Robust Obstacle Avoidance using Tube NMPC

Gowtham Garimella*, Matthew Sheckells*, Joseph Moore† and Marin Kobilarov*

*Department of Computer Science and Mechanical Engineering
Johns Hopkins University, 3400 N Charles Str, Baltimore, MD 21218, USA.
Email: ggarime1—msheckells—marin@jhu.edu
†The Johns Hopkins University Applied Physics Laboratory,
11100 Johns Hopkins Road, Laurel, Maryland 20723, USA.
Email: Joseph.Moore@jhuapl.edu

Abstract—This work considers the problem of avoiding obstacles for general nonlinear systems subject to disturbances. Obstacle avoidance is achieved by computing disturbance invariant sets along a nominal trajectory and ensuring these invariant sets do not intersect with obstacles. We develop a novel technique to compute approximate disturbance invariant sets for general nonlinear systems using a set of finite dimensional optimizations. A bi-level NMPC optimization strategy alternates between optimizing over the nominal trajectory and finding the disturbance invariant sets. Simulation results show that the proposed algorithm is able to generate disturbance invariant sets for standard 3D aerial and planar ground vehicles models, and the NMPC algorithm successfully computes obstacle avoidance trajectories using the disturbance invariant sets.

I. INTRODUCTION

Consider an agile aerial vehicle navigating autonomously to a goal in an obstacle-ridden environment subject to natural disturbances arising from wind or propeller downwash. This type of task is challenging since one has to account for the effect of disturbances on the system dynamics while planning a trajectory to avoid obstacles. Moreover, the disturbances are sometimes dependent on the state of the vehicle and, therefore, have a nonlinear effect on the propagation of the dynamics. For example, disturbances due to wall effects [16] scale based on the distance to wall. To account for the effects of uncertainty, current methods plan for a trajectory around the obstacles such that the obstacles do not intersect with the invariant set centered around the trajectory. The invariant set is defined as the region in state space in which the system is guaranteed to stay under the effect of disturbances for a given controller [11]. These methods usually depend on the structure of the dynamics or the controller and are not applicable to general nonlinear systems with nonlinear noise models. The goal of this work is, therefore, two-fold. We first propose a novel method to find approximate invariant sets for a general nonlinear dynamic system that can handle nonlinear noise models and which extends to higher dimensional systems. Next, we formulate a Nonlinear Model Predictive Control (NMPC) optimization that computes a nominal trajectory which avoids obstacles by relying on the computed invariant regions. Combining both the steps, we aim to tackle the obstacle avoidance problem for general nonlinear systems under external disturbances without any assumptions about the structure of the controller or the dynamics. For instance, one such obstacle avoiding trajectory

generated for a simple wheeled vehicle with a nonlinear noise model is shown in Figure 1.
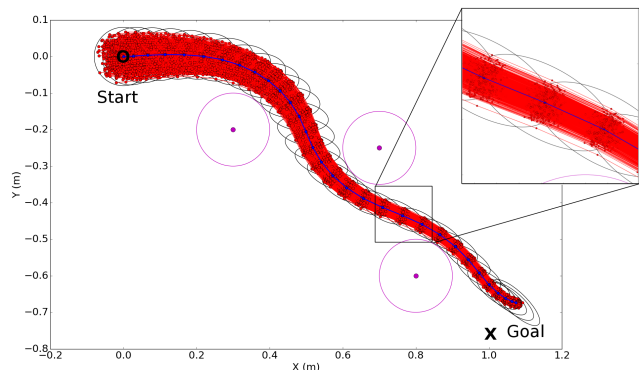

Fig. 1. Unicycle model subject to external disturbances avoiding obstacles. The blue dots and lines are the nominal trajectory while the red dots and lines are sample trajectories propagated using the computed control law.

A. Related work

Disturbance-invariant sets provide one means of studying the effects of uncertainty on dynamical systems. These are regions in the state-space of a system that guarantee that if the system starts in the invariant set, then the system will lie in the invariant set under bounded disturbances. Conservative approximations for invariant sets have been computed for linear systems [21] and for piece-wise affine systems [22].

Tube-based Model Predictive Control (MPC) techniques have been developed that compute the invariant sets online for linear dynamics [15] and have also been extended to certain classes of nonlinear-discrete systems [23]. These techniques design controllers that provide "robust asymptotic stability" to the control invariant region.

Semi-definite optimization techniques [18], such as sum-of-squares (SOS) programming, search for polynomial functions that are positive and produced from a sum of squared monomials. Tobenkin et al. [27] use SOS programming to compute funnels for nonlinear dynamics with a LQR controller. These funnels are time-varying state space regions such that if the system starts from the mouth of the funnel, it is guaranteed to stay in the funnel for all future time and eventually end

up at the goal. By combining funnel computation with motion primitives, Majumdar et al. [11] were able to compute robust funnels for a few primitives and perform online optimization on the order and duration of execution of each of the primitives to avoid obstacles. [25] explored finite-time verification for stochastic systems.

SOS programming has also been used to compute funnels for general Lipschitz nonlinear systems [28]. This approach computes the funnel around a nominal trajectory using invariant coordinates and optimizes only the nominal trajectory online. Unlike the previous method, this approach does not require recomputing funnels for every nominal trajectory since the funnel is specified in invariant coordinates around the nominal trajectory, and transformations to the nominal trajectory also transform the funnel around it. The main drawback with this approach is that the computation of the Lipschitz constant for a system is non-trivial [24]. Gao et al. applied this work to autonomous ground vehicles [7] and showed that tube MPC can avoid obstacles safely while following a desired trajectory.

Manchester et al. introduced Control Contraction Metrics (CCMs) as an alternative to Lyapunov functions to design globally stabilizing controllers for general nonlinear systems [12, 13]. Singh et al. used CCMs to compute the funnels for a nonlinear system [24]. In this approach, the funnel is computed in invariant coordinates and only the nominal trajectory is optimized online to perform obstacle avoidance. The success of this method relies on computing a valid CCM for the nonlinear system using SOS programming.

A different class of methods uses game-theory to find the worst possible noise for a given control [2, 26]. These methods usually contain a bi-level optimization [4] scheme in which the outer optimization searches for the controller that stabilizes the system while the inner-loop optimizes over the constraints to satisfy the process noise. The usual drawback with these methods is that they do not extend well to higher dimensional systems [24]. The approach presented in this paper falls into this class. The algorithm presented here alleviates to some extent the curse of dimensionality by using a novel method to find the funnel and avoids the usual inner-outer optimization with a sequential optimization scheme that is shown to work well in the experiments conducted.

In addition to funnel methods, there are a class of techniques that treat states as a distribution and plan in the space of the distributions known as belief space [17, 19]. These can handle unbounded disturbances as long as the expected trajectory cost which they minimize is finite. Work done by Desaraju et. al [5] showed belief space planning for a small quadrotor platform. Chance constrained programming [1] is a further extension of these methods to handle probabilistic state constraints. In chance constrained programming, the obstacle constraints and other state constraints are respected only in probability. Although chance constrained programming is usually restricted to linear systems due to the difficulty in propagating multivariate distributions through nonlinear dynamics, there have been some attempts to extend these techniques to nonlinear systems [10, 6].

The methods discussed so far provide either deterministic or probabilistic guarantees for the stability of a controller under process noise. There also exists a class of risk-sensitive controllers [3] that provide some robustness to process noise without providing any guarantees on the performance of the controllers. For example, Manchester et al. optimized controllers to minimize their sensitivity to disturbances while not necessarily providing guarantees in terms of state-space funnels [14]. This method scaled well to higher dimensional systems such as robot arms and quadrotor models.

B. Existing challenges

In general, the aim of the methods discussed above is to stabilize a general nonlinear system to a goal state under bounded disturbances. They use state-space funnels to describe the possible paths the uncertain system might take for given a noise model. The funnel computation described in the methods discussed above are limited in several ways. Work done Majumdar et. al [11] requires the computation of a Lyapunov candidate using SOS programming and therefore is limited to polynomial dynamics. Similarly, funnel computation for Lipschitz nonlinear systems has been shown by [28], which requires computing the Lipschitz constant of the system. The work done by Singh et al. [24] also relies on SOS for computing a Control Contraction Metric to define the funnel. In addition, this method assumes an additive noise model which can be restrictive in practice. In addition, most methods discussed above presented results for systems with only a small number of dimensions and assumed linear noise models. For example, Singh et al. showed funnel computation for only a planar quadrotor, which is a six dimensional system [24] with crosswind disturbances. Work done by Majumdar et al. [11] is among the few examples of high-dimensional funnel computation for a quadrotor, a 12-dimensional system.

C. Contributions

Our approach to finding the disturbance invariant set is to reduce the invariant set computation to a set of finite dimensional optimizations which provide an approximation to the invariant set, and a sequential NMPC is formulated to avoid obstacles using the approximate invariant sets computed. The NMPC consists of two optimizations running one after another. In the first optimization, the nominal trajectory is optimized to ensure the approximate invariant set around it avoids the obstacles. In the second optimization, the invariant set approximation is improved given the dynamics, the nominal trajectory, and the controller.

The algorithm, although providing only approximate guarantees, based on the experiments conducted, extends well for higher dimensional systems and can handle nonlinear noise models. For example, the quadrotor example shown in this work is a 14 dimensional system with a nonlinear backstepping controller. The presented approach does not rely on the specific structure of the dynamics, controller, or noise. Hence, it can be used to work with controllers without explicit stability certificates. For example, this method allows learning based

controllers (e.g. [20, 8]) to be implemented in practice by providing approximate guarantees using the invariant sets. Based on experiments performed on different systems, we show that the proposed method is able to compute good approximations of the invariant sets for high dimensional systems and effectively leverage them for robust obstacle avoidance.

The rest of the paper is organized as follows. In §II, we propose the algorithm to compute the invariant sets around a nominal trajectory. Continuing, in §III, the invariant sets are used to formulate a NMPC optimization to perform obstacle avoidance under disturbances. Section IV shows simulation results on unicycle and quadrotor models. Finally, conclusions and future work are presented in §V.

II. COMPUTING INVARIANT FUNNELS

A. Disturbance Invariant Sets

Consider a discrete nonlinear dynamic system with state $x \in \mathbb{R}^n$, control $u \in \mathbb{R}^m$, and a disturbance $w \in \mathbb{R}^n$. The dynamics can be written as

$$x_{i+1} = f(x_i, u_i, w_i) \tag{1}$$

where i denotes the discrete time index and the disturbance is assumed to be bounded (i.e. $\|w_i\| \leq \epsilon_i$). We assume a feedback controller ψ based on the current state x_i, a given goal state $\bar{x}_i$, and feed-forward controls $\bar{u}_i$ to compute the control u_i:

$$u_i = \psi(x_i, \bar{x}_i, \bar{u}_i). \tag{2}$$

The controller usually has additional parameters, such as feedback gains, which are assumed to be fixed and known in the rest of the work.

Assuming the state at step i is in some enclosing region P_i (i.e. $x_i \in P_i$), the disturbance invariant set at stage $i + 1$ is defined as

$$P_{i+1} = \{x_{i+1} = f(x_i, \psi(x_i, \bar{x}_i, \bar{u}_i), w_i) \mid x_i \in P_i, \|w_i\| \leq \epsilon_i\}.$$

For a general nonlinear system with a nonlinear controller, finding the disturbance set is intractable.

One solution to this problem [26] is to use a conservative estimate of P_i denoted by C_i, i.e. $P_i \subseteq C_i$, and propagate C_i to C_{i+1} subject to the closed loop dynamics. Since the enclosing region C_i contains P_i, the enclosing region C_{i+1} also encloses P_{i+1}. Constraints, such as obstacles, can then be handled in a robust fashion by ensuring no overlap with the enclosing regions C_i. For the rest of the section, we choose the shape of C_i to be an ellipsoid with the same dimension as the state x_i. The choice of the family of enclosing region (e.g. ellipsoid, cuboid, sphere) affects the dilation between P_i and C_i and therefore how conservative the resulting obstacle avoidance trajectory will be. Furthermore, the estimate C_i becomes more and more conservative as the index i increases since the ellipsoid C_{i+1} is propagated from ellipsoid C_i instead of the actual region P_i. In spite of these drawbacks, propagating and reasoning in terms of conservative regions permits robust NMPC-based obstacle avoidance under disturbances.

To formulate the ellipsoid propagation problem, first we define the region of dynamics obtained by propagating the ellipsoid C_i through the dynamics as

$$\bar{P}_{i+1} = \{x = f(x_i, \psi(x_i, \bar{x}_i, \bar{u}_i), w_i) \mid x_i \in C_i, \|w_i\| \leq \epsilon_i\}.$$

Unlike, the region P_{i+1} which is propagated from P_i, the region $\bar{P}_{i+1}$ is propagated from the previous ellipsoid C_i (see Figure 2). Hence, the region $\bar{P}_{i+1}$ contains the region P_{i+1} (i.e. $P_{i+1} \subseteq \bar{P}_{i+1} \subseteq C_{i+1}$).

Finding the least conservative enclosing ellipsoid can be mathematically stated as

$$\min_{C_{i+1}} Vol(C_{i+1}) \text{ s.t } \bar{P}_{i+1} \subseteq C_{i+1}, \tag{3}$$

where $Vol(\cdot)$ denotes the volume of the ellipsoid. We simplify the above problem by constraining the centers of the ellipsoids to follow a nominal trajectory without disturbances. Assuming the center of C_i to be μ_i and the center of the C_{i+1} to be μ_{i+1}, the nominal trajectory dynamics are written as $\mu_{i+1} = f(\mu_i, u_i, 0)$. The control u_i used in the nominal trajectory dynamics is provided by the controller with the goal being the same as the current state, i.e $u_i = \psi(\mu_i, \mu_i, \bar{u}_i)$. The feed-forward terms in the controller move the nominal state μ_i to the next nominal state μ_{i+1}. The control for any other point in the invariant set is given by $u_i = \psi(x_i, \mu_i, \bar{u}_i)$.
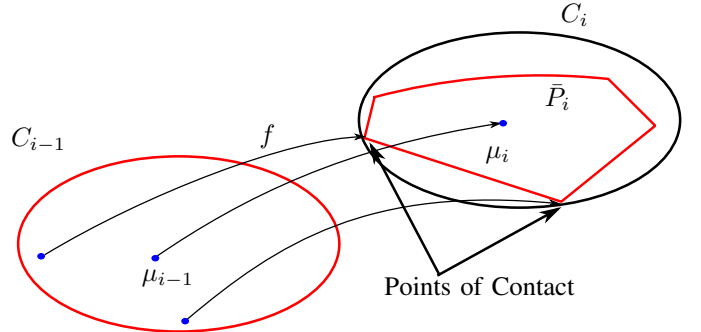
Fig. 2. The region $\bar{P}_i$ is propagated from the previous ellipsoid C_{i-1} using the closed-loop dynamics f. The algorithm for computing the enclosing ellipsoids finds points in C_{i-1} (shown in blue) that when propagated become the points of contact between $\bar{P}_i$ and the enclosing ellipsoid C_i.

The ellipsoid optimization problem is intractable since the constraint $\bar{P}_{i+1} \subseteq C_{i+1}$ should be satisfied for all the disturbances $\|w_i\| \leq \epsilon_i$ and for all starting states $x_i \in C_i$. One approach to solve this problem is through SOS programming [18] under certain assumptions on the structure of the dynamics. In this work, we take a different approach and transform the above optimization problem into a finite dimensional optimization to find a feasible, but not necessarily the least conservative, ellipsoid.

B. Algorithm to Find Enclosing Ellipsoid

The algorithm has to find the radii $r_{i+1} \in \mathbb{R}^n_{>0}$ and principal axes $R_{i+1} \in SO(n)$ of the ellipsoid $C_{i+1} = \{r_{i+1}, R_{i+1}\}$ which encloses the propagated region $\bar{P}_{i+1}$ assuming the

center of the ellipsoid is along some nominal trajectory. The procedure to find the ellipsoid consists of n successive optimization problems. Each optimization problem finds a principal axis and the ellipsoid radius along that principal axis. The first optimization simply searches for the farthest point from the center in the region $\bar{P}_{i+1}$:

$$r_{i+1,1}^2 = \max_{x \in \bar{P}_{i+1}} (x - \mu_{i+1})^T (x - \mu_{i+1}). \qquad (4)$$

The first radius and principal axis are chosen as $r_{i+1,1} = \|x_1 - \mu_{i+1}\|$ and $e_{i+1,1} = (1/r_{i+1,1})(x_1 - \mu_{i+1})$, respectively, where x_1 minimizes (4). Continuing, at step j of the optimization, we find a point $x_j \in \bar{P}_{i+1}$ that maximizes the radius along j-th principal axis assuming the point is on the boundary of the ellipsoid. The j-th radius $r_{i+1,j}$ and the principal axis $e_{i+1,j}$ of the ellipsoid assuming the point is on the boundary of the ellipsoid are given by

$$r_{i+1,j} = \frac{|x_{jj}|}{\sqrt{1 - \sum_{k=1}^{j-1} \left(\frac{x_{jk}}{r_{i+1,k}}\right)^2}}, \qquad (5)$$

$$e_{i+1,j} = (x_j - \mu) - \sum_{k=1}^{j-1} e_{i+1,k} x_{jk}, \qquad (6)$$

where x_{jk} is the projection of the point x_j along kth principal axis, i.e. $x_{jk} = (x_j - \mu_{i+1})^T e_{i+1,k}$. The vector $e_{i+1,j}$ is then normalized to ensure the jth principal axis is a unit vector. The j-th optimization problem can be formulated as

$$\max_{x_j \in \bar{P}_{i+1}} r_{i+1,j}^2, \quad s.t. \quad x_{jj} \neq 0. \qquad (7)$$

The ellipsoid at C_{i+1} is given by

$$C_{i+1} = \{ \; r_{i+1} = (r_{i+1,1}, \cdots, r_{i+1,n}),$$
$$R_{i+1} = [e_{i+1,1} \mid e_{i+1,2} \mid \cdots \mid e_{i+1,n}] \; \} \qquad (8)$$

Next, we prove that the ellipsoid generated by the above algorithm completely encloses the region $\bar{P}_{i+1}$ induced by the closed-loop dynamics. To start with we prove the following lemma:

Lemma 1: The cost function used in optimization (7) is bounded and $r_{i+1,j-1} \geq r_{i+1,j} \; \forall j \in \{2, \cdots, n\}$.

Proof: For $j = 1$, $r_{i+1,1}$ is given by (4). For any point x_2 in the region $\bar{P}_{i+1}$, using the definitions

$$x_{21} = (x_2 - \mu_{i+1})^T e_1, \qquad x_{22} = \|x_2 - x_{21} e_1\|,$$

we have $x_{21}^2 + x_{22}^2 = \|x_2 - \mu_{i+1}\|^2 \leq r_{i+1,1}^2$. Using the inequality, the cost function for $j = 2$ can be upper bounded as

$$r_{i+1,2}^2 = \frac{x_{22}^2}{1 - \left(\frac{x_{21}}{r_{i+1,1}}\right)^2} \leq \frac{r_{i+1,1}^2 - x_{11}^2}{1 - \left(\frac{x_{21}}{r_{i+1,1}}\right)^2} = r_{i+1,1}^2. \qquad (9)$$

Next, for any $j > 1$, we already have r_{j-1} obtained by maximizing the cost function (7) at $j - 1$. Thus, for the point x_j in the region $\bar{P}_{i+1}$ which maximizes r_j, the following inequality holds

$$\frac{x_{jj}^2 + x_{j,j-1}^2}{1 - \sum_{k=1}^{j-2} \left(\frac{x_{jk}}{r_{i+1,k}}\right)^2} \leq r_{i+1,j-1}^2, \qquad (10)$$

where x_{jj} is the residual left over after subtracting the components of $(x_j - \mu_{i+1})$ along the principal axes $\{e_1, e_2, \ldots, e_{j-1}\}$. This inequality can be transformed as

$$\frac{x_{jj}^2}{r_{i+1,j-1}^2} \leq 1 - \sum_{k=1}^{j-1} \left(\frac{x_{jk}}{r_{i+1,k}}\right)^2. \qquad (11)$$

The cost function for the optimization at stage j can then be bounded as

$$r_{i+1,j}^2 = \frac{x_{jj}^2}{1 - \sum_{k=1}^{j-1} \left(\frac{x_{jk}}{r_{i+1,k}}\right)^2} \leq r_{i+1,j-1}^2. \qquad (12)$$

Combining this inequality with the inequality for $j = 1$ given in (9), the lemma holds. ∎

Note that, we assume x_{jj} is not equal to zero while optimizing in (7). According to the inequality (11), for $x_{jj} \neq 0$, the right hand side of the inequality is greater than 0. Thus, the denominator of the $r_{i+1,j}$ is greater than zero. When x_{jj} is exactly zero, the cost function is undefined as both numerator and denominator are zero. Therefore, we avoid those points while minimizing the cost function.

Theorem 1: The ellipsoid C_{i+1} as defined in (8) encloses the region $\bar{P}_{i+1}$.

Proof: Using the Lemma 1, we showed that the cost function is bounded. Now, consider the n-th optimization problem above which maximizes the last principal axis r_n. Since we find the maximum possible radius along the last principal axis, the radius computed using any other point $x_n \in \bar{P}_{i+1}$ is going to be less than $r_{i+1,n}$ i.e.

$$\frac{x_{nn}}{\sqrt{1 - \sum_{k=1}^{n-1} \left(\frac{x_{nk}}{r_{i+1,k}}\right)^2}} \leq r_{i+1,n} \qquad (13)$$

$$\Rightarrow \sum_{k=1}^{n} \left(\frac{x_{nk}}{r_{i+1,k}}\right)^2 \leq 1 \qquad (14)$$

where x_{nk} is the projection of the point x_n along the principal axis e_k (i.e. $x_{nk} = (x_n - \mu)^T e_k$). The inequality (14) then implies that C encloses the region $\bar{P}_{i+1}$. ∎

C. Simplification for cases near singularity

The cost function in (7) can become ill-defined when $x_{jj} \to 0$. Close to the singularities, the maximization of the cost function produces ellipsoids that are very conservative. Therefore, we develop an alternative optimization scheme that maximizes over x_{jj} instead of r_j when such singularities are encountered. The simplification improves the convergence of the optimization with the drawback that it produces an

ellipsoid which only approximately encloses the propagated dynamics, i.e. the computed C_i may not completely enclose $\bar{P}_i$. The possible error between an ellipsoid computed by maximizing x_{jj} instead of r_j for a two-dimensional system is illustrated in Figure 3. The maximum error in the radius that can be encountered using this approximation is $\|r_j - r_{j+1}\|$ where r_j is computed by maximizing (7) and r_{j+1} is obtained using the proposed simplification.
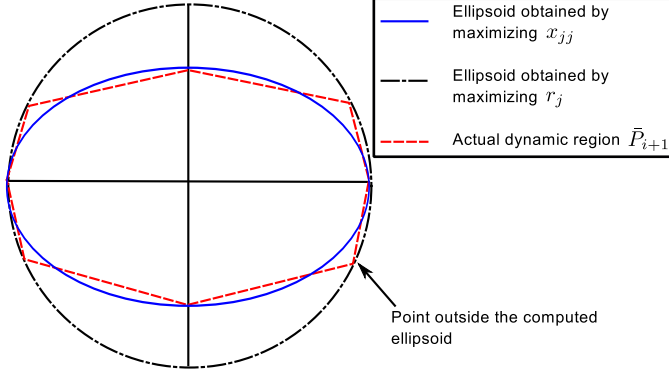

Fig. 3. Error induced by simplification of the maximizing cost function. The ellipsoid obtained by maximizing only the projection x_{jj} shown in blue leads to a smaller ellipsoid, although it leaves out some parts of the dynamics region shown in red. The ellipsoid obtained by maximizing the minor axis r_j is shown in black and completely encloses the dynamics region.

The optimization scheme used in practice at stage j is given by

$$\min_{x_j} (x_{jj} - r_{max})^2, \ \ s.t \ x_j \in \bar{P}_{i+1}, \quad (15)$$

where the maximization in (7) is simplified and converted into a nonlinear least squares minimization where r_{max} is chosen such that $x_{jj} < r_{max}$. The least squares minimization is a local optimization technique, and hence the solutions obtained are not guaranteed to be global minima.

D. Summary

In summary, the ellipsoid computation scheme follows these steps:

1) Assume C_0 is given, i.e we know the current ellipsoid enclosing $\bar{P}_0$. This ellipsoid can be initialized based on the sensor estimate or from user input.
2) Find $p_{1:n}$ in C_i and $\|w_i\| \leq \epsilon_i$ such that $x_i = f(p_i, u_i, w_i)$ minimizes (15).
3) Propagate the center of the ellipsoid μ_i using the closed loop dynamics to obtain μ_{i+1}.
4) Find the radii and principal axes of the ellipsoid using (5), (6).

The enclosing ellipsoid is completely defined by the starting points $p_{1:n}$, the noise terms $w_{1:n}$ and the center of the ellipsoid μ_i. Hence the ellipsoid propagation can be written as

$$C_{i+1} = g(p_{1:n}, w_{1:n}, \mu_i), \quad (16)$$

where the function g corresponds to the steps listed above. The starting points and noise terms are used to formulate an approximate ellipsoid propagation scheme later.

III. ROBUST OBSTACLE AVOIDANCE

The robust obstacle avoidance problem can be stated as finding a nominal trajectory $\{\bar{x}_{1:N}\}$ and the enclosing ellipsoids centered around the nominal trajectory $C_{1:N}$ such that the terminal state reaches some goal state x_d and the enclosing regions do not intersect with the obstacles $o_{1:P}$. A system is steered toward the nominal trajectory using a controller with feed-forward control inputs i.e. $u_i = \psi(x_i, \bar{x}_i, \bar{u}_i)$. The obstacle avoidance problem can be mathematically stated as

$$\min_{\bar{u}_{1:N}} \ L_f(\bar{x}_N, x_d) + \sum_{i=1}^{N} L_i(\bar{x}_i, \bar{u}_i)$$
$$\bar{x}_{i+1} = f(\bar{x}_i, u_i, 0), \quad (17)$$
$$C_{i+1} = f^+(C_i, \bar{u}_i, \bar{x}_i),$$
$$dist(C_{i+1}, O_j) \geq 0,$$

where f^+ propagates the disturbance invariant regions as described in §II and the $dist$ function denotes the closest distance between the disturbance invariant region and the jth obstacle. The distance constraint should be satisfied for all enclosing ellipsoids C_i and all the obstacles O_j. The cost function L_f minimizes the distance between the nominal terminal state $\bar{x}_N$ and the desired state x_d. The trajectory cost L_i minimizes the feed-forward controller inputs and the nominal trajectory velocities. Thus, by minimizing the cost function, we find a nominal trajectory that reaches the terminal state while minimizing the control effort along the trajectory. The inputs to the optimization problem are the feed-forward control inputs $\bar{u}_i$ to the controller ψ.

The optimization problem described in (17) steers the nominal trajectory to ensure the enclosing ellipsoids do not intersect the obstacles. This optimization consists of a two-level inner and outer optimization. The inner optimization propagates the enclosing ellipsoids given the nominal state and feed-forward control inputs as described in §II-B while the outer optimization ensures the nominal trajectory is such that it minimizes the cost specified in (17) and the enclosing ellipsoids do not intersect with the obstacles.

This optimization is not practical since propagating enclosing ellipsoids for every single nominal state and control input is computationally expensive. Based on experiments conducted, this algorithm works for double integrator dynamics but takes too long for the optimization to converge even for unicycle dynamics.

A. Approximate Ellipsoid Propagation

The ellipsoid propagation described in §II-B performs n optimizations to find the points $p_{1:n}$ and the noise terms $w_{1:n}$ that when propagated determine the ellipsoid radii and principal axes completely as specified in (16). These points and noise terms are a function of the starting ellipsoid and the feed-forward control inputs. In the approximate propagation

algorithm, we map the starting points from the input ellipsoid to a unit sphere as

$$e_j = diag(1/r_i)R_i^T(p_j - \mu_i), \text{ for } j \in \{1, \cdots, n\} \quad (18)$$

where $e_{1:n}$ are points inside a unit sphere and r_i, μ_i, R_i are the radius, center and principal axes of the input ellipsoid C_i. These mapped points are then assumed to be fixed even if the input ellipsoid and controller change. Therefore, given a new input ellipsoid $C_i' = \{r_i', \mu_i', R_i'\}$ and feed-forward control inputs $\bar{u}_i'$, the points on the unit sphere $e_{1:n}$ are projected back to the new input ellipsoid as

$$p_{1:n}' = \mu_i' + R_i' \, diag(r_i')(e_{1:n}). \quad (19)$$

The center of the ellipsoid at the next stage is found by propagating the center of the current ellipsoid using the feed-forward control inputs as

$$\mu_{i+1}' = f(\mu_i', \bar{u}_i', 0). \quad (20)$$

The radii and the principal axes of the ellipsoid are found by using the mapped points along with noise terms and the center of the ellipsoid in the ellipsoid propagation function specified in (16) as

$$C_{i+1} = g(p_{1:n}', w_{1:n}, \mu_{i+1}). \quad (21)$$

B. Sequential NMPC

This approach consists of running repeatedly two optimization steps. The first step optimizes the feed-forward control terms assuming the ellipsoid is propagated using the approximate ellipsoid propagation algorithm. Given points $e_{1:n}$ inside a unit sphere, the feed-forward control optimization can be written as

$$\begin{aligned}
&\min_{\bar{u}_{1:N}} \; L_f(\bar{x}_N, x_d) + \sum_{i=1}^{N} L_i(\bar{x}_i, \bar{u}_i) \\
&\bar{x}_{i+1} = f(\bar{x}_i, u_i, 0), \\
&p_{i,j} = \bar{x}_{i,j} + R_i diag(r_i) e_{i,j} \\
&C_{i+1} = g(p_{i,1:n}, w_{i,1:n}, \bar{x}_{i+1}), \\
&dist(C_{i+1}, o_j) \geq 0.
\end{aligned} \quad (22)$$

The approximate ellipsoid propagation is used to propagate the ellipsoids along the nominal trajectory assuming the input points used for propagation are fixed. This avoids a costly inner loop optimization while incurring an error due to not updating the propagation points. In practice, as the controller converges to the optimal value, there is no change in the controller, and therefore, the error incurred due to not updating the propagation points is negligible.

In the second optimization step, the points $p_{1:n}$ used for ellipsoid propagation are optimized using (15) for the updated controller and nominal trajectory. This updates the worst possible propagated points and the noise at those points for a given controller and nominal trajectory. The combined algorithm therefore successively tries to steer the nominal trajectory and improve the enclosing ellipsoid approximation. As both the optimization steps converge, we obtain an optimal

nominal trajectory with approximate disturbance invariant sets surrounding the trajectory. The pseudo-code for the sequential optimization scheme is summarized in (1)

Algorithm 1 Sequential NMPC for Robust Obstacle avoidance

Given C_0, x_0, x_d, L_i, L_f, max_iters
Start with an initial guess of $e_{1:n}$ on unit sphere
$i \leftarrow 0$
$iters \leftarrow 0$
for $iters <$ max_iters **do**
$\quad \min_{\bar{u}_{1:N}} \; L_f(\bar{x}_N, x_d) + \sum_{i=1}^{N} L_i(\bar{x}_i, \bar{u}_i)$,
$\quad$ assuming approximate ellipsoid propagation to update $C_{1:N}$.
$\quad$ **for** i in 1 to N **do**
$\quad\quad$ **for** j in 1 to n **do**
$\quad\quad\quad$ Given $\bar{u}_i$,
$\quad\quad\quad \min_{p_{i,j},w_{i,j}} (x_{jj} - r_{max})^2$,
$\quad\quad\quad$ s.t $x_j = f(p_{i,j}, \bar{u}_i, w_{i,j})$,
$\quad\quad\quad p_{i,j} \in C_i, \|w_{i,j}\| \leq \epsilon_i$.
$\quad\quad$ **end for**
$\quad\quad \mu_{i+1} \leftarrow f(\mu_i, \bar{u}_i, \mathbf{0})$.
$\quad\quad$ Use (5), (6) to find the enclosing ellipsoid $C_{i+1} = \{r_i, R_i\}$.
$\quad\quad$ Project: $e_{i,1:n} = diag(1/r_i)R_i^T(p_{i,1:n} - \mu_i)$.
$\quad$ **end for**
end for

---

1) Computational Complexity: The computational complexity of the NMPC algorithm is evaluated in terms of the number of calls to the closed loop dynamics of the system. The sequential NMPC method contains two steps of optimization. The first step is a regular NMPC step with the additional complexity of propagating the ellipsoids. The ellipsoid propagation requires n calls to the dynamics to propagate the input points from one ellipsoid to the next. Thus, to propagate ellipsoids for an entire trajectory requires $O(Nn)$ calls to the dynamics. The usual NMPC approach without encoding sparsity takes $O(N^2m)$ calls for one step of optimization, assuming the gradient (of dimension Nm) is computed using a finite-difference approximation. Thus, the first optimization step of the proposed sequential NMPC approach takes $O(N^2(m + mn)) = O(N^2mn)$ function calls to the dynamics to compute gradients and update the feed-forward controls and the nominal trajectory. Ellipsoid propagation creates a linear dependence on the state dimension.

The second step of the sequential NMPC performs n least squares minimizations along each step of the trajectory. Each of those least squares minimizations makes $2n$ function calls to update the corresponding input point and the noise at that point. Therefore, the second step of the optimization takes $O(Nn^2)$ calls to update the ellipsoids along the entire trajectory. Therefore, a single step of the entire sequential NMPC optimization takes $O(N^2mn + Nn^2)$ function calls. The additional burden of using the sequential NMPC is a term that is only linear in the number of trajectory steps and is quadratic in state dimension.

IV. COMPUTATIONAL RESULTS

We evaluate the robust obstacle avoidance procedure described in section III on two dynamical systems. The goal of the experiments is to show that the NMPC optimization can compute a feasible nominal trajectory that can avoid obstacles even when the system is subject to external disturbances. In addition, we show by sampling the disturbances and propagating several trajectories that most of the samples lie within the computed ellipsoids.

The first system is a dynamic unicycle model where the inputs to the model are the longitudinal acceleration a and the angular velocity ω of the vehicle. This system has a four dimensional state consisting of x and y positions, orientation θ and velocity v of the vehicle. The external disturbances $w \in \mathbb{R}^4$ are added into the model nonlinearly under the assumption that $\|w\| \leq 1$. The first two components of the noise correspond to the longitudinal and lateral noise respectively. The last two components of the noise correspond to the noise in angular velocity and acceleration. These components are scaled based on the velocity of the vehicle. The individual components of w are further scaled based on error magnitude $r_w \in \mathbb{R}^4$. The unicycle dynamics can be written as

$$\frac{d}{dt}\begin{bmatrix} x \\ y \\ \theta \\ v \end{bmatrix} = \begin{bmatrix} v\cos(\theta) \\ v\sin(\theta) \\ \omega \\ a \end{bmatrix} + \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & v & 0 \\ 0 & 0 & 0 & v \end{bmatrix} diag(r_w)w. \tag{23}$$

A feedback linearizing controller is used to track the nominal trajectory. The controller is designed only for the nominal dynamics assuming the disturbances are zero. Thus, the controller is not Lyapunov stable under the disturbances. The noise added to the dynamics is nonlinear in terms of the state, which further complicates the ellipsoid computation.

The NMPC optimization formulated in (22) is applied to this model where the nominal trajectory acceleration is optimized to steer around the obstacles. Figure 1 shows a trajectory computed using the NMPC optimization. The nominal trajectory goes around the obstacles and the enclosing ellipsoids do not intersect with the obstacles. The noise entering the system is nonlinear and is higher in the longitudinal direction than in the lateral direction. This causes the ellipsoids to align with the trajectory heading as the system progresses. This example demonstrates that the algorithm can effectively handle nonlinear noise models.

Figure 4 shows that the ellipsoid constraint (14) for sampled trajectories evaluated on the ellipsoids projected onto the xy plane. The ellipsoid constraint being less than zero implies that the trajectories lie within the ellipsoids computed. Among 1000 samples evaluated only one sample trajectory violated the ellipsoid constraint. This suggests that the computed ellipsoids are able to capture the effect of disturbances on the unicycle dynamics. Thus, by following the nominal trajectory using the feedback linearizing controller, the unicycle can avoid the obstacles even under external disturbances.

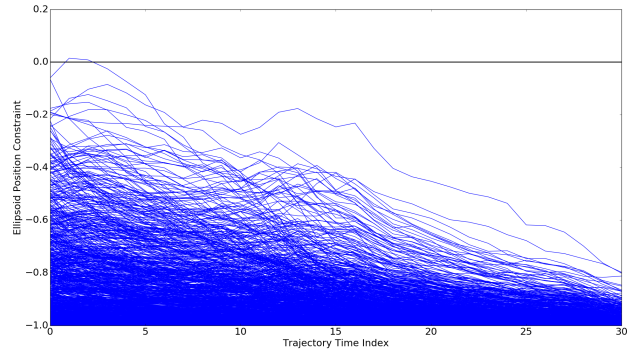The effect of the uncertainty on the planning process is

Fig. 4. Ellipsoid constraint for ellipsoids projected along the xy plane evaluated for 1000 sample trajectories with uniformly randomly sampled disturbances and initial state. The constraint is less than zero if the sampled trajectory is within the invariant set.

explained in Figure 5. In this experiment, the process noise is increased in successive steps keeping the obstacles and the controller gains the same. Further, the NMPC optimization has been initialized using the same initial feed-forward controls in all the cases. As the process noise increases, the NMPC chooses a trajectory that is farther away from obstacles which increases the trajectory cost. It also shows that the above NMPC optimization succeeds in finding a safe trajectory even under increased process noise conditions.
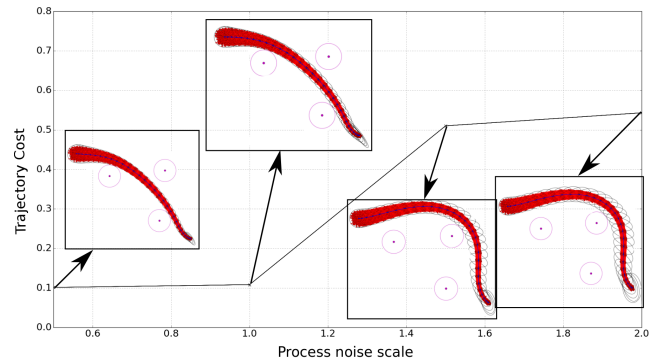

Fig. 5. The effect of the process noise on the trajectory planning for a fixed set of controller gains and obstacles. As the process noise increases, the NMPC still succeeds in reaching the goal by taking a more conservative trajectory.

The second system considered is a quadrotor aerial vehicle. A standard backstepping controller (such as the one described in [9]) is used for the nominal dynamics and while asymptotically stable in general, it loses strict stability guarantees under external disturbances. After performing a required *dynamic extension* of the model by adding the thrust and its time-derivative to the state, the state space becomes 14-dimensional. Figure 6 shows a nominal trajectory along with enclosing ellipsoids avoiding obstacles while reaching the goal. Figure 7 shows the ellipsoid constraint for 1000 sampled trajectories, where only two samples fall outside of the computed approximate invariant set. Figure 8 shows the 14 states along with the error bars along each axis that the

states are expected to stay within. The error bars on the states converge to a small value suggesting that the controller is converging to a small region around the nominal trajectory. This example shows that the NMPC optimization can readily extend to higher dimensional systems such as an agile aerial vehicle in 3D.
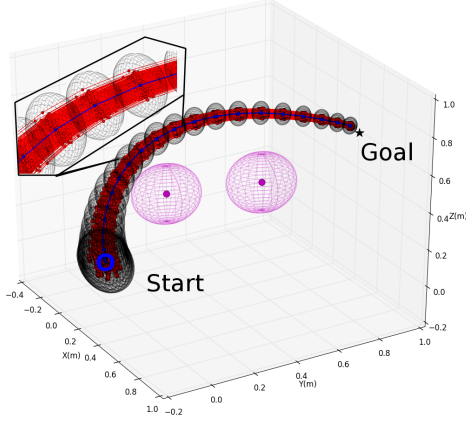

Fig. 6.   Quadrotor avoiding obstacles while reaching a goal. The nominal trajectory is blue while sample trajectories under the influence of disturbances are red. Obstacles are magenta, and the disturbance invariant sets are black ellipsoids surrounding the nominal trajectory. Note that the sample trajectories all fall within the computed approximate disturbance invariant sets.
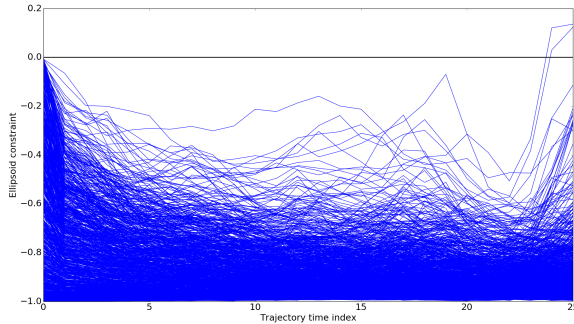

Fig. 7.   Ellipsoid constraint evaluated for 1000 sample trajectories with uniformly randomly sampled initial state and disturbances along the trajectory. The constraint is less than zero if the sampled trajectory is within the ellipsoid.

V. Conclusion

This work developed an algorithm for computing approximate disturbance invariant sets around a nominal trajectory for general nonlinear systems governed by nonlinear feedback controllers under external disturbances. We used the computed disturbance invariant sets to address the problem of robust obstacle avoidance by formulating a sequential NMPC optimization that optimizes over the nominal trajectory of the nonlinear dynamics while ensuring the approximate disturbance invariant sets do not intersect with the obstacles. Simulation results showed that the computed sets contain most of the samples and, therefore, are a good enough approximation to avoid obstacles under disturbances. We also demonstrated that the algorithm extends to nonlinear noise models in the case of
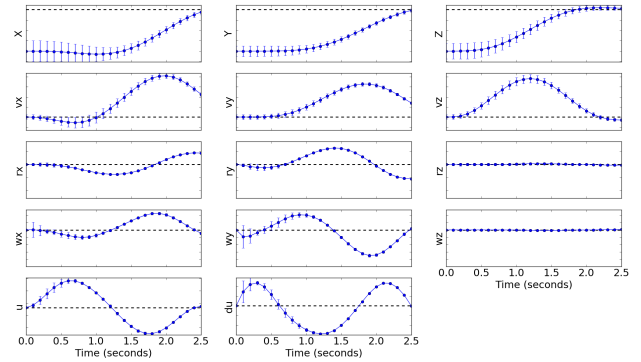

Fig. 8.   Quadrotor states along the nominal trajectory. Error bars denote the extent of the disturbance invariant set about the nominal trajectory. The black dashed line shows the desired state at the goal. The variables v, ω, r, u, and du are the velocity, angular velocity, RPY rotation, thrust, and thrust time derivative, respectively.

a unicycle model and higher dimensional systems such as a quadrotor system with a backstepping controller. Future work will focus on reducing the dilation between the true region of the propagated dynamics and the approximate enclosing region by evaluating different families of regions other than ellipsoids. Currently, the external disturbances are assumed to be within a bounded sphere. Future work will model the external disturbances using high confidence bounds computed from sample trajectories and will treat the obstacle avoidance constraints as a probabilistic constraints.

References

[1] Abraham Charnes and William W Cooper. Chance-constrained programming. *Management science*, 6(1): 73–79, 1959.

[2] Hong Chen, Carsten W Scherer, and F Allgower. A game theoretic approach to nonlinear robust receding horizon control of constrained systems. In *American Control Conference*, volume 5, pages 3073–3077. IEEE, 1997.

[3] Yin-Lam Chow and Marco Pavone. A framework for time-consistent, risk-averse model predictive control: Theory and algorithms. In *American Control Conference (ACC), 2014*, pages 4204–4211. IEEE, 2014.

[4] Benoît Colson, Patrice Marcotte, and Gilles Savard. An overview of bilevel optimization. *Annals of operations research*, 153(1):235–256, 2007.

[5] Vishnu R Desaraju, Alexander E Spitzer, and Nathan Michael. Experience-driven predictive control with robust constraint satisfaction under time-varying state uncertainty. In *Robotics: Science and Systems Conference (RSS), July*, 2017.

[6] Morteza Farrokhsiar and Homayoun Najjaran. Unscented model predictive control of chance constrained nonlinear systems. *Advanced Robotics*, 28(4):257–267, 2014.

[7] Yiqi Gao, Andrew Gray, H Eric Tseng, and Francesco Borrelli. A tube-based robust nonlinear predictive control approach to semiautonomous ground vehicles. *Vehicle System Dynamics*, 52(6):802–823, 2014.

[8] G. Garimella, J. Funke, C. Wang, and M. Kobilarov. Neural network modeling for steering control of an autonomous vehicle. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2609–2615, Sept 2017. doi: 10.1109/IROS.2017.8206084.

[9] Marin Kobilarov. Trajectory tracking of a class of underactuated systems with external disturbances. In *American Control Conference (ACC), 2013*, pages 1044–1049. IEEE, 2013.

[10] Pu Li, Harvey Arellano-Garcia, and Günter Wozny. Chance constrained programming approach to process optimization under uncertainty. *Computers & chemical engineering*, 32(1):25–45, 2008.

[11] Anirudha Majumdar and Russ Tedrake. Funnel libraries for real-time robust feedback motion planning. *The International Journal of Robotics Research*, 36(8):947–982, 2017.

[12] Ian R Manchester and Jean-Jacques E Slotine. Control contraction metrics and universal stabilizability. *IFAC Proceedings Volumes*, 47(3):8223–8228, 2014.

[13] Ian R Manchester and Jean-Jacques E Slotine. Output-feedback control of nonlinear systems using control contraction metrics and convex optimization. In *Australian Control Conference (AUCC)*, pages 215–220. IEEE, 2014.

[14] Zachary Manchester and Scott Kuindersma. Dirtrel: Robust trajectory optimization with ellipsoidal disturbances and lqr feedback. 2017.

[15] David Q Mayne, María M Seron, and SV Raković. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica*, 41(2): 219–224, 2005.

[16] S Norouzi Ghazbi, Y Aghli, M Alimohammadi, and AA Akbari. Quadrotors unmanned aerial vehicles: A review. *International Journal on Smart Sensing & Intelligent Systems*, 9(1), 2016.

[17] Robert Platt, Leslie Kaelbling, Tomas Lozano-Perez, and Russ Tedrake. Efficient planning in non-gaussian belief spaces and its application to robot grasping. In *Robotics Research*, pages 253–269. Springer, 2017.

[18] Stephen Prajna, Antonis Papachristodoulou, and Pablo A Parrilo. Introducing sostools: A general purpose sum of squares programming solver. In *IEEE Conference on Decision and Control (CDC)*, volume 1, pages 741–746. IEEE, 2002.

[19] Samuel Prentice and Nicholas Roy. The belief roadmap: Efficient planning in belief space by factoring the covariance. *The International Journal of Robotics Research*, 28 (11-12):1448–1465, 2009.

[20] Demetri Psaltis, Athanasios Sideris, and Alan A Yamamura. A multilayered neural network controller. *IEEE control systems magazine*, 8(2):17–21, 1988.

[21] Sasa V Rakovic, Eric C Kerrigan, Konstantinos I Kouramas, and David Q Mayne. Invariant approximations of the minimal robust positively invariant set. *IEEE Transactions on Automatic Control*, 50(3):406–410, 2005.

[22] SV Rakovic, P Grieder, Michal Kvasnica, DQ Mayne, and Manfred Morari. Computation of invariant sets for piecewise affine discrete time systems subject to bounded disturbances. In *IEEE Conference on Decision and Control (CDC)*, volume 2, pages 1418–1423. IEEE, 2004.

[23] SV Rakovic, AR Teel, DQ Mayne, and A Astolfi. Simple robust control invariant tubes for some classes of nonlinear discrete time systems. In *IEEE Conference on Decision and Control (CDC)*, pages 6397–6402. IEEE, 2006.

[24] Sumeet Singh, Anirudha Majumdar, Jean-Jacques Slotine, and Marco Pavone. Robust online motion planning via contraction theory and convex optimization. *IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[25] Jacob Steinhardt and Russ Tedrake. Finite-time regional verification of stochastic non-linear systems. *The International Journal of Robotics Research*, 31(7):901–923, 2012.

[26] Stefan Streif, Markus Kögel, Tobias Bäthge, and Rolf Findeisen. Robust nonlinear model predictive control with constraint satisfaction: A relaxation-based approach. *IFAC Proceedings Volumes*, 47(3):11073–11079, 2014.

[27] Mark M Tobenkin, Ian R Manchester, and Russ Tedrake. Invariant funnels around trajectories using sum-of-squares programming. *IFAC Proceedings Volumes*, 44 (1):9218–9223, 2011.

[28] Shuyou Yu, Christoph Maier, Hong Chen, and Frank Allgöwer. Tube mpc scheme based on robust control invariant set with application to lipschitz nonlinear systems. *Systems & Control Letters*, 62(2):194–200, 2013.