Joint synthesis of trajectory and controlled invariant funnel for discrete-time systems with locally Lipschitz nonlinearities

Taewan Kim | Purnanand Elango | Behçet Açıkmese

Abstract
This paper presents a joint synthesis algorithm of trajectory and controlled invariant funnel (CIF) for locally Lipschitz nonlinear systems subject to bounded disturbances. The CIF synthesis refers to a procedure of computing controlled invariance sets and corresponding feedback gains. In contrast to existing CIF synthesis methods that compute the CIF with a predefined nominal trajectory, our work aims to optimize the nominal trajectory and the CIF jointly to satisfy feasibility conditions without the relaxation of constraints and obtain a more cost-optimal nominal trajectory. The proposed work has a recursive scheme that mainly optimize trajectory update and funnel update. The trajectory update step optimizes the nominal trajectory while ensuring the feasibility of the CIF. Then, the funnel update step computes the funnel around the nominal trajectory so that the CIF guarantees an invariance property. As a result, with the optimized trajectory and CIF, any resulting trajectory propagated from an initial set by the control law with the computed feedback gain remains within the feasible region around the nominal trajectory under the presence of bounded disturbances. We validate the proposed method via two applications from robotics.

KEYWORDS
controlled invariant set, joint feedforward and feedback synthesis, nonlinear control, robust control, trajectory optimization

1 | INTRODUCTION

There has been significant amount of research on trajectory planning algorithms for nonlinear dynamics with non-convex constraints. A primary challenge in this area is handling uncertainty such as external disturbances, model mismatch in system dynamics, and state estimation error. Relying solely on a generated nominal trajectory, consisting of time-varying state and open-loop input signals, may not yield a robust control system. This is because the uncertainty can cause the system to deviate from the nominal trajectory. A potential solution is to synthesize not only the open-loop control but also feedback control law, which prevents the system from deviating too far from the nominal trajectory in the presence of uncertainties. To ensure safety by satisfying the prescribed constraints, it is necessary to compute a forward invariant or reachable set, also referred to as a controlled invariant funnel (CIF), that encapsulates all potential state trajectory under the uncertainty. Then, ensuring that the CIF remains inside the safety region can
The design of robust controllers for uncertain systems traditionally follows a two-step (or separate) scheme; initially, nominal trajectory planning is performed to compute the open-loop control and the corresponding state trajectory, followed by the synthesis of the feedback control and the associated CIF based on the analysis of the perturbed system around the nominal trajectory. In aerospace applications, the former is often termed to as guidance and the latter is referred to as control. However, such a two-step scheme has a potential drawback; the resulting control law, consisting of both the open-loop and closed-loop control, may be overly conservative. This conservatism stems from the lack of joint consideration of the open-loop and closed-loop control computations for given constraints such as actuator limits and obstacles. Consider, for example, a path planning scenario shown in Figure 1 where there are two obstacles between the start and end points. If the nominal trajectory is optimized independently of the CIF (Figure 1A), the resulting trajectory may be close to the obstacle boundary, so that the trajectory cost such as minimum-time or minimum-fuel is optimized. In such cases, the CIF can violate the constraints because the CIF size cannot be arbitrarily minimized under the presence of the external disturbances. One way to resolve issue is to introduce a safety margin in the constraints for the nominal trajectory planning. However, since it is not tractable to know the optimal margin beforehand, if the margin is too large, the resulting nominal trajectory and CIF can be conservative (Figure 1B).

To overcome this drawback, we propose an algorithm referred to as a joint synthesis, that performs the funnel synthesis jointly with the nominal trajectory optimization in an iterative scheme. The key advantage of the proposed approach for the joint synthesis of trajectory and CIF is its ability to reduce conservatism that can be caused with the aforementioned separate synthesis, and hence improve the optimality of the resulting control design without compromising robustness. The proposed joint synthesis algorithm does not need to estimate the safety margin of the given uncertainty in advance. Hence, this allows the algorithm to exploit a larger feasible set while designing the trajectory and the CIF jointly, thereby generating a more optimal trajectory and CIF (Figure 1A).

Specifically, the proposed method jointly generates trajectory and synthesizes funnel for locally Lipschitz nonlinear systems subject to bounded disturbances. The problem formulation can be viewed as a robust trajectory optimization in which we optimize both the trajectory and the CIF that consists of the forward invariant set and the corresponding feedback gain. To this end, we draw ideas from sequential convex programming (SCP), Lyapunov theory, and linear matrix inequalities (LMIs) for robust control. The proposed method has the following steps in each iteration: First, we update the nominal trajectory while ensuring the feasibility of the funnel. The next step estimates local Lipschitz constants of the nonlinearity in the system by sampling state space inside the funnel. With the trajectory computed in the first step, the third step then constructs a semi-definite programming (SDP) problem derived with funnel constraints and a Lyapunov condition to ensure the invariance property of the funnel. These steps are repeated until the convergence of both the trajectory and the funnel synthesis. We validate the proposed method through a numerical simulation.
1.1 Related work

In this subsection, we discuss related work with the comparison with the proposed work. Optimizing the trajectory and CIF jointly has been studied in the context of robust MPC. To reduce the computational complexity to satisfy the real-time performance, most works in robust MPC precompute the feedback gain or the forward reachable set and then optimize the nominal trajectory online. For example, a tube-based MPC scheme is developed for Lipschitz nonlinear systems subject to bounded disturbances in Reference 9. With the precomputed feedback gain and the tube set, the proposed method optimizes the nominal trajectory. The work in Reference 10 computes an incremental Lyapunov function and a corresponding feedback gain offline, and then optimizes the nominal trajectory and the support value of the invariant set online. In Reference 11, they obtain both the nominal trajectory and the invariant set by solving an SDP with the precomputed feedback gain. On the other hand, the proposed work separately parameterizes the nominal trajectory and the CIF that consists of the invariant state set and the feedback controller, and then optimizes them together in the recursive scheme. For nonlinear systems having incrementally conic uncertainties/nonlinearities, provides an LMI-based framework for the generation of the control policy and the invariant set for robust MPC. Our proposed work can be viewed as an extension of Reference 12 in order to handle more general nonlinear systems that are locally Lipschitz.

The work in Reference 13 proposes a robust tube-based MPC algorithm for obstacle avoidance. They provide a framework that optimizes the nominal trajectory and the corresponding funnel based on the derived ellipsoid uncertainty propagation. However, this propagation provides only an approximate guarantee, that is, the resulting funnel might not be invariant under the bounded disturbance, and they assume that the feedback controllers are a priori given. The proposed method, however, provides the exact invariance guarantee and optimizes the controller simultaneously, thereby reducing conservatism. For further examples of obstacle avoidance using MPC, we refer the reader to References 14–16. The most relevant results in robust MPC literature to the proposed work appear in References 17 and 18, which jointly synthesize the nominal trajectory and the funnel. The major difference compared to our method is that the resulting funnel in References 17 and 18 is only robust for the linearized closed-loop system (they ignored the higher-order terms due to nonlinearities). On the other hand, the proposed method considers the higher-order terms as state and input-dependent uncertainty using the local Lipschitz property. Thus, the robustness of the resulting funnel is not compromised. Another difference is that the proposed method sets the linear feedback gains and the invariant set variables as decision variables and then optimizes them simultaneously to satisfy the invariance and feasibility properties of the funnel. In contrast, the invariant set variables are determined by the ellipse propagation equation derived from the linearized closed-loop system in References 17. In Reference 18 the feedback gain is computed by solving the time-varying linear quadratic regulator (LQR) problem with the linearized closed-loop system. Again, in order to do that, both utilized the linearized dynamics.

The CIF computation in the proposed work is relevant to recent studies on finite horizon robustness analysis and robust synthesis. The finite horizon robust analysis for uncertain linear time-varying (LTV) systems is studied in Reference 19, where the behavior of the uncertain system is characterized by integral quadratic constraints (IQCs). This work is extended in Reference 6 where the robust controller is synthesized based on the established robustness analysis. Similar to References 19 and 6 we construct the CIF of the nonlinear systems by describing the nonlinearity with (incremental) quadratic inequality that can be viewed as a pointwise (incremental) IQC. The major difference is that the approach in References 6 and 19 focuses on analyzing and synthesizing the LTV systems rather nonlinear systems. To apply their approaches for the nonlinear systems, one needs obtain such LTV systems via linearization around a given nominal trajectory, but obtaining such nominal trajectories is not considered in their work. Meanwhile, the proposed work indeed considers this; we design an algorithm that optimizes the nominal trajectory in conjunction with synthesizing the uncertain LTV system obtained via linearization, computing both open-loop and feedback controllers, and the reachable set (funnel) for the nonlinear systems.

The proposed work is motivated by studies on the robust CIF generation. In Reference 5 sum-of-squares (SOS) programming is applied to design the CIF for nonlinear systems having polynomial dynamics subject to disturbances. They first design a finite library of open-loop nominal trajectories. Then, for each nominal trajectory, they optimize a feedback controller and a corresponding invariant set by solving SOS programming iteratively. Hence, their method is one shot procedure where the nominal trajectory is computed first, and then the computation of the funnel follows. The similar one shot approaches are conducted in References 20 and 21. For the fast computation of the CIF, the work in Reference 20 formulates an optimization problem for establishing the CIF as a linear program (LP) which is computationally cheaper than SOS programming. This research is extended to consider piecewise polynomial systems in Reference 21. These works, however, do not consider the controller synthesis, and hence focus on obtaining the reachable set (funnel) of the given (polynomial) closed-loop system. An LMI-based CIF computation for locally Lipschitz nonlinear systems is studied in
our previous work\(^2\) where the CIF computation is formulated as SDP that is also computationally efficient to solve than the problem with SOS programming, and the work is extended in Reference 3 to consider the bounded external disturbance. The aforementioned work including our previous work solely focuses on generating the CIF given the nominal trajectory, whereas the proposed work has a recursive scheme where the nominal trajectory, the feedback controller, and the invariant sets are optimized simultaneously.

Recently, a robust control method that jointly optimizes the nominal trajectory and the linear feedback gain was studied in Reference 22 Using a system-level synthesis framework, they formulated the problem of obtaining the nominal trajectory and the linear feedback gain for the LTV systems describing the error dynamics around the nominal trajectory. This work is extended to handle parametric uncertainties in Reference 23 The advantage of the proposed work is that it does not assume the disturbance to be additive, whereas in References 22 and 23 the bounded disturbances simply is added to the system dynamics. Hence, the proposed method can handle cases where the bounded disturbances manifest via a nonlinear relationship. We demonstrate this capability with the numerical simulation. Additionally, the approach in References 22 and 23 assumes system dynamics to be three times continuously differentiable, whereas the proposed work assumes one-time continuous differentiability.

1.2 Contributions

The contributions of the proposed work can be summarized as follows: First, we propose a novel algorithm that jointly synthesizes the nominal trajectory and the CIF to ensure robustness for nonlinear systems with locally Lipschitz nonlinearities. By jointly optimizing them together, the algorithm can mitigate the potential conservatism that may arise from optimizing them separately. Second, the proposed method can be applicable to a broad class of nonlinear systems that have locally Lipschitz nonlinearities (including systems with continuously differentiable dynamics) and nonadditive disturbances. Third, we extend the existing LMI-based CIF computation and robust MPC research\(^2,3,12\) in order to handle more general class of nonlinear systems that are discrete-time locally Lipschitz systems under the presence of norm-bounded uncertainties. Finally, we validate the proposed method through two distinct robotic applications: the first involves path planning with obstacle avoidance for a unicycle model, and the second focuses on a 6-degree-of-freedom (6-DoF) free-flying spacecraft.

1.3 Outline

We present the problem formulation in Section 2 and the proposed method in Section 3. In Section 4, we perform a numerical evaluation for our proposed method. Concluding remarks are provided in Section 5.

1.4 Notation

Let \(\mathbb{R}\) be the field of real numbers, \(\mathbb{R}^n\) be the \(n\)-dimensional Euclidean space, and \(\mathbb{N}\) be the set of natural numbers. A finite set of consecutive nonnegative integers is represented by \(\mathbb{N}^r := \{q, q + 1, \ldots, r\}\). The symmetric matrix \(Q = Q^\top(\succeq) > 0\) implies \(Q\) is positive-(semi)-definite matrix, and \((\mathbb{S}^n_+) \mathbb{S}^n_{++}\) denotes the set of all positive-(semi)-definite matrices whose size is \(n \times n\). The symbol \(\oplus\) denotes the Minkowski sum. The vector \((x, y)\) represents the concatenation of two vectors \(x\) and \(u\) into a longer vector. The notation * represents the symmetric part of a matrix, that is, \[
\begin{bmatrix}
a & b^\top \\
b & c
\end{bmatrix} = \begin{bmatrix}
a & * \\
b & c
\end{bmatrix},
\]
and \((\overline{x}_k, \overline{u}_k, \overline{w}_k)_{k=0}^K\) illustrates \((\overline{x}_0, \overline{u}_0, \overline{w}_0, \ldots, \overline{x}_K, \overline{u}_K, \overline{w}_K)\). The symmetric squared root of a symmetric matrix \(A\) is defined as \(A^{\frac{1}{2}}\) by eigenvalue decomposition.\(^{24}\) The operation \(\text{diag}(\cdot)\) is a diagonal matrix formed from its vector argument.

2 Problem Formulation

Consider a discrete-time uncertain nonlinear system of the following form:

\[
x_{k+1} = f(t_k, x_k, u_k, w_k), \quad \forall \ k \in \mathbb{N}_0^{\mathbb{N}-1},
\]
where \( N \in \mathbb{N} \) is the length of the time horizon and \( t_k \in \mathbb{R} \) is the time at the \( k \). The function \( f : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \times \mathbb{R}^{n_w} \rightarrow \mathbb{R}^{n_x} \) is assumed to be a locally Lipschitz and at least once differentiable. The vector \( x_k \in \mathbb{R}^{n_x} \) is the state, \( u_k \in \mathbb{R}^{n_u} \) is the control input, and the signal \( w_k \in \mathbb{R}^{n_w} \) is the exogenous disturbance or model mismatch that is assumed to be unknown but norm bounded: \( \|w_k\|_2 \leq 1 \) for all \( k \in \mathcal{N}_0^{N-1} \).

Let \( \{\tilde{x}_k\}_{k=0}^N \), \( \{\bar{u}_k, \bar{w}_k\}_{k=0}^{N-1} \) be a nominal trajectory that the CIF is centered around, and is feasible for the nonlinear dynamics (1). In this paper, the nominal trajectory is assumed to have zero disturbances, that is, \( \bar{w}_k = 0 \) for all \( k \in \mathcal{N}_0^{N-1} \). We define difference state \( \eta_k := x_k - \tilde{x}_k \) and difference input \( \xi_k := u_k - \bar{u}_k \), and assume a linear feedback \( \xi_k = K_k \eta_k \) for all \( k \in \mathcal{N}_0^{N-1} \), which leads to a closed-loop system and a control law given by

\[
\begin{align*}
\eta_{k+1} &= f(t_k, x_k, u_k, w_k) - f(t_k, \tilde{x}_k, \bar{u}_k, 0), \\
u_k &= \bar{u}_k + K_k \eta_k, \quad \forall k \in \mathcal{N}_0^{N-1},
\end{align*}
\]

where \( K_k \in \mathbb{R}^{n_x \times n_u} \) is a feedback gain. In this paper, we consider a specific class of funnels that consists of ellipsoids of state and input. The ellipsoid for the difference state is represented as

\[
\mathcal{E}_{Q_k} := \{\eta \in \mathbb{R}^{n_x} | \eta^T Q_k^{-1} \eta \leq 1\}, \quad \forall k \in \mathcal{N}_0^N,
\]

where \( Q_k \in \mathbb{S}^{n_x}_{++} \) is a positive definite matrix. With the linear feedback gain \( K_k \), it follows from Schur complement that \( \eta_k \in \mathcal{E}_{Q_k} \) implies \( \xi_k \in \mathcal{E}_{K_k Q_k K_k^T} \). Now we are ready to formally define the quadratic CIF.

**Definition 1.** A quadratic controlled positively invariant funnel, \( \mathcal{F}_k \), associated with a closed loop system (2) is a time-varying set in state and control space that is parameterized by a time-varying positive definite matrix \( Q_k \in \mathbb{S}^{n_x}_{++} \) and a time-varying matrix \( K_k \in \mathbb{R}^{n_x \times n_u} \) such that \( \mathcal{F}_k = \mathcal{E}_{Q_k} \times \mathcal{E}_{K_k Q_k K_k^T} \), and the funnel \( \mathcal{F}_k \) is invariant and feasible for all \( k \in \mathcal{N}_0^N \).

The invariance property of the CIF with the closed-loop system (2) and the control law (3) can be mathematically stated as follows:

\[
(\eta_0, \xi_0) \in \mathcal{F}_0 \Rightarrow (\eta_k, \xi_k) \in \mathcal{F}_k, \forall k \in \mathcal{N}_1^N.
\]

This condition implies that if a particular initial condition is inside the funnel, then a trajectory propagated with the closed-loop model (2) remains within the funnel as well. The feasibility property for the funnel \( \mathcal{F}_k \) can be mathematically expressed as:

\[
\begin{align*}
\{\tilde{x}_k\} \oplus \mathcal{E}_{Q_k} &\subseteq \mathcal{X}, \\
\{\bar{u}_k\} \oplus \mathcal{E}_{K_k Q_k K_k^T} &\subseteq \mathcal{U}, \quad \forall k \in \mathcal{N}_0^{N-1}.
\end{align*}
\]

The feasibility conditions require that every state and input in the funnel around the nominal trajectory should be feasible for the given state and input constraint sets \( \mathcal{X} \) and \( \mathcal{U} \), respectively.

Now we are ready to derive the problem formulation. The goal of the joint synthesis of trajectory and CIF is to solve a discrete-time nonconvex optimization problem of the following form:

\[
\begin{align*}
\text{minimize} & \quad \sum_{k=0}^{N-1} J_k(\tilde{x}_k, \bar{u}_k) + w_Q \sum_{k=0}^{N} \mu_k^Q + w_K \sum_{k=0}^{N-1} \mu_k^K \\
\text{subject to} & \quad \tilde{x}_{k+1} = f(t_k, \tilde{x}_k, \bar{u}_k, 0), \forall k \in \mathcal{N}_0^{N-1} \\
& \quad Q_k \preceq \mu_k^Q I, \forall k \in \mathcal{N}_0^N \\
& \quad K_k Q_k K_k^T \preceq \mu_k^K I, \forall k \in \mathcal{N}_0^{N-1} \\
& \quad \text{conditions (5) - (6)} , \\
& \quad \tilde{x}_0 \oplus \mathcal{E}_{Q_0} \supseteq \mathcal{X}_0, \\
& \quad \tilde{x}_N \oplus \mathcal{E}_{Q_N} \subseteq \mathcal{X}_N,
\end{align*}
\]

where the summands in the objective function consist of the trajectory cost and the funnel cost, and \( 0 < w_Q \in \mathbb{R} \) and \( 0 < w_K \in \mathbb{R} \) are user-defined weights. The function \( J_k \) is a cost for the trajectory and is assumed to be convex in \( \tilde{x}_k \) and \( \bar{u}_k \).
FIGURE 2  A block diagram of the control procedure.

The slack variables $\mu^Q_k \in \mathbb{R}$ and $\mu^K_k \in \mathbb{R}$ are introduced to minimize the diameter of the ellipsoidal sets $E_{Q_k}$ and $E_{K_kQ_kK_k^T}$ in the funnel by imposing the constraints (7c) and (7d). Minimizing the size of the funnel leads the effect of the propagated disturbances starting from the initial set to be minimized. While minimizing the cost, the formulation guarantees the invariance property in (5) and ensures the feasibility of the ellipsoids encapsulating the nominal states and inputs in (6). For boundary conditions, the initial and final ellipsoids, $\mathcal{X}_0$ and $\mathcal{X}_N$, are given as

$$\mathcal{X}_0 = \left\{ x \mid (x - x_i)^T Q^{-1}_i (x - x_i) \leq 1 \right\}, \quad (8a)$$

$$\mathcal{X}_N = \left\{ x \mid (x - x_f)^T Q^{-1}_f (x - x_f) \leq 1 \right\}, \quad (8b)$$

where $x_i \in \mathbb{R}^{n_x}$ is a nominal initial state, $Q_i \in S_{++}^{n_x}$ is a constant matrix defining the initial ellipsoidal set, $x_f \in \mathbb{R}^{n_x}$ is the nominal final state, and $Q_f \in S_{++}^{n_x}$ is a constant matrix defining the final ellipsoidal set. The computed funnel at $k = 0$ should include the initial set $\mathcal{X}_0$ to generate the trajectory from any state in the initial set. Also, the ellipsoid corresponding to the state in the funnel at $k = N$ should be a subset of $\mathcal{X}_N$ so that the resulting trajectory is guaranteed to terminate in $\mathcal{X}_N$.

It is worth mentioning that the system dynamics (7b) for the nominal trajectory has no disturbances ($w_k = 0$), but the invariance property is achieved with the closed-loop dynamics (2) and (3) in which the disturbances exist. Hence, any trajectory propagated for the uncertain nonlinear dynamics (1) with the control law (3) from any initial state in $\mathcal{X}_0$ remains within the feasible region under the presence of norm bounded uncertainties. The block diagram of the resulting control signal is illustrated in Figure 2.

3  ITERATIVE ROBUST TRAJECTORY OPTIMIZATION

In this section, we discuss the details of the proposed method to solve the robust trajectory optimization problem given in (7a)–(7f). The method tackles the problem by iteratively updating the nominal trajectory $\{\bar{x}_k\}_{k=0}^N$, $\{\bar{u}_k\}_{k=0}^{N-1}$, the parameters of the set $\{Q_k\}_{k=0}^N$ and the feedback controller $\{K_k\}_{k=0}^N$ in the CIF. In each iteration, the method consists of three steps: the nominal trajectory update, the estimation of the locally Lipschitz constant, and the funnel update. In this section, we denote an initial guess or solution variables of the previous iteration (i.e., reference trajectory and funnel parameters) by $\{\bar{x}_k, \bar{Q}_k\}_{k=0}^N$, $\{\bar{u}_k, \bar{K}_k\}_{k=0}^{N-1}$. The block diagram of the proposed algorithms is given in Figure 3.

3.1  Nominal trajectory update

We require the nominal trajectory to satisfy the (possibly nonconvex) constraints (7b) and (6) while minimizing the trajectory cost $J_t$ by approximating the original problem with a convex subproblem. This is a typical process in many SCP methods to solve nonconvex trajectory optimization problems. In contrast to the typical SCP methods, the feasibility problem in (6) involves the funnel parameters that are fixed as the reference funnel variables $\{\bar{Q}_k, \bar{K}_k\}_{k=0}^{N-1}$ in this trajectory update step.

In each subproblem, the intermediate trajectory solution should satisfy the following affine system:

$$x_{k+1} = \bar{A}_k x_k + \bar{B}_k u_k + \bar{c}_k + v_k, \quad \forall k \in \mathcal{N}_0^{N-1}, \quad (9)$$
where \( \bar{A}_k, \bar{B}_k, \bar{Z}_k \) define the linearized model of the nonlinear dynamics given in (1) evaluated around the reference trajectory \( \{\hat{x}_k, \hat{u}_k\}_{k=0}^{N-1} \) with zero disturbance \( \bar{w}_k = 0 \). The term \( v_k \) is a virtual control variable that serves to prevent the sub-problem from being artificially infeasible due to linearization of dynamics and constraints.

The feasible sets \( \mathcal{X} \) and \( \mathcal{U} \) are expressed as

\[
\mathcal{X} = \{x|h_i(x) \leq 0, \ i = 1, \ldots, m_x\},
\]

\[
\mathcal{U} = \{u|g_j(u) \leq 0, \ j = 1, \ldots, m_u\},
\]

where \( h_i \) and \( g_j \) are at least once differentiable functions. While we assume here that \( \mathcal{X} \) and \( \mathcal{U} \) are time-invariant for brevity, the proposed framework, however, can easily incorporate time-varying sets. The nonlinear constraints need to be linearized to ensure convexity of the subproblem. Thus, we approximate the feasible set \( \mathcal{X} \) and \( \mathcal{U} \) as polytopes, which are obtained via linearization around \( \{\hat{x}_k, \hat{u}_k\}_{k=0}^{N-1} \) as follows:

\[
P^x_k = \{x|(a^*_i)^{\top}x_k \leq (b^*_i)_k, \ i = 1, \ldots, m_x\},
\]

\[
P^u_k = \{u|(a^{\mu}_j)^{\top}u_k \leq (b^{\mu}_j)_k, \ j = 1, \ldots, m_u\},
\]

where \((a^*_i, b^*_i)\) and \((a^{\mu}_j, b^{\mu}_j)\) are first-order approximations of \( h_i \) and \( g_j \), respectively. Notice that while \( \mathcal{X} \) and \( \mathcal{U} \) are assumed to be time-invariant, their polytopic approximations \( P^x_k \) and \( P^u_k \) would be time-varying due to the time variation in the reference trajectory \( \hat{x}_k, \hat{y}_k \). Then, the feasibility conditions with the fixed funnel parameters \( \{\hat{Q}_k, \hat{K}_k\}_{k=0}^{N-1} \) in (6) can be approximated as linear constraints as follows:

\[
\begin{align}
\| \hat{Q}_k^{\top} (a^*_i)^{\top}x_k \|_2 + (a^*_i)^{\top}x_k &\leq (b^*_i)_k, i = 1, \ldots, m_x, \quad (10a) \\
\| \hat{K}_k \hat{Q}_k^{\top} (a^{\mu}_j)^{\top}u_k \|_2 + (a^{\mu}_j)^{\top}u_k &\leq (b^{\mu}_j)_k, j = 1, \ldots, m_u, \quad \forall k \in \mathbb{N}_0^{N-1}. \quad (10b)
\end{align}
\]

The trajectory update step for the nominal trajectory has the following form of a second-order cone program:

\[
\begin{align}
\text{minimize} & \sum_{k=0}^{N-1} J_t(\bar{x}_k, \bar{u}_k) + J_{vc}(\bar{v}_k) + J_T(\bar{x}_k, \bar{u}_k). \\
\text{subject to conditions} & (9) - (10), \\
\bar{x}_0 & = x_i, \quad \bar{x}_N = x_f. 
\end{align}
\]
the vicinity of the reference trajectory \( \{ \dot{x}_k, \dot{u}_k \}_{k=0}^{N-1} \) where the linearization error is small. They are formulated as follows:

\[
\begin{align*}
J_{w_c}(v_k) &= w_c \| v_k \|_1, \\
J_{p}(x_k, u_k) &= w_p (\| x_k - \dot{x}_k \|_2^2 + \| u_k - \dot{u}_k \|_2^2),
\end{align*}
\]

(12a, 12b)

where \( w_c \in \mathbb{R} \) and \( w_p \in \mathbb{R} \) are user-defined weight parameters for the virtual control and the trust region, respectively. As a result of the optimization problem (11), the solution becomes a new nominal trajectory \( \{ \bar{x}_k \}_{k=0}^{N} \), \( \{ \bar{u}_k \}_{k=0}^{N-1} \) that will be used for the funnel computation in the following section. This type of penalized-trust region-based optimization has been studied for trajectory optimization\(^{25}\) and general nonlinear programming.\(^{26}\)

### 3.2 CIF update

In this section, we describe how to optimize the CIF around the nominal trajectory obtained from the previous section. The optimization problem derived in this section aims to make the funnel invariant \((5)\) and feasible for the constraints \((6)\) and the boundary conditions \((7f)\) for locally Lipschitz nonlinear systems. To this end, we construct a SDP whose solution provides the parameters of the invariant set and the feedback gains \( \{ Q_k \}_{k=0}^{N} \), \( \{ K_k \}_{k=0}^{N-1} \).

#### 3.2.1 Nonlinear dynamics

Since the nonlinear dynamics in \((1)\) is at least once differentiable, it can be rewritten as

\[
x_{k+1} = f(t_k, x_k, u_k, w_k),
\]

(13a)

\[
= A_k x_k + B_k u_k + F_k w_k + E p_k,
\]

(13b)

\[
p_k = \phi_k(q_k),
\]

(13c)

\[
q_k = C x_k + D u_k + G w_k.
\]

(13d)

Notice that all nonlinearities are lumped into a vector \( p_k \in \mathbb{R}^{n_p} \) represented by a function \( \phi_k : \mathbb{R}^{n_q} \rightarrow \mathbb{R}^{n_p} \) with its argument \( q_k \in \mathbb{R}^{n_q} \). The matrix \( E \in \mathbb{R}^{n_x \times n_p} \) is introduced since not all states are affected by the nonlinearities. The matrices \( A_k, B_k, \) and \( F_k \) can be arbitrary, but we specifically choose \( A_k, B_k, \) and \( F_k \) to be the first-order approximation of the nonlinear dynamics \( f \) around the nominal trajectory as follows:

\[
A_k := \frac{\partial f}{\partial x} \bigg|_{x = \bar{x}_k, u = \bar{u}_k, w = 0}, \quad B_k := \frac{\partial f}{\partial u} \bigg|_{x = \bar{x}_k, u = \bar{u}_k, w = 0},
\]

\[
F_k := \frac{\partial f}{\partial w} \bigg|_{x = \bar{x}_k, u = \bar{u}_k, w = 0}, \quad \forall k \in \mathcal{N}_0^{N-1}.
\]

With difference state \( \eta_k \) and input \( \zeta_k \), the difference dynamics can be derived as

\[
x_{k+1} - \bar{x}_{k+1} = A_k \eta_k + B_k \zeta_k + F_k w_k + E (p_k - \bar{p}_k) + f(t_k, \bar{x}_k, \bar{u}_k, 0) - \bar{x}_{k+1},
\]

where \( \bar{p}_k = \phi_k(\bar{q}_k) \) and \( \bar{q}_k = C \bar{x}_k + D \bar{u}_k. \) The term \( f(t_k, \bar{x}_k, \bar{u}_k, 0) - \bar{x}_{k+1} \) on the right-hand side exists because of the dynamical error in the intermediate nominal trajectory \( \{ \bar{x}_k \}_{k=0}^{N}, \{ \bar{u}_k \}_{k=0}^{N-1} \). This error is gradually reduced as the iteration proceeds because the trajectory update \((11)\) ensures that the nominal trajectory becomes dynamically feasible for the entire interval. Thus, we intentionally do not consider this error in the funnel update step since it is sufficient for the funnel to satisfy the invariance and feasibility properties with the converged nominal trajectory that is dynamically feasible. The difference dynamics we consider for the funnel update is consequently written as

\[
\eta_{k+1} = A_k \eta_k + B_k \zeta_k + F_k w_k + E \delta p_k,
\]

\[
\delta p_k = \phi_k(q_k) - \phi_k(\bar{q}_k),
\]

\[
\delta q_k = C \eta_k + D \zeta_k + G w_k.
\]
where $\delta p_k := p_k - \bar{p}_k$ and $\delta q_k := q_k - \bar{q}_k$. With the linear feedback controller $\xi_k = K_k \eta_k$, the inclusion $\eta_k \in E_{Q_k}$ implies that $q_k$ is in a compact set $\mathcal{Q}$ that is given as

$$
\delta Q_k = E_{C^t Q_k} \oplus \{ F_k w_k \| w_k \|_2 \leq 1 \},
$$

$$
Q_k = \{ \bar{q}_k \} \oplus \delta Q_k, \quad \forall \ k \in \mathcal{N}_0^{N-1},
$$

where $C^t_k := C + DK_k$. The assumption that the function $f$ is locally Lipschitz implies that the function $\phi_k$ is locally Lipschitz as well. Thus, for the compact (closed and bounded) set $Q_k$, there exists a $\gamma_k$ such that

$$
\| \phi_k(q_k) - \phi_k(\bar{q}_k) \|_2 \leq \gamma_k \| q_k - \bar{q}_k \|_2,
$$

$$
\forall \ q_k \in Q_k, \forall \ k \in \mathcal{N}_0^{N-1}.
$$

Considering them together, the closed-loop system becomes

$$
\eta_{k+1} = A^d_k \eta_k + F_k w_k + E \delta p_k,
$$

$$
\delta q_k = C^d_k \eta_k + G w_k,
$$

$$
\| \delta p_k \|_2 \leq \gamma_k \| \delta q_k \|_2,
$$

$$
\| w_k \|_2 \leq 1,
$$

$$
\delta q_k \in \delta Q, \quad \forall \ k \in \mathcal{N}_0^{N-1},
$$

where $A^d_k := A_k + B_k K_k$.

### 3.2.2 Invariance of a quadratic funnel

Consider a scalar-valued quadratic Lyapunov function $V$ defined by

$$
V(k, \eta) = \eta^T_k Q_k^{-1} \eta_k.
$$

For the closed-loop system model (14), we aim to design $\{ Q_k \}_{k=0}^N$, $\{ K_k \}_{k=0}^{N-1}$ that satisfies the following quadratic stability condition:

$$
V(k + 1, \eta_{k+1}) \leq \alpha V(k, \eta_k),
$$

$$
\forall \ \| \delta p_k \|_2 \leq \gamma_k \| \delta q_k \|_2,
$$

$$
\forall \ V(k, \eta_k) \geq \| w_k \|_2^2,
$$

$$
\forall \ k \in \mathcal{N}_0^{N-1},
$$

where $0 < \alpha < 1$. The above condition ensures the quadratic stability (16a) whenever the locally Lipschitz property of the nonlinearity $\phi_k$ expressed in (16b) holds. The condition (16c) exists to obtain the invariance property of the funnel under the presence of the bounded disturbance $w_k$. In the rest of this subsection, we construct a condition that implies the stability condition (16). In the following corollary, we also show that the derived LMI condition ensures the invariance property of the funnel.

**Theorem 1.** Suppose that there exists $Q_k \in \mathbb{R}^{n_n \times n_n}$, $Y_k \in \mathbb{R}^{n_x \times n_n}$, $\nu^p_k > 0$, $\lambda^w_k > 0$, and $0 < \alpha < 1$ such that $\lambda^w_k < \alpha$ and the following matrix inequality holds for all $k \in \mathcal{N}_0^{N-1}$:

$$
\begin{bmatrix}
\alpha Q_k - \lambda^w_k Q_k & * & * & * & * \\
0 & \nu^p_k I & * & * & * \\
0 & 0 & \lambda^w_k I & * & * \\
A_k Q_k + B_k Y_k & \nu^p_k E_k & F_k & Q_{k+1} & * \\
C_k Q_k + D_k Y_k & 0 & G_k & 0 & \nu^p_k \frac{1}{\gamma^2_k} I
\end{bmatrix} \succeq 0.
$$

Then the Lyapunov condition (16) holds for the closed loop system (14) with $K_k = Y_k Q_k^{-1}$. 


Proof. With the closed-loop system (14), the condition (16) holds if there exists a $\lambda^p_k > 0$, $\lambda^w_k > 0$, and $0 < \alpha < 1$ such that the following inequality holds by S-procedure\textsuperscript{27} for all $\eta_k \in \mathbb{R}^{n_\eta}$, $w_k \in \mathbb{R}^{n_w}$, $\delta p \in \mathbb{R}^{n_p}$:

$$V(k + 1, \eta_{k+1}) - aV(k, \eta_k) + \lambda^w_k(V(k, \eta_k) - \|w_k\|_2^2) + \lambda^p_k(\nu_k^2\|\delta q_k\|_2^2 - \|\delta p_k\|_2^2) \leq 0. \quad (18)$$

This is equivalent to

$$\begin{bmatrix} A_k^c E_k & F_k \end{bmatrix}^T Q_{k+1}^{-1} \begin{bmatrix} A_k^c E_k & F_k \end{bmatrix} + \lambda^w_k \begin{bmatrix} C_k^c & 0 & G_k \end{bmatrix}^T \begin{bmatrix} \gamma_k^2 I & 0 \end{bmatrix} \begin{bmatrix} C_k^c & 0 & G_k \end{bmatrix} \preceq 0.$$  

With the appropriate re-arrangement and applying Schur complement, we obtain

$$\begin{bmatrix} H_k^1 & * & * & * & * \\ 0 & \lambda^p_k I & * & * & * \\ 0 & 0 & \lambda^w_k I & * & * \\ Q_{k+1}^{-1} A_k^c & Q_{k+1}^{-1} E_k & Q_{k+1}^{-1} F_k & Q_{k+1}^{-1} & * \\ C_k^c & 0 & G_k & 0 & H_k^2 \end{bmatrix} \succeq 0,$$

where $H_k^1$ and $H_k^2$ are given by

$$H_k^1 = aQ_k^{-1} - \lambda^w_k Q_k^{-1},$$

$$H_k^2 = (\lambda^p_k)^{-1} \gamma_k^2 I.$$

Multiplying both sides by $\text{diag}(Q_k, \lambda_k^{-1} I, Q_{k+1}, I)$ yields

$$\begin{bmatrix} aQ_k - \lambda^w_k Q_k & * & * & * & * \\ 0 & \nu_k^2 I & * & * & * \\ 0 & 0 & \nu_k^2 I & * & * \\ A_k^c Q_k & \nu_k^2 E_k & F_k & Q_{k+1} & * \\ C_k^c Q_k & 0 & G_k & 0 & \nu_k^2 \gamma_k^{-1} I \end{bmatrix} \succeq 0,$$

where $\nu_k^2 = (\lambda^p_k)^{-1}$. Finally, expanding $A_k^c$ and $C_k^c$ completes the proof. $\blacksquare$

Corollary 1. The condition (17) in Theorem 1 implies the following invariance condition for all $k \in \mathcal{N}_0^{N-1}$:

$$V(k + 1, \eta_{k+1}) \leq 1, \quad (19a)$$

$$\forall V(k, \eta_k) \leq 1, \quad (19b)$$

$$\forall \|\delta p_k\|_2 \leq \gamma_k \|\delta q_k\|_2, \quad (19c)$$

$$\forall \|w_k\|_2 \leq 1. \quad (19d)$$

Proof. Observe that (18) can be equivalently written as

$$V(k + 1, \eta_{k+1}) - a + (a - \lambda^w_k(1 - V(k, \eta_k)) + \lambda^w_k(1 - \|w_k\|_2^2) + \lambda^p_k(\nu_k^2\|\delta q_k\|_2^2 - \|\delta p_k\|_2^2) \leq 0.$$
This implies $V(k + 1, \eta_{k+1}) \leq \alpha$ with (19b)-(19d) since $0 < \lambda_k \leq \alpha$ and $\lambda_k > 0$. This is sufficient for the invariance condition (19) since $\alpha < 1$.

Notice that the matrix inequality (17) is a LMI once $\alpha$ and $\lambda_k$ are fixed.

### 3.2.3 Computing the funnel via SDP

The goal of computing the CIF is to bound the effects of disturbances going forward in time by minimizing the size of the funnel while satisfying the invariance and the feasibility of the boundary conditions. To this end, the funnel computation is posed as the following SDP:

\[
\begin{align*}
\text{minimize} & \quad W_0 \sum_{k=0}^{N-1} \mu_k^Q + W_k \sum_{k=0}^{N-1} \mu_k^P + \sum_{k=0}^{N-1} J_{tr}(Q_k, Y_k), \\
\text{subject to} & \quad Q_k \leq \mu_k^Q, \forall k \in \mathcal{N}_0^N, \\
& \quad \mu_k^P Y_k = 0, \forall k \in \mathcal{N}_0^{N-1}, \\
& \quad \text{condition (17)}, \\
& \quad \left( (b_i^p - (a_i^u_k) \bar{b}_k)^2 \begin{bmatrix} (a_i^u_k)^T Q_k \\ Q_k \end{bmatrix} \right) \geq 0, i = 1, \ldots, m_u, \\
& \quad \left( (b_j^m - (a_j^m_k) \bar{u}_k)^2 \begin{bmatrix} (a_j^m_k)^T Y_k \\ Y_k \end{bmatrix} \right) \geq 0, j = 1, \ldots, m_u, \\
& \quad Q_0 \preceq Q_1, \quad Q_N \preceq Q_f, \\
\end{align*}
\]

where (20c) is equivalent to (7d) which can be derived by Schur complement with $Y_k = K_k Q_k$. The LMI constraints in (20e) and (20f) are the funnel feasibility conditions that are equivalent to (10) that can be derived by Schur complement. The cost $J_{tr}$ is given as

\[J_{tr} = W_{\text{eff}} \sum_{k=0}^{N-1} \left( \|Q_k - \hat{Q}_k\|_F^2 + \|Y_k - \hat{Y}_k\|_F^2 \right),\]

where $W_{\text{eff}} \in \mathbb{R}$ is a user-defined parameter, $\| \cdot \|_F$ is the Frobenius norm, and $\hat{Y}_k = \hat{K}_k \hat{Q}_k$ for all $k \in \mathcal{N}_0^{N-1}$. This cost, similar to the trust region penalty $J_{tr}$, penalizes the difference between the current solution $\{Q_k, Y_k\}_{k=0}^{N-1}$ and the previous solution $\{\hat{Q}_k, \hat{Y}_k\}_{k=0}^{N-1}$ which is beneficial for the better convergence performance.

The choice of parameters in the proposed method affects the performance of the control law in (3). The weights $W_0$ and $W_k$ in (7a) balances the size of the state funnel $E_{Q_k}$ and input funnel $E_{K_k Q_k K_k^T}$. For example, a relatively larger $W_0$ compared to $W_k$ drives the algorithm to put more effort on minimizing the size of the state funnel over the input funnel, and vice versa. The choices of the decay rate $\alpha$ and the slack variable $\lambda_k$ resulted from $S$-procedure in (18) also affects the control performance. As the decay rate decreases, the controller places greater emphasis on faster convergence to the nominal trajectory due to the condition outlined in (7a). Likewise, the larger $\lambda_k$ places more emphasis on the convergence to the nominal trajectory. This is attributed to the term $V(k + 1, \eta_{k+1}) - \alpha V(k, \eta_k)$ in (19) becoming smaller (more negative) as $\lambda_k (V(k, \eta_k) - \|w_k\|_2^2)$ increases.

### 3.2.4 Local Lipschitz constant estimation via sampling

To compute the LMI (17), the Lipschitz constant $\gamma_k$ in (16) should be available. We estimate the Lipschitz constant by employing a sampling method. It is worth mentioning that the sampling method for the estimation of the Lipschitz constant $\gamma_k$ brings about an algebraic loop: to estimate the Lipschitz constant $\gamma_k$, the funnel variables $Q_k$ and $K_k$ should be
available, whereas the computation $Q_k$ and $K_k$ in (17) requires the constant $\gamma_k$. However, a well-behaved iterative scheme with the sampling method for $\gamma_k$ can make the funnel computation converge.\footnote{4168}

By sampling a set of $N_i$ pairs of state and disturbance $(\eta^s_k, w^s_k)_{s=1}^{N_i}$ from the ellipsoid $\mathcal{E}_Q$ and the set $\{w \in \mathbb{R}^{n_w}||w||_2 \leq 1\}$, respectively, we compute

$$\delta^s_k = \frac{||p^s_k - \bar{p}_k||}{||q^s_k - \bar{q}_k||}, \quad s = 1, \ldots, N_s,$$  \hfill (21)

where $p^s_k$ and $q^s_k$ are computed by (13). Depending on the discretization method, only $Ep_k$ might be available instead of $p_k$. So, it might not be possible to compute (21). In that case, we instead solve the following optimization to obtain the value $\delta^s_k$:

$$\delta_k^s = \text{minimize}_\Delta ||\Delta||_2,$$  \hfill (22a)

subject to $\eta^s_k + A_{\Delta}^{2d} \eta^s_k - F_k w^s_k + \bar{x}_{k+1} - f(\bar{x}_k, \bar{u}_k, 0) = E \Delta (C^{cl}_k \eta^s_k + G_k w^s_k)$,

$$\text{where } \Delta \in \mathbb{R}^{n_x \times n_y}. \text{ After obtaining } \delta_k^s \text{ by (21) or (22), the following maximization operation is performed to estimate the local Lipschitz constant:}$$

$$\gamma_k = \max_{s=1, \ldots, N_s} \delta_k^s, \quad \forall \ k \in \mathcal{N}_0^{N-1}.$$  \hfill (23)

It is worth noting that the disadvantage of the illustrated sampling-based method is that the computed $\gamma_k$ might be lower than the true level of nonlinearity. To handle this issue, one may able to use a probabilistic approach for overestimating the local Lipschitz constant from samples provided in Reference 28.

Another way to estimate the local Lipschitz constant is to use the optimization-based approach provided in Sec. 6.5.1 of Reference 29. To illustrate, we consider

$$\Gamma_k^* = \max_{\eta_k, w_k} \frac{1}{2} \delta_k^s(\eta_k, w_k)^2$$

subject to $\eta_k Q_k^{-1} \eta_k \leq 1$,

$$\|w_k\|_2 \leq 1,$$  \hfill (24a)

where

$$\delta_k^s(\eta_k, w_k) = \text{minimize}_\Delta ||\Delta||_2$$

subject to $\eta_{k+1} - A_{\Delta}^{cl} \eta_k - F_k w_k + \bar{x}_{k+1} - f(\bar{x}_k, \bar{u}_k, 0) = E \Delta (C^{cl}_k \eta_k + G_k w_k)$.

(25b)

The inner optimization (25) aims to find the smallest matrix $\Delta$ in terms of the matrix 2-norm for the given $\eta_k, w_k$, and the outer optimization (24) finds the values of $\eta_k, w_k$ that maximize $\delta_k^s$. After solving these optimization problems for each $k$, the local Lipschitz constant can be obtained by computing

$$\gamma_k = \sqrt{2 \Gamma_k^*}.$$  \hfill (26)

To make the outer optimization computationally tractable, one could potentially utilize an analytic upper bound for the problem’s optimal value. The constraint (25b) can be rewritten as

$$y(\eta_k, w_k) = E \Delta (C^{cl}_k \eta_k + G_k w_k),$$

$$y = \left[ E(e_1^T (C^{cl}_k \eta_k + G_k w_k)) \cdots E(e_{n_q}^T (C^{cl}_k \eta_k + G_k w_k)) \right] \bar{\Delta}_k,$$  \hfill (26b)

where $y(\eta_k, w_k) = \eta_{k+1} - A_{\Delta}^{cl} \eta_k - F_k w_k + \bar{x}_{k+1} - f(\bar{x}_k, \bar{u}_k, 0)$ and $\bar{\Delta}_k \in \mathbb{R}^{n_x \times n_q}$ is a concatenated vector that stacks the columns of $\Delta_k$. Then, consider the following optimization:

$$\delta_k^*(\eta_k, w_k) = \text{minimize}_\bar{\Delta}_k \|\bar{\Delta}_k\|_2 \text{ subject to } y(\eta_k, w_k) = H(\eta_k, w_k) \bar{\Delta}_k.$$  \hfill (27)
Since (27) is a minimum-norm least squares problem, the solution is \( \Delta_k = H'(\eta_k, w_k)\gamma(\eta_k, w_k) \) where \( H' \) represents the pseudo-inverse of matrix \( H \). The optimal value \( \delta_k^* \) of (27) is the upper bound of the inner optimization (25) as \( \|\Delta\|_2 \leq \|\Delta\|_F = \|\Delta_k\|_2 \) where \( \| \cdot \|_F \) is the Frobenius norm. Then, the outer optimization (24) can be transformed into

\[
\Gamma^*_k = \max_{\eta_k, w_k} \frac{1}{2} y(\eta_k, w_k)^T H'(\eta_k, w_k) H'(\eta_k, w_k) y(\eta_k, w_k),
\]

subject to \( \eta_k Q_k^{-1} \eta_k \leq 1, \quad \|w_k\|_2 \leq 1, \tag{28a-c} \)

More details in the derivation and the computation results can be found Sec. 6.5.1 in Reference 29.

### 3.3 Algorithm details and summary

To start the algorithm, we need to generate an initial guess hat is used as a reference trajectory for the first iteration. It is worth noting that the initial guess does not need to be feasible to constraints for the proposed method. The first way is to employ a straight-linear interpolation for the initial nominal trajectory \( \{x_k\} \), \( \{u_k\} \) Then the feedback gain \( \{K_k\} \) can be obtained by solving a discrete-time LQR problem with a linearized model of (1) evaluated around the nominal trajectory. The initial guess for the ellipsoid variable \( \{Q_k\} \) can then be set to a diagonal matrix having user-defined diameters. The second way to generate the initial guess is to use the separate synthesis; the nominal trajectory is generated by the SCP algorithm without considering the funnel. This provides the dynamically-feasible trajectory that can be used as the initial guess of the nominal trajectory \( \{x_k\} \), \( \{u_k\} \) for the proposed method. Then, the feedback gain \( \{K_k\} \) and the ellipsoid variable \( \{Q_k\} \) can be obtained via solving (20) with \( w_{trf} = 0 \) while ignoring the funnel feasibility. The second way is more systematical since it exploits the result of the separate synthesis and hence gives a better initial guess compared to the solution computed by the straight-line interpolation in the first way.

To set the stopping criteria, we define \( \Delta_{ic}, \Delta_{dyn}, \Delta_T, \) and \( \Delta_F \) as

\[
\Delta_{ic} = \sum_{k=0}^{N-1} \|v_k\|_2, \quad \Delta_{dyn} = \sum_{k=0}^{N-1} \|f(t_k, \bar{x}_k, \bar{u}_k, 0) - \bar{x}_{k+1}\|_2,
\]

\[
\Delta_T = \|x_N - \bar{x}_N\|_2^2 + \sum_{k=0}^{N-1} \|x_k - \bar{x}_k\|_2^2 + \|u_k - \bar{u}_k\|_2^2,
\]

\[
\Delta_F = \|Q_N - \bar{Q}_N\|_F^2 + \sum_{k=0}^{N-1} \|Q_k - \bar{Q}_k\|_F^2 + \|Y_k - \bar{Y}_k\|_F^2.
\]

Then the stopping criteria is given as the following logical statement:

\[
(\Delta_{ic} < \Delta_{ic}^{tol}) \land (\Delta_{dyn} < \Delta_{dyn}^{tol}) \land (\Delta_T < \Delta_T^{tol}) \land (\Delta_F < \Delta_F^{tol}), \tag{29}
\]

where \( \Delta_{ic}^{tol}, \Delta_{dyn}^{tol}, \Delta_T^{tol} \) and \( \Delta_F^{tol} \) are user-defined tolerance parameters. The proposed algorithm is summarized in Algorithm 1.

While the convergence guarantee of the proposed method has not been a focus of this paper, one can construct a safety alternative that is assured not to diverge by modifying the proposed algorithm with results from References 2 and 26. Instead of updating the trajectory and the funnel sequentially in each iteration, the safety approach performs updating only the trajectory with a fixed funnel until convergence of the nominal trajectory is achieved. This part of the process, being solely trajectory optimization, benefits from the established convergence results in Reference 26. The subsequent phase involves computing the Lipschitz constant and updating the funnel with the computed nominal trajectory, with convergence analysis from Thm. 6.12 of Reference 2. Since each phase of the safety approach has a guaranteed convergence, it prevents the overall solution from diverging.
Algorithm 1. Joint synthesis

Input: $(\hat{x}_k, \hat{u}_k, \hat{Q}_k, \hat{K}_k)$

for $i = 1 \ldots N_{max}$ do
  optimize $\tilde{x}_i, \tilde{u}_i$ by (11)
  estimate $\gamma_k$ via (23) or (28)
  optimize $Q_k, K_k$ by (20)
  if (29) is True then
    break
  end if
  update $(\hat{x}_k, \hat{u}_k, \hat{Q}_k, \hat{K}_k) \leftarrow (\tilde{x}_k, \tilde{u}_k, Q_k, K_k)$
end for
Output: $(\hat{x}_k, \hat{u}_k, Q_k, K_k)$

4 | NUMERICAL SIMULATION

In this section, we validate the proposed method via two robotic applications with a unicycle model and a 6-DoF free-flying spacecraft. For both examples, we used an Apple MacBook Pro having M1 Pro with 8-core CPU, and the simulation result can be reproduced using the code available at https://github.com/taewankim1/joint_synthesis.

4.1 | Unicycle model

We consider the motion of a unicycle-type model under different disturbance conditions, represented by $w_1$ and $w_2$

Model I:

$$
\begin{bmatrix}
\dot{r}_x \\
\dot{r}_y \\
\dot{\theta}
\end{bmatrix} =
\begin{bmatrix}
u_v \cos \theta + 0.1 w_1 \\
u_v \sin \theta + 0.1 w_2 \\
u_\theta
\end{bmatrix},
$$

Model II:

$$
\begin{bmatrix}
\dot{r}_x \\
\dot{r}_y \\
\dot{\theta}
\end{bmatrix} =
\begin{bmatrix}
(u_v + 0.1 w_1) \cos \theta \\
(u_v + 0.1 w_1) \sin \theta \\
u_\theta + 0.1 w_2
\end{bmatrix},
$$

Model III:

$$
\begin{bmatrix}
\dot{r}_x \\
\dot{r}_y \\
\dot{\theta}
\end{bmatrix} =
\begin{bmatrix}
u_v \cos(\theta + 0.03 w_1) \\
u_v \sin(\theta + 0.03 w_1) \\
u_\theta + 0.05 w_2
\end{bmatrix},
$$

where $r_x, r_y,$ and $\theta$ are x-axis position, y-axis position, and heading angle, respectively, and $u_v \in \mathbb{R}$ is velocity and $u_\theta \in \mathbb{R}$ is angular velocity. The scalars $w_1$ and $w_2$ represent disturbances or model mismatch. Model I depicts direct disturbances on the translational motion, Model II introduces disturbances affecting both the velocity and rotational control inputs, and Model III captures disturbances influencing the orientation and the rotation control. These models are considered to have a comprehensive understanding of how the system behaves according to different types of the disturbances. It is worth noting that in Model II and Model III, the disturbances introduce additional nonlinearity to the system, while in Model I, they appear as linear additive terms.

For all unicycle models, we consider $N = 30$ nodes evenly distributed over a time horizon of 10 s, that is, $t_0 = 0$ and $t_f = 10$. The continuous-time model (30) is discretized by following a variational approach. The initial boundary set $X_0$ and the final boundary set $X_f$ in (8) have the following parameters: $x_0 = [0, 0, 0]^T$, $Q_f = Q_f = \text{diag}(0.2^2 (\text{m}), 0.2^2 (\text{m}), 10^2 (\text{deg})^2)$, and $x_f = [8, 4, 0]^T$. There are multiple circular obstacles the unicycle robot should avoid, which leads to nonconvex constraints on the state represented in set $\mathcal{X}$. All obstacles have a diameter of 1.0 m, and their center positions are illustrated in Figure 4. The input constraints for the set $\mathcal{U}'$ are given as: $0 \leq u_v \leq 1.5$ and $|u_\theta| \leq 1.0$ (rad). The cost function for the trajectory $J$ is a quadratic function of the input given by $u_v^2 + u_\theta^2$. Both weight parameters $w_Q, w_K$ in (7a) are chosen as 1. The decay rate $\alpha$ is set as 0.99 and the parameter $\Delta_t^w$ is set as 0.2 for all $k$. The tolerance parameters $\Delta_t^\text{tol}, \Delta_t^\text{syn}, \Delta_t^\text{pol}$ and $\Delta_F^\text{pol}$ are all $10^{-8}$. The number of samples $N_s$ used for the Lipschitz constant $\gamma_k$ estimation is set as 100, for each $k$, so a total of 3000 samples are used for each iteration. We use an interior-point method solver, Clarabel, for both the trajectory update (11) and the funnel update (20), using CVXPY in Python.
To test the invariance and the feasibility properties, we sample 100 points at the surface of the ellipse \( \mathcal{E}_{Q_k} \) at \( k = 0 \), and then generate the corresponding 100 trajectories with the nonlinear dynamics (1) and the control law (3) under the presence of the disturbances. In this generation process, we randomly set the disturbance \( w = (w_1, w_2) \) such that \( \|w\|_2 = 1 \) and keep them constant during the entire horizon for each sample. Note that making the disturbance constant during the entire horizon increases the impact of the disturbance compared to varying the disturbance randomly for each interval.

The computed nominal trajectory and the CIF for all unicycle models (30) are depicted in Figure 4, and the input results are given in Figure 5. The test results of the invariance property for the trajectory samples is given in Figure 6 where the radius \( r_Q^O \) is defined as

\[
r_Q^O := (x^*_k - \bar{x}_k)^\top Q_k^{-1}(x^*_k - \bar{x}_k),
\]

for each sample \( s \) and time \( k \). The result shows that the nominal trajectory and the CIF satisfy the invariance and feasibility conditions. For the initial guess, we employ the first method illustrated in Section 3.3 using the straight-linear interpolation. The convergence performance in Figure 7 shows that the proposed approach makes the trajectory and the CIF satisfy the tolerances as the iteration count increases. Table 1 summarizes the average computational time of each subproblem within the iterations.

To obtain a baseline solution to compare against, we compute an approximate funnel that is generated with the linear closed-loop system where the higher-order terms are ignored, that can be established by setting \( E_k = 0 \) and \( \gamma_k = 0 \) in (14), as considered in References 17 and 18. It is worth noting that the approximate funnel, which is used for the comparison with the proposed method, can yield more optimal solutions compared to References 17 and 18 under the linear approximation. This is because the approximate funnel is computed by simultaneously optimizing the linear feedback gains and the invariant set parameters as decision variables, whereas Reference 17 determines the invariant set variables by the uncertainty forward equation and Reference 18 sets the feedback gains by solving a discrete-time LQR problem.
FIGURE 5  Nominal trajectories and synthesized input funnels (projected on each input coordinate) of Model I (left), Model II (middle), and Model III (right). The zeroth-order hold on the input is used to generate the nominal trajectory.

FIGURE 6  Invariance property tests for Model I (left), Model II (middle), and Model III (right).

FIGURE 7  Convergence performance of the proposed method for the unicycle models.
The approximate nominal trajectory and funnel are depicted in Figure 4, and the invariance test \( r^Q_k \leq 1 \) in (31) with the trajectory samples is given in Figure 6. We can see that the value of \( r^Q_k \) for the approximate funnel is greater than 1 especially for Model III since the bounded disturbances contribute the nonlinearity of the system. This violation shows that the approximate CIF does not necessarily guarantee the invariance property for the original nonlinear system, which can result in safety issues for safety-critical nonlinear systems. As the contribution of higher-order terms increase, for example, for large Lipshitz constants, these violations can become more pronounced.

### 4.2 6-DoF free-flying spacecraft

We consider the following 6-DoF free-flying spacecraft dynamics\(^{1,31}\) under the presence of disturbances:

\[
\begin{align*}
\dot{r}_I &= v_I, \\
\dot{v}_I &= m^{-1}(T_I + \beta_I w_T), \\
\Phi &= R(\Phi)\omega_B, \\
\dot{\omega}_B &= J^{-1}(M_B + \beta_M w_M - \omega_B \times J \omega_B).
\end{align*}
\]

The state of the system (32) consists of the inertial position \( r_I \in \mathbb{R}^3 \), the inertial velocity \( v_I \in \mathbb{R}^3 \), the ZYX Euler angles \( \Phi \in \mathbb{R}^3 \), and the body angular velocity \( \omega_B \in \mathbb{R}^3 \). The control input consists of the inertial thrust \( T_I \in \mathbb{R}^3 \) and the body torque \( M_B \in \mathbb{R}^3 \). The constant \( m \in \mathbb{R} \) is the mass, matrix \( J \in \mathbb{R}^{3 \times 3} \) is the inertia matrix, and \( R \) maps the Euler angles to the rotation matrix

\[
R(\Phi) = \begin{bmatrix}
1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\
0 & \cos \phi & -\sin \phi \\
0 & \sin \phi \sec \theta & \cos \phi \sec \theta
\end{bmatrix},
\]

with \( \Phi = [\phi, \theta, \psi]^T \). For the mass and inertia matrix parameters, we choose \( m = 7.2 \) (kg) and \( J = 0.1083I_{3 \times 3} \) (kg m\(^2\)) where \( I_{3 \times 3} \) is the 3 by 3 identity matrix. The vector \( w_T \in \mathbb{R}^3 \) and \( w_M \in \mathbb{R}^3 \) affect the system as disturbances by acting on the control inputs \( T_I \) and \( M_B \) with coefficients \( \beta_T = 10^{-3} \) and \( \beta_M = 10^{-6} \). Further details about the free-flying system dynamics can be found in Reference 1.

For the free-flying spacecraft example, we consider \( N = 15 \) nodes evenly distributed over a time horizon of 200 s. The initial boundary set \( \mathcal{X}_0 \) and the final boundary set \( \mathcal{X}_f \) in (8) have the following parameters:

\[
\begin{align*}
\mathcal{X}_0 &= [r_0^T, v_0^T, \Phi_0^T, \omega_0^T]^T, \\
\mathcal{X}_f &= [r_f^T, v_f^T, \Phi_f^T, \omega_f^T]^T, \\
Q_I &= Q_f = \text{diag}\left(0.2^2, 0.2^2, 0.2^2, 0.02^2, 0.02^2, 0.02^2, \left(\frac{5\pi}{180}\right)^2, \left(\frac{5\pi}{180}\right)^2, \left(\frac{5\pi}{180}\right)^2, \left(\frac{0.1\pi}{180}\right)^2, \left(\frac{0.1\pi}{180}\right)^2, \left(\frac{0.1\pi}{180}\right)^2\right).
\end{align*}
\]

As state constraints for the set \( \mathcal{X} \), there are two cylindrical obstacles to avoid, and all obstacles have a diameter of 0.8 m, and their center positions are illustrated in Figure 8. In addition, we have constraints on the velocity and the angular velocity that are \( ||v_I||_2 \leq 0.4 \) (m/s) and \( ||\omega_B||_2 \leq 1 \) (deg). The input constraints for the set \( \mathcal{U} \) are given as: \( ||T_I||_2 \leq 10 \) (mN) and \( ||M_B||_2 \leq 50 \) (\( \mu \)Nm). The decay rate \( \alpha \) is set as 0.99 and the parameter \( \lambda^w_k \) is set as 0.1 for all \( k \). Similar to the unicycle examples, the tolerance parameters \( \Delta_{\text{wloc}}, \Delta_{\text{dyn}}, \Delta_T, \Delta_F \) and \( \Delta_{\text{tol}} \) are all \( 10^{-8} \). The number of samples \( N_s \) used for the

| Subproblem | Trajectory update | Estimate \( \gamma_k \) | Funnel update |
|------------|------------------|---------------------------|--------------|
| Model I    | 0.026            | 0.863                     | 0.577        |
| Model II   | 0.026            | 0.895                     | 0.691        |
| Model II   | 0.025            | 0.904                     | 0.880        |

### TABLE 1 Average computational time (s) for each iteration.
Lipschitz constant $\gamma_k$ estimation is set as 256 for each $k$. We provide the initial guess using the second method illustrated in Section 3.3 that uses the result of the separate synthesis, and these used initial trajectory and funnel are illustrated in Figure 8. Starting from the initial guess, the proposed algorithm converges at 6 iterations. We use Clarabel for the trajectory update (11) and MOSEK for the funnel update (20), using CVXPY in Python3. Mosek’s solve time is observed to scale better than Clarabel’s for large problem sizes. So, for the free-flyer system, we have used Mosek for the funnel update. The average computational time (s) of the trajectory update, the estimation of $\gamma_k$, and the funnel update at each iteration are 0.024, 2.698, and 10.072, respectively.

The results of the synthesized trajectory and funnel projected on position coordinates are illustrated in Figure 8. It is clear that the resulting funnel is feasible to the obstacle avoidance constraints although the initial guess has infeasible trajectory and funnel. Similar to the test performed for the unicycle models, to test the invariance of the synthesized trajectory and funnel, we sample 300 at the surface of the initial funnel $E_{Q_0}$ and generate the corresponding 300 trajectories by propagating the system dynamics (32) with the control law (3) consisting of the open-loop input in the nominal trajectory and the feedback control from the funnel. For each sample, we randomly set the disturbance $w = [w_r^T, w_M^T]^T$ such that $\|w\|_2 = 1$. The input results of the nominal trajectory and the samples are illustrated in Figure 9. We can see that the input history of the samples remain feasible within the given input constraints. Finally, we obtain the values of $r_Q^k$ by computing (31) for each trajectory sample, illustrated in Figure 10. The result shows that $r_Q^k$ for each sample and for all time $k$ maintain less than 1 and hence the trajectory samples remain inside the funnel.
FIGURE 10  Invariance property tests for trajectory samples of the free-flying spacecraft.

5 | CONCLUSIONS

This paper introduces a method for joint trajectory optimization and funnel synthesis for locally Lipschitz nonlinear systems under the presence of disturbances. The proposed method has a recursive approach in which both nominal trajectory and funnel are iteratively updated. The trajectory update step optimizes the nominal trajectory to satisfy the feasibility of the funnel. Then, the funnel update step solves an SDP to guarantee the invariance property of the funnel. The numerical evaluation for a unicycle model and a 6-DoF free-flying spacecraft shows that the converged trajectory and funnel satisfy the invariance and feasibility properties under the disturbances.

ACKNOWLEDGMENTS

We thank Dr. Taylor P. Reynolds and Prof. Mehran Mesbahi for their helpful inputs on funnel synthesis algorithms. This work is supported in part by Boeing under Grant 2021-PD-PA-471, Office of Navel Research under Grant N00014-20-1-2288, and AFOSR under Grant FA9550-20-1-0053. Taewan Kim is supported by UW+Amazon Science Hub fellowship.

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no conflict of interest to this work.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Taewan Kim https://orcid.org/0000-0003-4843-2630

REFERENCES

1. Malyuta D, Reynolds TP, Szmuk M, et al. Convex optimization for trajectory generation: a tutorial on generating dynamically feasible trajectories reliably and efficiently. IEEE Control Syst Mag. 2022;42(5):40-113.
2. Reynolds T, Malyuta D, Mesbahi M, Açıkmeše B, Carson JM. Funnel synthesis for the 6-DOF powered descent guidance problem. Paper presented at: AIAA SciTech 2021 Forum. 2021:0504. https://www.aiaa.org/SciTech/utility/past-forums#2021
3. Kim T, Elango P, Reynolds TP, Açıkmeše B, Mesbahi M. Optimization-based constrained funnel synthesis for systems with Lipschitz nonlinearities via numerical optimal control. IEEE Control Syst Lett. 2023;7:2875-2880. doi:10.1109/LCSYS.2023.3290229
4. Bonalli R, Cauligi A, Bylard A, Pavone M. GuSTO: guaranteed sequential trajectory optimization via sequential convex programming. 2019 International Conference on Robotics and Automation (ICRA). IEEE; 2019:6741-6747.
5. Majumdar A, Tedrake R. Funnel libraries for real-time robust feedback motion planning. Int J Robot Res. 2017;36(8):947-982.
6. Buch J, Seiler P. Finite horizon robust synthesis using integral quadratic constraints. Int J Robust Nonlinear Control. 2021;31(8):3011-3035.
7. Xu X, Açıkmeše B, Corless MJ. Observer-based controllers for incrementally quadratic nonlinear systems with disturbances. IEEE Trans Autom Control. 2020;66(3):1129-1143.
8. Bemporad A, Morari M. Robust model predictive control: a survey. In: Milanese M, Tempo R, Vicino A, eds. Robustness in Identification and Control. Springer; 1999:207-226. Accessed January 11, 2024. https://link.springer.com/book/10.1007/978-1-4615-9552-6
9. Yu S, Maier C, Chen H, Allgöwer F. Tube MPC scheme based on robust control invariant set with application to Lipschitz nonlinear systems. Systems Control Lett. 2013;62(2):194-200.
10. Köhler J, Soloperto R, Müller MA, Allgöwer F. A computationally efficient robust model predictive control framework for uncertain nonlinear systems. IEEE Trans Autom Control. 2020;66(2):794-801.
11. Villanueva ME, Quirynen R, Diehl M, Chachuat B, Houska B. Robust MPC via min–max differential inequalities. Automatica. 2017;77:311-321.
12. Açıkmeşe B, Carson JM III, Bayard DS. A robust model predictive control algorithm for incrementally conic uncertain/nonlinear systems. Int J Robust Nonlinear Control. 2011;21(5):563-590.
13. Garimella G, Scheckells M, Moore JL, Kobilarov M. Robust obstacle avoidance using tube NMPC. Robot Sci Syst. 2018. Accessed January 11, 2024. https://www.roboticsproceedings.org/rss14/p55.html
14. Richards A, How JP. Robust variable horizon model predictive control for vehicle maneuvering. Int J Robust Nonlinear Control. 2006;16(7):333-351.
15. Di Cairano S, Park H, Kolmanovsky I. Model predictive control approach for guidance of spacecraft rendezvous and proximity maneuvering. Int J Robust Nonlinear Control. 2012;22(12):1398-1427.
16. Seo H, Lee D, Son CY, Jang I, Tomlin CJ, Kim HJ. Real-time robust receding horizon planning using Hamilton–Jacobi reachability analysis. IEEE Trans Robot. 2022;39(1):90-109.
17. Messerer F, Diehl M. An efficient algorithm for tube-based robust nonlinear optimal control with optimal linear feedback. 2021 60th IEEE Conference on Decision and Control (CDC). IEEE; 2021:6714-6721. Accessed January 11, 2024. https://ieeexplore.ieee.org/document/9683712
18. Manchester Z, Kuindersma S. Robust direct trajectory optimization using approximate invariant funnels. Auton Robots. 2019;43(2):375-387.
19. Seiler P, Moore RM, Meissen C, Arcak M, Packard A. Finite horizon robustness analysis of LTV systems using integral quadratic constraints. Automatica. 2019;100:135-143.
20. Seo CY, Kim HJ. Fast funnel computation using multivariate bernstein polynomial. IEEE Robot Autom Lett. 2021;6(2):1351-1358.
21. Jang I, Seo H, Kim HJ. Fast computation of tight funnels for piecewise polynomial systems. IEEE Control Syst Lett. 2021;6(2):1351-1358.
22. Leeman AP, Köhler J, Zanelli A, Bennani S, Zeilinger MN. Robust nonlinear optimal control via system level synthesis. arXiv preprint arXiv:2301.04943. 2023.
23. Leeman AP, Sieber J, Bennani S, Zeilinger MN. Robust optimal control for nonlinear systems with parametric uncertainties via system level synthesis. arXiv preprint arXiv:2304.00752. 2023.
24. Boyd S, Vandenberghe L. Convex Optimization. Cambridge University Press; 2004.
25. Reynolds TP, Szmuk M, Malyuta D, Mesbahi M, Açıkmeşe B, Carson JM III. Dual quaternion-based powered descent guidance with state-triggered constraints. J Guid Control Dyn. 2020;43(9):1584-1599.
26. Cartis C, Gould NI, Toint PL. On the evaluation complexity of composite function minimization with applications to nonconvex nonlinear programming. SIAM J Optim. 2011;21(4):1721-1739.
27. Yakubovich VA. S-procedure in nonlinear control theory. Vestn Leningrad Univ Math. 1997;4:73-93.
28. Chou G, Ozay N, Berenson D. Model error propagation via learned contraction metrics for safe feedback motion planning of unknown systems. 2021 60th IEEE Conference on Decision and Control (CDC). IEEE; 2021:3576-3583. Accessed January 11, 2024. https://ieeexplore.ieee.org/document/9683354
29. Reynolds TP. Computational Guidance and Control for Aerospace Systems. University of Washington; 2020.
30. Diehl M, Gros S. Numerical optimal control. Optim Eng Center. 2011. Accessed January 11, 2024. https://cdn.syscop.de/publications/Diehl2017.pdf
31. Lew T, Bonalli R, Pavone M. Chance-constrained sequential convex programming for robust trajectory optimization. 2020 European Control Conference (ECC). IEEE; 2020:1871-1878. Accessed January 11, 2024. https://ieeexplore.ieee.org/document/9143595

How to cite this article: Kim T, Elango P, Açıkmeşe B. Joint synthesis of trajectory and controlled invariant funnel for discrete-time systems with locally Lipschitz nonlinearities. Int J Robust Nonlinear Control. 2024;34(6):4157-4176. doi: 10.1002/rnc.7186