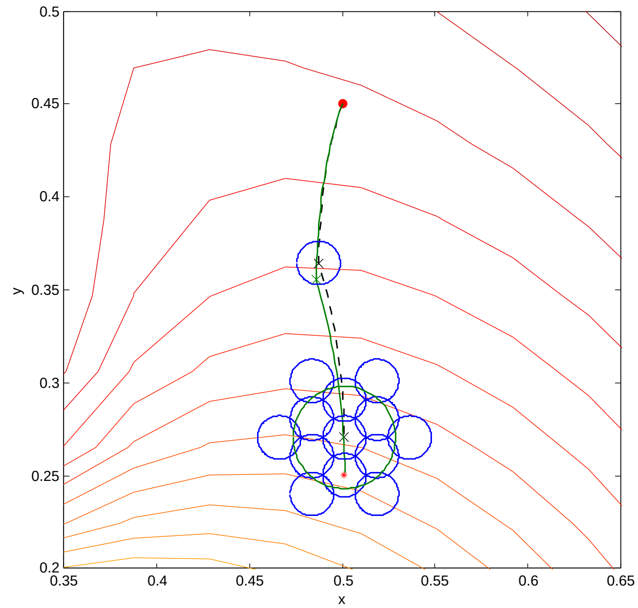

Fig. 4. An scaled portion of Fig. 3, showing the predicted (dashed) and perturbed (solid) trajectories more clearly, along with the sphere covering. At the end of the first control horizon, the actual state is within $R = 0.0117$ distance from the predicted, with the bound marked by a green circle. At the end of the second control horizon, the radius of the uncertainty region has grown to $R = 0.0275$, and $N = 13$ smaller balls of radius $r = 0.0117$ are used to cover it.

landed within the ball $\mathbb{X}_f(\delta; \mathbf{x}_0)$ of radius 0.0117. At time δ, the robot implements its precomputed policy for the period $[\delta, 2\delta]$, and based on its nominal model it predicts the center of $\mathbb{X}_f(\delta; \mathbf{x}^{\{1\}})$, which now has a radius $R^{\{2\}} = R^{\{1\}} e^{L\delta} + \frac{\mu}{L}[e^{L\delta} - 1] = 0.0275$. To cover $\mathbb{X}_f(\delta; \mathbf{x}^{\{1\}})$ with balls of radius $\varepsilon = 0.0117$ we need $N = 13$ such balls (figure 4). From the center of each one of these balls, a new receding horizon policy can be computed. When the robot arrives at $\mathbb{X}_f(\delta, \mathbf{x}^{\{1\}})$ it can pick the policy that is associated with an initial condition at the ball center which is closer to its current position. In Fig. 4, that center is marked with a small red * marker. Proceeding this way, the actual state at the end of each control horizon is not more than $R = \varepsilon e^{L\delta} + \frac{\mu}{L}[e^{L\delta} - 1] = 0.0275$ from the predicted state at that time.

VI. Conclusions and Future Work

Using known results from computational geometry it is possible to keep the bounds of uncertainty around the predicted trajectories at the end of each control horizon, bounded. The uncertainty bound is tunable, and depends on the length of the control horizon, and the number of sample trajectories computed from that point on, which is ultimately related to the amount of additional computational overhead that can be afforded in order to keep the uncertainty from growing exponentially. Ongoing work is along completing a framework for robust model predictive control based on navigation functions in the role of robust control Lyapunov functions. Along these lines it is expected to use lattices in more than two dimensions, in which case the trade-offs between the additional computational overhead and the reduction in the uncertainty cones will be revealed.

References

[1] D. DeHaan and M. Guay, *Model Predictive Control*, T. Zheng, Ed. Sciyo, 2010.

[2] H. Tanner and J. Piovesan, "Randomized receding horizon navigation," *Automatic Control, IEEE Transactions on*, vol. 55, no. 11, pp. 2640 –2644, 2010.

[3] E. F. Camacho and C. Bordons Alba, *Model predictive control*, 2nd ed. New York : Springer, 2004.

[4] E. Camacho and C. Bordons, "Nonlinear Model Predictive Control: An Introductory Review," 2007, pp. 1–16.

[5] D. Ramirez, T. Alamo, and E. Camacho, "Efficient implementation of constrained min-max model predictive control with bounded uncertainties," in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, vol. 3, dec. 2002, pp. 3168 – 3173.

[6] D. Ramrez, T. Alamo, E. Camacho, and D. M. de la Pea, "Min-max mpc based on a computationally efficient upper bound of the worst case cost," *Journal of Process Control*, vol. 16, no. 5, pp. 511 – 519, 2006.

[7] J. M. Carson III, "Robust model predictive control with a reactive safety mode." Ph.D. dissertation, California Institute of Technology, 2008.

[8] D. Marruedo, T. Alamo, and E. Camacho, "Input-to-state stable mpc for constrained discrete-time nonlinear systems with bounded additive uncertainties," in *Proceedings of the 41st IEEE Conference on Decision and Control 2002*, vol. 4, dec. 2002, pp. 4619 – 4624 vol.4.

[9] A. A. Jalali and V. Nadimi, "A survey on robust model predictive control from 1999-2006," in *Proceedings of the International Conference on Computational Inteligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce*. Washington, DC, USA: IEEE Computer Society, 2006, pp. 207–.

[10] A. Bemporad and M. Morari, "Robust model predictive control: A survey," in *Robustness in identification and control*, ser. Lecture Notes in Control and Information Sciences, A. Garulli and A. Tesi, Eds. Springer Berlin / Heidelberg, 1999, vol. 245, pp. 207–226.

[11] I. Dumer, "Covering spheres with spheres," *Discrete and Computational Geometry*, vol. 38, pp. 665–679, 2007.

[12] N. J. A. Sloane, R. H. Hardin, W. D. Smith *et al.* Tables of spherical codes. [Online]. Available: www.research.att.com/∼njas/packings/

[13] F. Vallentin, "Sphere covering, lattices, and tilings (in low dimensions)," Ph.D. dissertation, Technische Universität Münchena, 2003.

[14] A. Schrmann, F. Vallentin *et al.* Best known lattice coverings in dimension 1–24. [Online]. Available: http://fma2.math.uni-magdeburg.de/∼latgeo/covering_table.html

[15] J. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*. New York : Springer, 1999.

[16] A. Jadbabaie, "Receding horizon control of nonlinear systems: a control lyapunov function approach." Ph.D. dissertation, California Institute of Technology, 2001.

[17] H. K. Khalil, *Nonlinear Systems (3rd Edition)*, 3rd ed. Prentice Hall, December 2001.

[18] R. Kershner, "The number of circles covering a set," *American Journal of Mathematics*, vol. 61, no. 3, pp. pp. 665–671, 1939.

[19] D. E. Koditschek and E. Rimon, "Robot navigation functions on manifolds with boundary," *Adv. Appl. Math.*, vol. 11, no. 4, pp. 412–442, 1990.