

Solving Optimal Control with Nonlinear Dynamics Using Sequential Convex Programming

Rebecca Foust*

University of Illinois at Urbana-Champaign, Champaign, IL, 61801

Soon-Jo Chung †

California Institute of Technology, Pasadena, CA 91125

Fred Y. Hadaegh ‡

Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 91109

Sequential convex programming (SCP) is a useful tool in obtaining real-time solutions to direct optimal control, but it is unable to adequately model nonlinear dynamics due to the linearization and discretization required. As nonlinear program solvers are not yet functioning in real-time, a tool is needed to bridge the gap between satisfying the nonlinear dynamics and completing execution fast enough to be useful. This paper presents a real-time control algorithm, sequential convex programming with nonlinear dynamics correction (SCPn), which ameliorates the performance of SCP under nonlinear dynamics. Simulations are presented to validate the efficacy of the method.

Nomenclature

β	=	Trust region shrink rate
dt	=	Time step
\mathbf{f}	=	Nonlinear dynamics at time step k
\mathcal{F}	=	Cost integrand function
g_i	=	Convex inequality constraints
J	=	Cost function
k	=	Discrete time parameter
N	=	Number of agents
N_j	=	Set of neighbor agents
p	=	Number of inequality constraints
R_{col}	=	Collision avoidance radius
R_{comm}	=	Communication radius
\mathbf{x}_0	=	Initial state constraint
\mathbf{x}_f	=	Terminal state constraint
$\mathbf{x}_k^{(w)}$	=	State trajectory at the k -th time step for the w -th iteration
$\mathbf{x}_{n,k}^{(w)}$	=	Nominal, nonlinear state trajectory at the k -th time step for the w -th iteration
t	=	Time
t_0	=	Initial time
t_f	=	Final time
T	=	Final discrete time step
$\mathbf{u}_k^{(w)}$	=	Control input trajectory at the k -th time step for the w -th iteration
U_{max}	=	Maximum control input
w	=	SCPn iteration
v_k	=	Numerical integration quadrature weight
$\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3, \mathbf{y}_4$	=	Classical Runge-Kutta parameters

*Graduate Student, Department of Aerospace Engineering, 104 S Wright St, Urbana, IL 61801, AIAA Student Member

†Associate Professor of Aerospace and Bren Scholar, GALCIT, Pasadena, CA, 91125, AIAA Member

‡Senior Research Scientist and Chief Technologist, 4800 Oak Grove Dr, Pasadena, CA 91109, AIAA Fellow

I. Introduction

As the field of robotics advances, more complex robots emerge that are capable of fast, highly dynamic motion. Such robots need control algorithms that are equally as advanced. Optimal control methods for nonlinear systems are currently lacking in real-time implementations, so other avenues must be taken to achieve fast, efficient motion onboard advanced robots. In previous work, sequential convex programming was used to successively linearize the nonlinear dynamics about a trajectory, but the error in linearization and discretization of the dynamics adds up and results in a trajectory that may no longer satisfy the original constraints. This paper seeks to augment sequential convex programming to satisfy the nonlinear dynamics and constraints.

Several methods exist to tackle nonlinear optimal control problems, but they often fall short in capability when it comes to real-time implementation for multi-agent systems. Pseudospectral methods are well suited to handle the nonlinear dynamics, but grow computationally prohibitive as the number of agents increases and are not yet implemented in real-time [1, 2]. Mixed integer linear programming can be implemented in real-time, but also scales poorly with the number of agents [3]. Convex optimization is easily and efficiently solved, but the safety constraints for collision avoidance can make the problem intractable particularly when the workspace is densely populated by obstacles or other agents. Sequential convex programming (SCP) parses the full convex problem into a sequence of convex programs with less restrictive collision avoidance constraints. This prevents the constraint gridlock and allows the algorithm to make use of efficient convex optimization solvers. For these reasons, previous work in our lab has used SCP for nonlinear problems ranging from collision-free guidance and control of swarms [4–7] to optimal robotic motion planning in cluttered environments [8]. In order to make the nonlinear optimal control problem convex, the problem must be discretized and the nonlinear dynamics constraint must be linearized. The linearization and discretization of the dynamics cause error to accumulate with each successive iteration of SCP. This error causes the linearized/discretized optimal trajectory to diverge from the nonlinear dynamics and can lead to failed constraints as seen in Figure 1 where the terminal constraint is missed. This behavior must be corrected before implementation on fragile problems, like multi-agent docking or aggressive, cluttered trajectory planning where small errors in the commanded trajectory can result in mission failure.

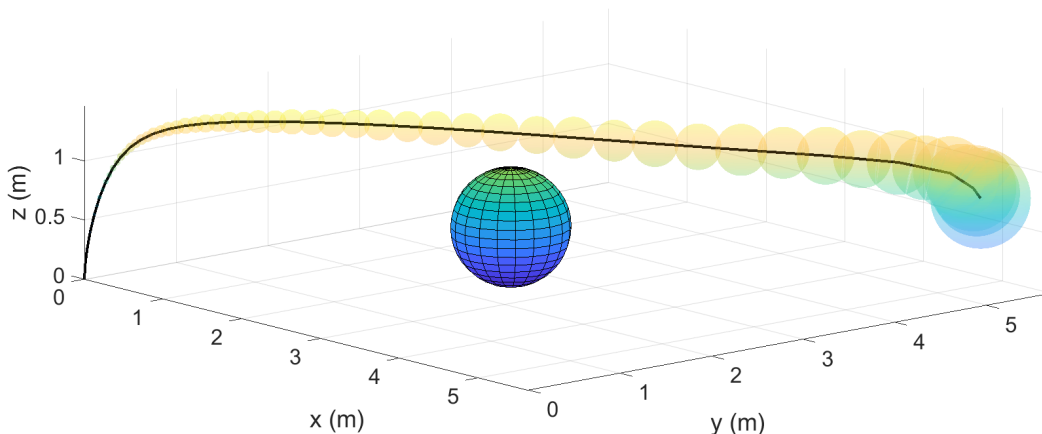


Fig. 1 Linearization and discretization error accumulates over time when trajectories are found using SCP. Mean trajectory error resulting from SCP is shown by the colored circles along the nonlinear trajectory

This paper presents an efficient upgrade to the widely used SCP method which allows the method to be used to more accurately solve nonlinear optimal control problems. The new algorithm, SCPn, corrects for the linearization and discretization error in each SCP loop. This prevents the error from building up and allows the algorithm to converge to a solution that satisfies the nonlinear dynamics and the optimization constraints. A new constraint is added to SCPn to ensure that the resulting optimal trajectory is a feasible solution for the original nonconvex problem. Using this constraint, it is proven that SCPn converges to the optimal solution and the solution remains feasible for the nonlinear dynamics. SCPn is then validated through simulations involving a quadrotor traversing a simple obstacle field. The paper is organized as follows. The nonlinear optimal control problem in continuous-time is presented in Section II.A, then recast into a discrete-time nonlinear optimal control problem in Section III.A along with the convex optimal control

problem with linearized dynamics, and the convex optimal control problem with nonlinear dynamics. Then, in Section IV the convergence and optimality proofs are presented for SCPn and another method, M-SCPn, is proposed. In Section V, the results of quadrotor dynamics simulations of SCP, SCPn, and M-SCPn are presented. Finally, Section VI contains the conclusions and future work for the paper.

II. Problem Statement

In this section, the original continuous-time nonlinear optimal control problem is presented.

A. Nonlinear Optimal Control Problem

We define the original finite-horizon optimal control problem for $\mathbf{x}(t) \in \mathbb{R}^n$ and $\mathbf{u}(t) \in \mathbb{R}^m$ as follows

Problem 1 (Constrained, Nonlinear Optimal Control).

$$\underset{\mathbf{x}, \mathbf{u}}{\text{minimize}} \quad \int_{t_0}^{t_f} \mathcal{F}(\mathbf{x}(t), \mathbf{u}(t)) dt \quad \text{subject to} \quad (1)$$

$$\dot{\mathbf{x}}(t) - \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t)) = \mathbf{0} \quad \forall t \in [t_0, t_f] \quad (2)$$

$$\tilde{g}_i(\mathbf{x}(t), \mathbf{u}(t)) \leq \mathbf{0}, \quad i = 1, \dots, r, \quad \forall t \in [t_0, t_f] \quad (3)$$

$$\mathbf{x}(t_0) = \mathbf{x}_0, \quad \mathbf{x}(t_f) = \mathbf{x}_f, \quad (4)$$

where (2)-(3) represent the dynamics constraint and the general inequality constraint of the state (\mathbf{x}) and the control vector (\mathbf{u}), respectively. Examples of the cost integrand function include $\mathcal{F}(\mathbf{x}(t), \mathbf{u}(t)) = \|\mathbf{u}(t)\|_q$. Then, the \mathcal{L}_1 integration of \mathcal{F} shown in (1) correctly captures the fuel cost of a spacecraft dynamic model with $q \in \{1, 2, \infty\}$.

Example 1 (State and Control Constraint).

If Problem 1 involves multiple (N) agents [4, 5] such that $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_N)$ and $\mathbf{u} = (\mathbf{u}_1, \dots, \mathbf{u}_N)$, examples of $\tilde{g}_i(\mathbf{x}(t), \mathbf{u}(t)) \leq 0$ in (3) include the following maximum control constraint and collision avoidance constraint:

$$\|\mathbf{u}(t)\|_r \leq U_{\max} \quad \forall t \in [t_0, t_f], \quad r \in \{1, 2, \infty\}, \quad j = 1, \dots, N \quad (5)$$

$$\|G[\mathbf{x}_j(t) - \mathbf{x}_i(t)]\|_2 \geq R_{\text{col}} \quad \forall t \in [t_0, t_f], \quad i \in \mathcal{N}_{[j]}, \quad j = 1, \dots, N \quad (6)$$

$$\mathcal{N}_{[j]} = \{i \mid \|\mathbf{x}_j(t) - \mathbf{x}_i(t)\|_2 \leq R_{\text{comm}}\} \quad (7)$$

where $G = [\mathbf{I}_{3 \times 3} \quad \mathbf{0}_{3 \times 3}]$ is used to take the position state if each $\mathbf{x}_j(t)$ is composed of both the position and velocity states. Also, R_{comm} is the communication radius of each agent, U_{\max} denotes the maximum control magnitude, and R_{col} denotes the minimum allowable distance between two agents. Note that the q in and r in the norms $\|\mathbf{u}\|_q$ in (1) and $\|\mathbf{u}\|_r$ in (5) could be different depending on the thruster/actuator architecture. Note that the collision avoidance constraints in (6) can be convexified into a convex polytope around the nominal position of the spacecraft, drawn from the intersection of half-space approximations of (6) [4, 5].

III. Convex Optimization with Direct Transcription of Dynamics

To solve Problem 1 efficiently, the state and control constraints (3) are assumed to be convexified and decoupled so that each agent can use SCP to determine its optimal trajectories.

A. Non-Convex Optimization Problem Using Discretization

The first step in the process of converting (2) into a constraint that can be used in direct optimization is to convert the ordinary differential equation in (2) to a finite number of algebraic constraints by using a zero-order hold discretization approach such that $\mathbf{x}_k = \mathbf{x}(t_k)$, $\mathbf{u}_k = \mathbf{u}(t_k)$ for the $\mathbf{x}(t)$ and $\mathbf{u}(t)$ values at $t \in [t_k, t_{k+1})$, $k = k_0, \dots, T-1$, where T is the final discrete time step ($t_T = t_f$) and $t_{k_0} = t_0$. The stacked vector is denoted by $\mathbf{x}_{k_0:T} = (\mathbf{x}_{k_0}, \dots, \mathbf{x}_T)$ and $\mathbf{u}_{k_0:T-1} = (\mathbf{u}_{k_0}, \dots, \mathbf{u}_{T-1})$.

Furthermore, we assume that the integrand cost function \mathcal{F} in (1) is written as

$$\mathcal{F}(\mathbf{x}(t), \mathbf{u}(t)) = \mathcal{F}_x(\mathbf{x}(t)) + \mathcal{F}_u(\mathbf{u}(t)) \quad (8)$$

where $\mathcal{F}_x : \mathbb{R}^n \mapsto \mathbb{R}^1$ and $\mathcal{F}_u : \mathbb{R}^m \mapsto \mathbb{R}^1$ is a convex function. First, we consider the problem with \mathcal{F}_u only before generalizing the cost function to (8).

The discretized version of Problem 1 is written as the following optimization:

Problem 2 (Non-Convex Program (NCP) with Nonlinear Dynamics).

$$\underset{\mathbf{u}_{k_0:T-1}}{\text{minimize}} \quad \sum_{k=k_0}^{T-1} \mathcal{F}_u(\mathbf{u}_k) v_k \quad \text{subject to} \quad (9)$$

$$\mathbf{x}_{k+1} - \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) = \mathbf{0}, \quad \mathbf{x}_{k_0} = \mathbf{x}_0, \quad k = k_0, \dots, T-1 \quad (10)$$

$$g_i(\mathbf{x}_k, \mathbf{u}_k) \leq 0, \quad i = 1, \dots, p, \quad k = k_0, \dots, T \quad (11)$$

where v_k denotes the quadrature weight of numerical integration (e.g., $v_k = \Delta t = t_{k+1} - t_k$ for the Euler method or see [9] for pseudospectral integration). Note that $g_i \leq 0, i = 1, \dots, p$ includes all the convex (or convexified) inequality constraints (3) as well as the initial and terminal time conditions of (4) relaxed as inequality constraints (e.g., $\|\mathbf{x}_{k=T} - \mathbf{x}_f\| \leq \epsilon$). Hence, Problem 2 is a nonconvex optimization problem simply because of (10).

Example 2 (Discretization of Dynamics (2)). *The discrete-time nonlinear dynamic model (10) of the dynamic model of (2) can be derived from the fourth-order Runge-Kutta (RK4) integration scheme as follows*

$$\begin{aligned} \mathbf{y}_{1k} &= \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k) \\ \mathbf{y}_{2k} &= \mathbf{f}\left(\mathbf{x}_k + \frac{\Delta t}{2} \mathbf{y}_{1k}, \mathbf{u}_{k+\frac{1}{2}}\right) \\ \mathbf{y}_{3k} &= \mathbf{f}\left(\mathbf{x}_k + \frac{\Delta t}{2} \mathbf{y}_{2k}, \mathbf{u}_{k+\frac{1}{2}}\right) \\ \mathbf{y}_{4k} &= \mathbf{f}(\mathbf{x}_k + \Delta t \mathbf{y}_{3k}, \mathbf{u}_{k+1}) \\ \mathbf{x}_{k+1} &= \mathbf{x}_k + \frac{\Delta t}{6} (\mathbf{y}_{1k} + 2\mathbf{y}_{2k} + 2\mathbf{y}_{3k} + \mathbf{y}_{4k}) \end{aligned} \quad (12)$$

where $\mathbf{u}_{k+\frac{1}{2}} = \frac{\mathbf{u}_k + \mathbf{u}_{k+1}}{2}$. SCPn is independent of the type of numerical integration/discretization method used, for further options see [9–11].

B. Sequential Convex Programming with Linearized Constraints

In prior work [4], the dynamics and nonconvex constraints in Problem 2 are sequentially linearized about some nominal trajectory to form a convex program. The basic problem formulation is presented in Problem 3 and the method is presented in Method 1. The starting nominal trajectory is an initial guess of the solution, then the solution of the previous iteration of SCP is used for all subsequent nominal trajectories. For more details see references [4, 5].

Problem 3 ((w)-th Sequential Convex Program: SCP^($w+1$)($\bar{\mathbf{x}}, \bar{\mathbf{u}}$) given $\bar{\mathbf{x}}_{k_0:T} = \mathbf{x}_{k_0:T}^{(w-1)}, \bar{\mathbf{u}}_{k_0:T-1} = \mathbf{u}_{k_0:T-1}^{(w-1)}$).

$$\underset{\mathbf{u}_{k_0:T-1}}{\text{minimize}} \quad \sum_{k=k_0}^{T-1} \mathcal{F}_u(\mathbf{u}_k) v_k \quad \text{subject to} \quad (13)$$

$$\mathbf{x}_{k+1} - A(\bar{\mathbf{x}}_k) \mathbf{x}_k - B(\bar{\mathbf{u}}_k) \mathbf{u}_k - z(\bar{\mathbf{x}}_k, \bar{\mathbf{u}}_k) = \mathbf{0}, \quad \mathbf{x}_{k_0} = \mathbf{x}_0, \quad k = k_0, \dots, T-1 \quad (14)$$

$$g_i(\mathbf{x}_k, \mathbf{u}_k) \leq 0, \quad i = 1, \dots, p, \quad k = k_0, \dots, T \quad (15)$$

where $A(\bar{\mathbf{x}}_{n,k}) = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}_k} \right|_{(\bar{\mathbf{x}}_{n,k}, \bar{\mathbf{u}}_k)}$, $B(\bar{\mathbf{u}}_k) = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}_k} \right|_{(\bar{\mathbf{x}}_{n,k}, \bar{\mathbf{u}}_k)}$, and $z(\bar{\mathbf{x}}_{n,k}, \bar{\mathbf{u}}_k) = \mathbf{f}(\bar{\mathbf{x}}_{n,k}, \bar{\mathbf{u}}_k) - A(\bar{\mathbf{x}}_{n,k}) \bar{\mathbf{x}}_{n,k} - B(\bar{\mathbf{u}}_k) \bar{\mathbf{u}}_k$. The Lipschitz and convex functions $g_i(\mathbf{x}_k, \mathbf{u}_k), i = 1, \dots, p$ are from (11).

C. Sequential Convex Programming with Nonlinear Dynamics Constraints

In contrast with prior work, the contribution of this paper is to show that the numerical integration of nonlinear dynamics equations between each SCP iteration is essential to the optimality of the SCP solutions that use sequential linearizations.

The solution ($\mathbf{x}_{k_0:T}, \mathbf{u}_{k_0:T-1}$) of the (w)-th iteration of the following convex programming approximation of the nonconvex program in Problem 2 is denoted as $\mathbf{x}_{k+1} = \mathbf{x}_{k+1}^{(w)}, \mathbf{u}_k = \mathbf{u}_k^{(w)}, k = k_0, \dots, T-1$:

Method 1 Sequential Convex Programming with Linearized Constraints (SCP Method)

- 1: $\mathbf{x}_k^{(0)} :=$ feasible solution to Problem 3, $\mathbf{u}_k^{(0)} := \mathbf{0}$, $\forall k$
 - 2: $\bar{\mathbf{x}}_k := \mathbf{x}_k^{(0)}$, $\bar{\mathbf{u}}_k := \mathbf{u}_k^{(0)}$, $\forall k$
 - 3: $w := 1$
 - 4: **while** $\|\mathbf{x}_k^{(w)} - \mathbf{x}_k^{(w-1)}\| < \epsilon \forall k$ **do**
 - 5: $\mathbf{x}_k^{(w)}, \mathbf{u}_k^{(w)} :=$ the solution to Problem 3, $SCP(\bar{\mathbf{x}}, \bar{\mathbf{u}}) \forall k$
 - 6: $\bar{\mathbf{x}}_k := \mathbf{x}_k^{(w)}$, $\bar{\mathbf{u}}_k := \mathbf{u}_k^{(w)}$, $\forall k$
 - 7: $w := w + 1$
 - 8: **end while**
 - 9: $(\bar{\mathbf{x}}_k^{(w-1)}, \bar{\mathbf{u}}_k^{(w-1)})$ is the approximate solution to Problem 2
-

Problem 4 ((w) -th Sequential Convex Program: $SCP_n^{(w)}(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ given $\bar{\mathbf{x}}_{n,k_0:T} = \mathbf{x}_{n,k_0:T}^{(w-1)}$, $\bar{\mathbf{u}}_{k_0:T-1} = \mathbf{u}_{k_0:T-1}^{(w-1)}$).

$$\underset{\mathbf{u}_{k_0:T-1}}{\text{minimize}} \quad \sum_{k=k_0}^{T-1} \mathcal{F}_u(\mathbf{u}_k) \nu_k \quad \text{subject to} \quad (16)$$

$$\mathbf{x}_{k+1} - A(\bar{\mathbf{x}}_{n,k})\mathbf{x}_k - B(\bar{\mathbf{u}}_k)\mathbf{u}_k - z(\bar{\mathbf{x}}_{n,k}, \bar{\mathbf{u}}_k) = \mathbf{0}, \quad \mathbf{x}_{k_0} = \mathbf{x}_0, \quad k = k_0, \dots, T-1 \quad (17)$$

$$g_i(\bar{\mathbf{x}}_k, \mathbf{u}_k) + \sum_{j=k_0}^{k-1} \mathcal{L}_{i,j} \|\mathbf{u}_j - \bar{\mathbf{u}}_j\| \leq 0, \quad i = 1, \dots, p, \quad k = k_0, \dots, T \quad (18)$$

$$\|\mathbf{u}_k - \bar{\mathbf{u}}_k\| \leq (\beta)^{w-1} \mathcal{T}_0, \quad \forall k = k_0, \dots, T-1 \quad (19)$$

where $A(\bar{\mathbf{x}}_{n,k}) = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}_k} \right|_{(\bar{\mathbf{x}}_{n,k}, \bar{\mathbf{u}}_k)}$, $B(\bar{\mathbf{u}}_k) = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{u}_k} \right|_{(\bar{\mathbf{x}}_{n,k}, \bar{\mathbf{u}}_k)}$, and $z(\bar{\mathbf{x}}_{n,k}, \bar{\mathbf{u}}_k) = \mathbf{f}(\bar{\mathbf{x}}_{n,k}, \bar{\mathbf{u}}_k) - A(\bar{\mathbf{x}}_{n,k})\bar{\mathbf{x}}_{n,k} - B(\bar{\mathbf{u}}_k)\bar{\mathbf{u}}_k$. The Lipschitz and convex functions $g_i(\bar{\mathbf{x}}_k, \mathbf{u}_k)$, $i = 1, \dots, p$ are from (11), and a positive constant $\mathcal{L}_{i,j}$ is defined in Section IV. Moreover, the nominal trajectories $\bar{\mathbf{x}}_k = \mathbf{x}_{n,k}^{(w-1)}$ for the current (w)-th SCPn iteration is obtained by integrating the original nonlinear dynamics (2) starting from the initial condition (\mathbf{x}_0) using the input trajectory $(\mathbf{u}_k^{(w-1)})$, $\forall k$ of the previous $(w-1)$ -th SCPn iteration:

$$\mathbf{x}_{n,k+1}^{(w-1)} = \mathbf{f}_k(\mathbf{x}_{n,k}^{(w-1)}, \mathbf{u}_k^{(w-1)}), \quad k = k_0, \dots, T-1, \quad \text{and} \quad \mathbf{x}_{n,k_0}^{(w-1)} = \mathbf{x}_0^{(w-1)} = \mathbf{x}_0 \quad (20)$$

This SCPn optimization along with nonlinear dynamic correction (20) is repeated until the sequence of trajectories converges. The nominal trajectory $\bar{\mathbf{x}}_{n,k+1}^{(w)}$, $\forall k$ for the $(w+1)$ -th SCPn is obtained by integrating $\bar{\mathbf{x}}_{n,k+1}^{(w)} = \mathbf{f}_k(\bar{\mathbf{x}}_{n,k}^{(w)}, \mathbf{u}_k^{(w)})$, $\bar{\mathbf{x}}_{n,k_0}^{(w)} = \mathbf{x}_{k_0}^{(w)}$ using the (w) -th SCPn solution $\mathbf{u}_{k_0:T-1}^{(w)}$, similar to (20). The convergence of SCPn solutions is proven in Proposition 1 by exploiting the trust region (19) that is exponentially shrinking at a rate $\beta \in (0, 1)$ from the initial size \mathcal{T}_0 .

The SCPn method is described in Method 2. First, an initial approximate trajectory is generated with or without state and control constraints (line 1). Then, the iterative process begins with the agent solving for its optimal state and control trajectories (line 5) and numerically integrating the control trajectory to get the nominal, nonlinear state trajectory for the next iteration (line 7). Finally, iteration is continued until the trajectories converge and satisfy all the constraints (line 4).

IV. Convergence and Optimality of SCPn

In this section we will show that SCPn (Method 2) converges to a point, which satisfies the KKT conditions of the nonconvex optimization in Problem 2. First, we will show that SCPn converges as the number of iterations goes to infinity ($w \rightarrow \infty$).

Proposition 1 (Convergence of SCPn Trajectories). *The error at the k -th time corrected by nonlinear integration (20) for the $(w+1)$ -th SCPn is given as $\bar{\mathbf{x}}_{n,k}^{(w+1)} - \bar{\mathbf{x}}_k^{(w+1)}$, while the accuracy of the linearized dynamics (17) for the $(w+1)$ -th SCPn is $\bar{\mathbf{x}}_k^{(w+1)} - \bar{\mathbf{x}}_k^{(w)}$. Since the trust region $\|\mathbf{u}_k^{(w+1)} - \mathbf{u}_k^{(w)}\| \leq \beta^w \mathcal{T}_0$, $\forall k$ (19) is shrinking exponentially fast as $w \rightarrow \infty$, the following holds for $k = k_0, \dots, T$,*

$$\lim_{w \rightarrow \infty} \|\bar{\mathbf{x}}_k^{(w+1)} - \bar{\mathbf{x}}_k^{(w)}\| = 0, \quad \lim_{w \rightarrow \infty} \|\bar{\mathbf{x}}_{n,k}^{(w+1)} - \bar{\mathbf{x}}_{n,k}^{(w)}\| = 0, \quad \lim_{w \rightarrow \infty} \|\bar{\mathbf{x}}_{n,k}^{(w+1)} - \bar{\mathbf{x}}_k^{(w+1)}\| = 0. \quad (21)$$

Method 2 Sequential Convex Programming with Nonlinear Dynamic Correction (SCPn Method)

- 1: $\mathbf{x}_k^{(0)} :=$ feasible solution to Problem 4, $\mathbf{u}_k^{(0)} := \mathbf{0}$, $\forall k$
 - 2: $\bar{\mathbf{x}}_k := \mathbf{x}_k^{(0)}$, $\bar{\mathbf{u}}_k := \mathbf{u}_k^{(0)}$, $\forall k$
 - 3: $w := 1$
 - 4: **while** $\|\mathbf{x}_k^{(w)} - \mathbf{x}_{n,k}^{(w-1)}\| < \epsilon \forall k$, (18) and (19) hold **do**
 - 5: $\mathbf{x}_k^{(w)}, \mathbf{u}_k^{(w)} :=$ the solution to Problem 4, $SCPn(\bar{\mathbf{x}}, \bar{\mathbf{u}}) \forall k$
 - 6: $\mathbf{x}_{n,k}^{(w)} := \mathbf{f}(\mathbf{x}_0, \mathbf{u}_k^{(w)})$, $\forall k$
 - 7: $\bar{\mathbf{x}}_k := \mathbf{x}_{n,k}^{(w)}$, $\bar{\mathbf{u}}_k := \mathbf{u}_k^{(w)}$, $\forall k$
 - 8: $w := w + 1$
 - 9: **end while**
 - 10: $(\mathbf{x}_{n,k}^{(w-1)}, \mathbf{u}_k^{(w-1)})$ is the approximate solution to Problem 2
-

Proof. Equation (17) for the $(w + 1)$ -th SCP becomes

$$\mathbf{x}_{k+1}^{(w+1)} - \mathbf{x}_{n,k+1}^{(w)} = A(\mathbf{x}_{n,k}^{(w)})(\mathbf{x}_k^{(w+1)} - \mathbf{x}_{n,k}^{(w)}) + B(\mathbf{u}_k^{(w)})(\mathbf{u}_k^{(w+1)} - \mathbf{u}_k^{(w)}) \quad (22)$$

whose fixed initial condition $\mathbf{x}_{k_0}^{(w+1)} = \mathbf{x}_{n,k_0}^{(w)}$ leads to

$$\mathbf{x}_k^{(w+1)} - \mathbf{x}_{n,k}^{(w)} = \sum_{j=k_0}^{k-2} \left(\prod_{i=j+1}^{k-1} A(\mathbf{x}_{n,k+j-i}^{(w)}) \right) B(\mathbf{u}_j^{(w)})(\mathbf{u}_j^{(w+1)} - \mathbf{u}_j^{(w)}) + B(\mathbf{u}_{k-1}^{(w)})(\mathbf{u}_{k-1}^{(w+1)} - \mathbf{u}_{k-1}^{(w)}) \quad (23)$$

Hence, due to $\|\mathbf{u}_k^{(w+1)} - \mathbf{u}_k^{(w)}\| \rightarrow 0$, $\forall k$ as $w \rightarrow \infty$, $\|\mathbf{x}_k^{(w+1)} - \mathbf{x}_{n,k}^{(w)}\| \rightarrow 0$, $\forall k$ is proven.

The error corrected by nonlinear integration at the $(w + 1)$ -th SCPn and $(k + 1)$ -th time is given as

$$\mathbf{x}_{n,k+1}^{(w+1)} - \mathbf{x}_{k+1}^{(w+1)} = \mathbf{x}_{n,k+1}^{(w+1)} - \mathbf{x}_{n,k+1}^{(w)} - \left(\mathbf{x}_{k+1}^{(w+1)} - \mathbf{x}_{n,k+1}^{(w)} \right) = \mathbf{f}_k(\mathbf{x}_{n,k}^{(w+1)}, \mathbf{u}_k^{(w+1)}) - \mathbf{f}(\mathbf{x}_{n,k}^{(w)}, \mathbf{u}_k^{(w)}) - \left(\mathbf{x}_{k+1}^{(w+1)} - \mathbf{x}_{n,k+1}^{(w)} \right) \quad (24)$$

If $\mathbf{f}(\mathbf{x}_{n,k}^{(w)}, \mathbf{u}_k^{(w)})$ has bounded, continuous partial derivatives in the convex domain (18-19), $\mathbf{f}(\mathbf{x}_{n,k}^{(w)}, \mathbf{u}_k^{(w)})$ is Lipschitz [12, p. 89-91]:

$$\|\mathbf{x}_{n,k+1}^{(w+1)} - \mathbf{x}_{n,k+1}^{(w)}\| = \|\mathbf{f}_k(\mathbf{x}_{n,k}^{(w+1)}, \mathbf{u}_k^{(w+1)}) - \mathbf{f}(\mathbf{x}_{n,k}^{(w)}, \mathbf{u}_k^{(w)})\| \leq \|\bar{A}\| \|\mathbf{x}_{n,k}^{(w+1)} - \mathbf{x}_{n,k}^{(w)}\| + \|\bar{B}\| \|\mathbf{u}_k^{(w+1)} - \mathbf{u}_k^{(w)}\| \quad (25)$$

where $\|\bar{A}\| = \sup_{(\mathbf{x}_k, \mathbf{u}_k) \in \mathcal{D}} \|A(\mathbf{x}_k)\|$, $\|\bar{B}\| = \sup_{(\mathbf{x}_k, \mathbf{u}_k) \in \mathcal{D}} \|B(\mathbf{u}_k)\|$, and \mathcal{D} is the convex domain defined in Problems 2 or 4. This can be expressed as a function of $\|\mathbf{u}_k^{(w+1)} - \mathbf{u}_k^{(w)}\|$ using $\mathbf{x}_{n,k_0}^{(w+1)} = \mathbf{x}_{n,k_0}^{(w)}$ as follows

$$\|\mathbf{x}_{n,k}^{(w+1)} - \mathbf{x}_{n,k}^{(w)}\| \leq \sum_{j=k_0}^{k-1} (\|\bar{A}\|)^{k-j-1} \|\bar{B}\| \|\mathbf{u}_j^{(w+1)} - \mathbf{u}_j^{(w)}\| \quad (26)$$

Hence, $\lim_{w \rightarrow \infty} \|\mathbf{x}_{n,k}^{(w+1)} - \mathbf{x}_{n,k}^{(w)}\| = 0$ as $\lim_{w \rightarrow \infty} \|\mathbf{u}_k^{(w+1)} - \mathbf{u}_k^{(w)}\| = 0$. Also, the first equality of (24) verifies $\|\mathbf{x}_{n,k}^{(w+1)} - \mathbf{x}_k^{(w+1)}\| \leq \|\mathbf{x}_{n,k}^{(w+1)} - \mathbf{x}_{n,k}^{(w)}\| + \|\mathbf{x}_k^{(w+1)} - \mathbf{x}_{n,k}^{(w)}\|$. Therefore, $\lim_{w \rightarrow \infty} \|\mathbf{x}_{n,k}^{(w+1)} - \mathbf{x}_k^{(w+1)}\| = 0$ holds due to the limit results mentioned immediately below (23) and (26). \square

A tighter bound can be found by implementing a trust region around the state trajectories: $\|\mathbf{x}_k^{(w+1)} - \mathbf{x}_{n,k}^{(w)}\| \leq \beta \mathcal{T}_0$:

$$\|\mathbf{x}_{n,k}^{(w+1)} - \mathbf{x}_{n,k}^{(w)}\| \leq \sum_{j=k_0}^{k-1} (\|\bar{A}\|)^{k-j} \beta \mathcal{T}_0 + (\|\bar{A}\|)^{k-j-1} \|\bar{B}\| \|\mathbf{u}_j^{(w+1)} - \mathbf{u}_j^{(w)}\| \quad (27)$$

Definition 1 (\mathcal{OS} and \mathcal{FS} of $SCPn^{(w)}(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ and NCP). We define $SCPn^{(w)}(\bar{\mathbf{x}}, \bar{\mathbf{u}})$ as the (w) -th Problem 4 where the nominal trajectories $\bar{\mathbf{x}}_k$ and $\bar{\mathbf{u}}_k$ are used in (17), (18), and (19). An optimal solution and a feasible solution to $SCPn(\bar{\mathbf{x}}, \bar{\mathbf{u}})$

are denoted by $OS(SCPn(\bar{\mathbf{x}}, \bar{\mathbf{u}}))$ and $\mathcal{FS}(SCPn(\bar{\mathbf{x}}, \bar{\mathbf{u}}))$, respectively. For example, the (w) -th and the $(w + 1)$ -th SCPn optimal solutions of Problem 4 yield

$$\begin{aligned} (\mathbf{x}_{k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)}) &= OS(SCPn^{(w)}(\mathbf{x}_{n,k_0:T}^{(w-1)}, \mathbf{u}_{k_0:T-1}^{(w-1)})) \\ (\mathbf{x}_{k_0:T}^{(w+1)}, \mathbf{u}_{k_0:T-1}^{(w+1)}) &= OS(SCPn^{(w+1)}(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})) \end{aligned} \quad (28)$$

Similarly, $OS(NCP)$ and $\mathcal{FS}(NCP)$ denote an optimal solution and a feasible solution of the nonconvex program in Problem 2, respectively.

As described in Algorithm 2, the optimal solution of $SCPn^{(w)}(\mathbf{x}_{n,k_0:T}^{(w-1)}, \mathbf{u}_{k_0:T-1}^{(w-1)})$ is numerically integrated to obtain $\mathbf{x}_{n,k_0:T}^{(w)}$. This nonlinear trajectory is then used as the nominal trajectory for the next SCPn iteration, $SCPn^{(w+1)}(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})$ and can be shown to be a feasible solution to the nonconvex problem.

Proposition 2 A nominal trajectory, $(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})$, that is a feasible solution to the NCP (Problem 2) is also a feasible solution to the $(w + 1)$ -th iteration of SCPn.

Proof. It follows that $(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})$ is a feasible solution to $SCPn^{(w+1)}(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})$ because substituting $(\mathbf{x}_k = \mathbf{x}_{n,k}^{(w)}, \mathbf{u}_k = \mathbf{u}_k^{(w)})$ into (17) and (18) for the $(w + 1)$ -th SCPn straightforwardly shows that (17) and (18) reduce to the corresponding constraints in Problem 2: (10) and (11). Also, the trust region condition (19) automatically holds. \square

In summary, starting from some feasible solution $\mathcal{FS}(NCP)$, we establish

$$(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)}) = \mathcal{FS}(NCP) \quad (29)$$

$$(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)}) = \mathcal{FS}(SCPn^{(w+1)}(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})) \quad (30)$$

$$(\mathbf{x}_{k_0:T}^{(w+1)}, \mathbf{u}_{k_0:T-1}^{(w+1)}) = OS(SCPn^{(w+1)}(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})) \quad (31)$$

These relationships can also be seen in Figure 2.

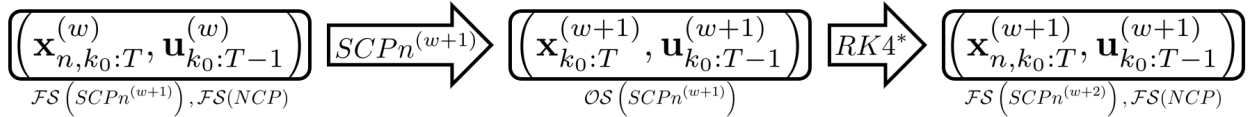


Fig. 2 Relationship between SCPn components

Note that $\lim_{w \rightarrow \infty} \|\mathbf{u}_k - \mathbf{u}_k^{(w-1)}\| = 0$ from (19) and Proposition 1 implies that the inequality constraint function $g_i(\mathbf{x}_k, \mathbf{u}_k) + \sum_{j=k_0}^{k-1} \mathcal{L}_{i,j} \|\mathbf{u}_j - \mathbf{u}_j^{(w-1)}\|$ in (18) increases as w increases, thereby expanding the size of the feasible region that tends toward that of (11). The following theorem shows the cost further decreases through a sequence of convex optimization and nonlinear integration when the inequality constraint (11) of Problems 2 and 4 is restricted to functions of \mathbf{u}_k only, as follows:

<p>Problem 5 (Restricted Non-Convex Program (R-NCP) with Nonlinear Dynamics).</p> $\text{minimize}_{\mathbf{u}_{k_0:T-1}} \sum_{k=k_0}^{T-1} \mathcal{F}_u(\mathbf{u}_k) v_k \quad \text{subject to} \quad (32)$ $\mathbf{x}_{k+1} - \mathbf{f}_k(\mathbf{x}_k, \mathbf{u}_k) = \mathbf{0}, \quad (33)$ $\mathbf{x}_{k_0} = \mathbf{x}_0, \quad k = k_0, \dots, T-1$ $g_i(\mathbf{u}_k) \leq 0, \quad i = 1, \dots, p, \quad k = k_0, \dots, T \quad (34)$	<p>Problem 6 (Restricted (w)-th Sequential Convex Program: $R\text{-SCPn}^{(w)}(\bar{\mathbf{x}}, \bar{\mathbf{u}})$).</p> $\text{minimize}_{\mathbf{u}_{k_0:T-1}} \sum_{k=k_0}^{T-1} \mathcal{F}_u(\mathbf{u}_k) v_k \quad \text{subject to} \quad (35)$ $\mathbf{x}_{k+1} - A(\bar{\mathbf{x}}_{n,k})\mathbf{x}_k - B(\bar{\mathbf{u}}_k)\mathbf{u}_k - z(\bar{\mathbf{x}}_{n,k}, \bar{\mathbf{u}}_k) = \mathbf{0}, \quad (36)$ $\mathbf{x}_{k_0} = \mathbf{x}_0, \quad k = k_0, \dots, T-1$ $g_i(\mathbf{u}_k) \leq 0, \quad i = 1, \dots, p, \quad k = k_0, \dots, T \quad (37)$
---	---

Theorem 1 (Decreasing Cost over Optimal SCPn Sequence with Restricted Constraints). *If there exists a feasible solution to the restricted nonconvex problem (Problem 5 such that $(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)}) = \mathcal{FS}(NCP)$, then*

$$J(\mathbf{u}_{k_0:T-1}^{(w+1)}) \leq J(\mathbf{u}_{k_0:T-1}^{(w)}) \quad (38)$$

where $J(\mathbf{x}, \mathbf{u})$ is the cost function (16) of Problems 5 and 6.

Proof. If the state and control trajectory at some w is a feasible solution to the restricted nonconvex problem, expressed as $(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)}) = \mathcal{FS}(\text{R-NCP})$, then the solution must satisfy the constraints of the restricted NCP, (33) and (34). This feasible solution to R-NCP is used as a nominal trajectory for the $(w+1)$ -th R-SCPn (Problem 6) to yield an optimal solution: $(\mathbf{x}_{k_0:T}^{(w+1)}, \mathbf{u}_{k_0:T-1}^{(w+1)}) = \mathcal{OS}(\text{SCPn}^{(w+1)}(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)}))$.

Consequently, applying Proposition 2 and $J(\mathcal{OS}(\text{SCPn}^{(w+1)})) \leq J(\mathcal{FS}(\text{SCPn}^{(w+1)}))$, $\forall w$ to (30) and (31) results in

$$J(\mathbf{u}_{k_0:T-1}^{(w+1)}) \leq J(\mathbf{u}_{k_0:T-1}^{(w)}) \quad (39)$$

where strict inequality is used unless $(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)}) = \mathcal{OS}(\text{SCPn}(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)}))$. Since the nonlinear integration does not change the input trajectory, the above relationship holds across SCPn iterations. Furthermore, this optimal input solution $(\mathbf{u}_{k_0:T-1}^{(w+1)})$ of SCPn^(w+1) (29) is used to determine the nonlinear trajectory (20) from the initial condition $\mathbf{x}_{n,k_0}^{(w+1)} = \mathbf{x}_0$, thereby yielding the new nominal trajectory $(\mathbf{x}_{n,k_0:T}^{(w+1)}, \mathbf{u}_{k_0:T-1}^{(w+1)})$ as seen in Figure 2. We show herein that this solution is a feasible solution to Problem 2. $(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})$ already satisfies the nonlinear dynamics constraint (33) and because the restricted inequality constraints are independent of the state, satisfying (34) is the same as satisfying (37). Therefore, the constraints are satisfied and the solution is feasible to the restricted nonconvex problem. We conclude that given that $(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})$ is a feasible solution to the R-NCP, when R-SCPn is applied, $(\mathbf{x}_{n,k_0:T}^{(w+1)}, \mathbf{u}_{k_0:T-1}^{(w+1)})$ is also a feasible solution to the R-NCP. \square

Next, we show that the same results can be obtained for the unrestricted problems (2 and 4) provided that a modification is made to the inequality constraint of SCPn.

Theorem 2 (Decreasing Cost over Optimal SCPn Sequence). *If there exists a feasible solution to the original nonconvex problem (Problem 2) such that $(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)}) = \mathcal{FS}(\text{NCP})$, $\exists w$, then*

$$J(\mathbf{u}_{k_0:T-1}^{(w+1)}) \leq J(\mathbf{u}_{k_0:T-1}^{(w)}) \quad (40)$$

where $J(\mathbf{u})$ is the cost function (16) of Problems 2 and 4. under the following condition for each $g_i(\mathbf{x}_k, \mathbf{u}_k)$ in (18)

$$\mathcal{L}_{i,j} = 2\|\bar{B}_k\|(\|\bar{A}_k\|)^{k-j-1} \sup_{(\mathbf{x}_k, \mathbf{u}_k) \in \mathcal{D}} \left\| \frac{\partial g_i(\mathbf{x}_k, \mathbf{u}_k)}{\partial \mathbf{x}_k} \right\| \quad (41)$$

The resulting nominal trajectory $(\mathbf{x}_{n,k_0:T}^{(w+1)}, \mathbf{u}_{k_0:T-1}^{(w+1)})$ obtained from SCPn is also a feasible solution to the nonconvex problem.

Proof. This theorem starts with some value of w in which there is a feasible solution to the original nonconvex problem (Problem 2) such that $(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)}) = \mathcal{FS}(\text{NCP})$. This means that $\mathbf{x}_{n,k+1}^{(w)} = \mathbf{f}_k(\mathbf{x}_{n,k}^{(w)}, \mathbf{u}_k^{(w)})$ from (10) and $\mathbf{g}(\mathbf{x}_{n,k}^{(w)}, \mathbf{u}_k^{(w)}) \leq \mathbf{0}$ (11) hold for all appropriate values of k . Also, by Proposition 2, $(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})$ is also a feasible solution of SCPn^(w+1).

Therefore, applying $J(\mathcal{OS}(\text{SCPn}^{(w+1)})) \leq J(\mathcal{FS}(\text{SCPn}^{(w+1)}))$, $\forall w$ to (30) and (31) results in

$$J(\mathbf{u}_{k_0:T-1}^{(w+1)}) \leq J(\mathbf{u}_{k_0:T-1}^{(w)}) \quad (42)$$

Furthermore, this optimal input solution $(\mathbf{u}_{k_0:T-1}^{(w+1)})$ of SCPn^(w+1) (29) is used to integrate $\mathbf{x}_{n,k+1}^{(w+1)} = \mathbf{f}_k(\mathbf{x}_{n,k}^{(w+1)}, \mathbf{u}_k^{(w+1)})$, $k = k_0, \dots, T-1$ (20) from the initial condition $\mathbf{x}_{n,k_0}^{(w+1)} = \mathbf{x}_0$, thereby yielding the new nominal trajectory $(\mathbf{x}_{n,k_0:T}^{(w+1)}, \mathbf{u}_{k_0:T-1}^{(w+1)})$. We show herein that this solution is a feasible solution to Problem 2 if (41) holds.

First, the convex inequality constraint (18) of SCPn^(w+1) is given as

$$g_i(\mathbf{x}_k^{(w+1)}, \mathbf{u}_k^{(w+1)}) + \sum_{j=k_0}^{k-1} \mathcal{L}_{i,j} \|\mathbf{u}_j^{(w+1)} - \mathbf{u}_j^{(w)}\| \leq 0, \quad i = 1, \dots, p, \quad k = k_0, \dots, T \quad (43)$$

The first-order condition of a convex function holds for each g_i as follows

$$g_i(\mathbf{x}_k^{(w+1)}, \mathbf{u}_k^{(w+1)}) \geq g_i(\mathbf{x}_{n,k}^{(w+1)}, \mathbf{u}_k^{(w+1)}) + \frac{\partial g_i}{\partial \mathbf{x}_k} \Big|_{(\mathbf{x}_{n,k}^{(w+1)}, \mathbf{u}_k^{(w+1)})} (\mathbf{x}_k^{(w+1)} - \mathbf{x}_{n,k}^{(w+1)}) + \frac{\partial g_i}{\partial \mathbf{u}_k} \Big|_{(\mathbf{x}_{n,k}^{(w+1)}, \mathbf{u}_k^{(w+1)})} (\mathbf{0}) \quad (44)$$

Combining (43) and (44) results in

$$g_i(\mathbf{x}_{n,k}^{(w+1)}, \mathbf{u}_k^{(w+1)}) \leq \frac{\partial g_i}{\partial \mathbf{x}_k} \Big|_{(\mathbf{x}_{n,k}^{(w+1)}, \mathbf{u}_k^{(w+1)})} (\mathbf{x}_{n,k}^{(w+1)} - \mathbf{x}_k^{(w+1)}) - \sum_{j=k_0}^{k-1} \mathcal{L}_{i,j} \|\mathbf{u}_j^{(w+1)} - \mathbf{u}_j^{(w)}\| \quad (45)$$

In order to show $(\mathbf{x}_{n,k_0:T}^{(w+1)}, \mathbf{u}_{k_0:T-1}^{(w+1)}) = \mathcal{FS}(\text{NCP})$, we need to prove $g_i(\mathbf{x}_{n,k}^{(w+1)}, \mathbf{u}_k^{(w+1)}) \leq \mathbf{0}$, whose sufficient condition can be given as

$$\left\| \frac{\partial g_i}{\partial \mathbf{x}_k} \right\| (\|\mathbf{x}_{n,k}^{(w+1)} - \mathbf{x}_{n,k}^{(w)}\| + \|\mathbf{x}_k^{(w+1)} - \mathbf{x}_{n,k}^{(w)}\|) \leq \sum_{j=k_0}^{k-1} \mathcal{L}_{i,j} \|\mathbf{u}_j^{(w+1)} - \mathbf{u}_j^{(w)}\| \quad (46)$$

Applying a property of submultiplicativity of norms to (23) shows that both $\|\mathbf{x}_{n,k}^{(w+1)} - \mathbf{x}_{n,k}^{(w)}\|$ and $\|\mathbf{x}_k^{(w+1)} - \mathbf{x}_{n,k}^{(w)}\|$ possess the same upper-bound given in (26). Hence, substituting (26) into (46) shows that (46) is satisfied by

$$\sum_{j=k_0}^{k-1} 2 \left\| \frac{\partial g_i}{\partial \mathbf{x}_k} \right\| (\|\bar{A}\|)^{k-j-1} \|\bar{B}\| \|\mathbf{u}_j^{(w+1)} - \mathbf{u}_j^{(w)}\| = \sum_{j=k_0}^{k-1} \mathcal{L}_{i,j} \|\mathbf{u}_j^{(w+1)} - \mathbf{u}_j^{(w)}\| \quad (47)$$

This bound is overly conservative in practice due to the large value of $\|\bar{A}\|$ in certain dynamics so the local Lipschitz constants are used to re-derive (26), where $\|\bar{A}_k\| = \sup_{\|\mathbf{x}_k - \mathbf{x}_k^{(w)}\| \leq \epsilon} \|A(\mathbf{x}_k)\|$, $\|\bar{B}_k\| = \sup_{\|\mathbf{u}_k - \mathbf{u}_k^{(w)}\| \leq \epsilon} \|B(\mathbf{u}_k)\|$. This is then substituted into (46) as above.

Consequently, the condition of $\mathcal{L}_{i,j}$ (41) is established. Since $(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})$ already satisfies the nonlinear dynamics constraint (10), we conclude that given that $(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})$ is a feasible solution to the NCP, when SCPn is applied, $(\mathbf{x}_{n,k_0:T}^{(w+1)}, \mathbf{u}_{k_0:T-1}^{(w+1)})$ is also a feasible solution to the NCP as can be seen in Figure 2. \square

This theorem shows that a sequence of SCPn optimal solutions $(\mathbf{x}_{k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})$ and feasible solutions $(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})$ to NCP (Problem 2) has a nonincreasing cost. We will now prove that a sequence of optimal solutions exists and converges to an optimal solution (KKT point) of Problem 2.

Theorem 3 (Convergence of SCPn to KKT Point). *If $(\mathbf{x}^{(w)}, \mathbf{u}^{(w)})$ is a feasible solution to Problem 2 for some w_0 , then a sequence of optimal solutions $(\{\mathbf{x}^{(w)}\}, \{\mathbf{u}^{(w)}\})$ exists. If each optimal solution is unique, the sequence converges to $(\mathbf{x}^{(\infty)}, \mathbf{u}^{(\infty)})$, which is a KKT point of Problem 2.*

Proof. Since equations (10)-(11) form a closed and bounded set and the feasible solutions in the sequence $(\mathbf{x}_{n,k_0:T}^{(w)}, \mathbf{u}_{k_0:T-1}^{(w)})$ satisfy the equations, there exists an infinite subsequence $(\{\mathbf{x}^{(w_i)}\}, \{\mathbf{u}^{(w_i)}\})$ that converges. Let the convergence point be called $(\mathbf{x}^{(\infty)}, \mathbf{u}^{(\infty)})$. The Weierstrass theorem [13] establishes that a continuous function over a closed and bounded set achieves a minimum and a maximum on that set, so $J(\mathbf{u}^{(\infty)})$ exists in the set of feasible solutions. By completeness, this gives the sequence $J(\mathbf{u}^{(w)}) \rightarrow J(\mathbf{u}^{(\infty)})$.

The mapping $M(\mathbf{x})$ is equivalent to solving the KKT conditions of SCPn (\mathbf{x}, \mathbf{u}) , which are continuous with respect to \mathbf{x} . Therefore, the mapping M is continuous. Since the subsequence $(\{\mathbf{x}^{(w_i)}\}, \{\mathbf{u}^{(w_i)}\}) \rightarrow (\mathbf{x}^{(\infty)}, \mathbf{u}^{(\infty)})$ and the limit of a continuous function of a convergent sequence is the function of the limit of that sequence, the following is true:

$$\{M(\mathbf{x}^{(w_i)})\} \rightarrow M(\mathbf{x}^{(\infty)}) \quad (48)$$

Additionally, $\mathbf{x}^{(\infty)}$ is a fixed point and $\mathbf{x}^{(w_i+1)} = M(\mathbf{x}^{(w_i)})$. Therefore,

$$\{\mathbf{x}^{(w_i+1)}\} \rightarrow \mathbf{x}^{(\infty)} \quad (49)$$

Finally, we will show that $(\mathbf{x}^{(\infty)}, \mathbf{u}^{(\infty)}) = (\mathbf{x}_n^{(\infty)}, \mathbf{u}^{(\infty)})$ from Proposition 1 is a KKT point of Problem 2. Since $\mathbf{x}^{(\infty)}$ is a fixed point of M , it is a solution to SCPn $(\mathbf{x}^{(\infty)}, \mathbf{u}^{(\infty)})$ and from Proposition 2, it is a feasible solution to Problem 2.

Additionally, Problem 4 is convex so any solution to this problem is a KKT point $(\mathbf{x}^{(\infty)}, \mathbf{u}^{(\infty)}) = \text{SCPn}(\mathbf{x}_n^{(\infty)}, \mathbf{u}^{(\infty)})$ and satisfies stationarity (50), complementary slackness (51), and dual feasibility (52):

$$\begin{aligned} \begin{pmatrix} \mathbf{0} \\ \mathbf{0} \end{pmatrix} &= \begin{bmatrix} \nabla_{\mathbf{x}_k} J(\mathbf{x}^{(\infty)}, \mathbf{u}^{(\infty)}) \\ \nabla_{\mathbf{u}_k} J(\mathbf{x}^{(\infty)}, \mathbf{u}^{(\infty)}) \end{bmatrix} + \begin{bmatrix} \sum_{i=1}^P \lambda_{k,i}^{(\infty)} \nabla_{\mathbf{x}_k} g_i(\mathbf{x}_k^{(\infty)}, \mathbf{u}_k^{(\infty)}) \\ \sum_{i=1}^P \lambda_{k,i}^{(\infty)} \nabla_{\mathbf{u}_k} g_i(\mathbf{x}_k^{(\infty)}, \mathbf{u}_k^{(\infty)}) \end{bmatrix} + \begin{bmatrix} \nabla_{\mathbf{x}_k} \mathbf{h}_{\text{dyn}}(\mathbf{x}_{k+1}^{(\infty)}, \mathbf{x}_k^{(\infty)}, \mathbf{u}_k^{(\infty)}) \\ \nabla_{\mathbf{u}_k} \mathbf{h}_{\text{dyn}}(\mathbf{x}_{k+1}^{(\infty)}, \mathbf{x}_k^{(\infty)}, \mathbf{u}_k^{(\infty)}) \end{bmatrix} \boldsymbol{\mu}_k^{(\infty)} \\ &\quad + \begin{bmatrix} \nabla_{\mathbf{x}_k} \mathbf{h}_{\text{dyn}}(\mathbf{x}_k^{(\infty)}, \mathbf{x}_{k-1}^{(\infty)}, \mathbf{u}_{k-1}^{(\infty)}) \\ \mathbf{0} \end{bmatrix} \boldsymbol{\mu}_{k-1}^{(\infty)}, \quad k = k_0, \dots, T-1 \end{aligned} \quad (50)$$

$$0 = \lambda_{k,i}^{(\infty)} g_i(\mathbf{x}_k^{(\infty)}, \mathbf{u}_k^{(\infty)}), \quad i = 1, \dots, p, \quad k = k_0, \dots, T \quad (51)$$

$$\lambda_{k,i}^{(\infty)} \geq 0, \quad i = 1, \dots, p, \quad k = k_0, \dots, T \quad (52)$$

where we used $\sum_{j=k_0}^{k-1} \mathcal{L}_{i,j} \|\mathbf{u}_j - \mathbf{u}_j^{(w-1)}\| \rightarrow 0$ due to the exponentially shrinking trust region (19). Also, we can find from (17)

$$\mathbf{h}_{\text{dyn}}(\mathbf{x}_{k+1}^{(w)}, \mathbf{x}_k^{(w)}, \mathbf{u}_k^{(w)}) = A(\mathbf{x}_{n,k}^{(w-1)})\mathbf{x}_k^{(w)} + B(\mathbf{u}_k^{(w-1)})\mathbf{u}_k^{(w)} + z(\mathbf{x}_{n,k}^{(w-1)}, \mathbf{u}_k^{(w-1)}) - \mathbf{x}_{k+1}^{(w)} \quad (53)$$

$$\nabla_{\mathbf{x}_k} \mathbf{h}_{\text{dyn}}(\mathbf{x}_{k+1}^{(\infty)}, \mathbf{x}_k^{(\infty)}, \mathbf{u}_k^{(\infty)}) = A(\mathbf{x}_{n,k}^{(\infty)}), \quad \nabla_{\mathbf{x}_k} \mathbf{h}_{\text{dyn}}(\mathbf{x}_k^{(\infty)}, \mathbf{x}_{k-1}^{(\infty)}, \mathbf{u}_{k-1}^{(\infty)}) = -\mathbf{I}, \quad \nabla_{\mathbf{u}_k} \mathbf{h}_{\text{dyn}}(\mathbf{x}_{k+1}^{(\infty)}, \mathbf{x}_k^{(\infty)}, \mathbf{u}_k^{(\infty)}) = B(\mathbf{u}_{n,k}^{(\infty)})$$

Proposition 3 and Theorem 2 indicate that the original cost function (9) of Problem 2 is expressed as $J(\mathbf{u})$ for the fixed KKT point $(\mathbf{x}_n^{(\infty)}, \mathbf{u}^{(\infty)}) = \mathcal{FS}(\text{SCPn}(\mathbf{x}_n^{(\infty)}, \mathbf{u}^{(\infty)})) = \mathcal{FS}(\text{NCP})$. Now let $\lambda_k^* = \lambda_k^{(\infty)}$, $\mathbf{v}_k^* = \mathbf{v}_k^{(\infty)}$, $\boldsymbol{\mu}_{\mathbf{x}_k}^* = \boldsymbol{\mu}_{\mathbf{x}_k}^{(\infty)}$, and $\boldsymbol{\mu}_{\mathbf{u}_k}^* = \boldsymbol{\mu}_{\mathbf{u}_k}^{(\infty)}$, where the Lagrange multipliers with subscript ∞ are the KKT multipliers that satisfy the KKT conditions for the convex program. Then, we can show that (50)-(52) become the KKT conditions for the nonconvex program (Problem 2) because $\nabla_{\mathbf{x}_k} \mathbf{h}_{\text{dyn}}(\mathbf{x}_{k+1}^{(\infty)}, \mathbf{x}_k^{(\infty)}, \mathbf{u}_k^{(\infty)}) = A(\mathbf{x}_{n,k}^{(\infty)}) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}_{n,k}^{(\infty)}}$ and $\nabla_{\mathbf{u}_k} \mathbf{h}_{\text{dyn}}(\mathbf{x}_{k+1}^{(\infty)}, \mathbf{x}_k^{(\infty)}, \mathbf{u}_k^{(\infty)}) = \frac{\partial \mathbf{f}}{\partial \mathbf{u}_{n,k}^{(\infty)}}$, $k = k_0, \dots, T-1$. Hence, all the KKT conditions (50)-(52) remain the same for Problem 2 after substitution. \square

Now, we can relax the cost function to include convex functions of the state trajectory. In the following proposition, it is shown that cost function for each iteration can be expressed solely in terms of the control input trajectory for that iteration.

Proposition 3 (Convexity of Cost Function of Input). *The cost function (16) can be written as a convex function of only $\mathbf{u}_{k_0:T-1}^{(w)} = (\mathbf{u}_{k_0}^{(w)}; \dots; \mathbf{u}_{T-1}^{(w)})$ such that*

$$J(\mathbf{u}_{k_0:T-1}^{(w)}) = \sum_{k=k_0}^{T-1} \left(\mathcal{F}_x(\mathbf{x}_k^{(w)}(\mathbf{u}_k^{(w)})) + \mathcal{F}_u(\mathbf{u}_k^{(w)}) \right) v_k \quad (54)$$

Proof. Nesting Equation (17) in on itself for $k = k_0, \dots, T$ results in:

$$\begin{aligned} \mathbf{x}_{k+1}^{(w)} &= \left(\prod_{j=k_0}^k A(\mathbf{x}_{n,j}^{(w-1)}) \right) \mathbf{x}_{k_0}^{(w)} + \sum_{j=k_0}^{k-1} \left(\prod_{i=j+1}^k A(\mathbf{x}_{n,i}^{(w-1)}) \right) \left(B(\mathbf{u}_j^{(w-1)})\mathbf{u}_j^{(w)} + z(\mathbf{x}_{n,j}^{(w-1)}, \mathbf{u}_j^{(w-1)}) \right) \\ &\quad + B(\mathbf{u}_k^{(w-1)})\mathbf{u}_k^{(w)} + z(\mathbf{x}_{n,k}^{(w-1)}, \mathbf{u}_k^{(w-1)}) \\ &= \left(\prod_{j=k_0}^k A(\mathbf{x}_{n,j}^{(w-1)}) \right) \mathbf{x}_{k_0}^{(w)} + \sum_{j=k_0}^{k-1} \left(\prod_{i=j+1}^k A(\mathbf{x}_{n,i}^{(w-1)}) \right) \left(B(\mathbf{u}_j^{(w-1)})\mathbf{u}_j^{(w)} - \mathbf{u}_j^{(w-1)} + \mathbf{x}_{n,j+1}^{(w-1)} - A(\mathbf{x}_{n,j}^{(w-1)})\mathbf{x}_{n,j}^{(w-1)} \right) \\ &\quad + B(\mathbf{u}_k^{(w-1)})\mathbf{u}_k^{(w)} - \mathbf{u}_k^{(w-1)} + \mathbf{x}_{n,k+1}^{(w-1)} - A(\mathbf{x}_{n,k}^{(w-1)})\mathbf{x}_{n,k}^{(w-1)} \\ &= \mathbf{b}_k^{(w)}(\mathbf{x}_{n,k_0:k}^{(w-1)}, \mathbf{u}_{k_0:k}^{(w-1)}) + \left[\mathbf{C}_k^{(w)}(\mathbf{x}_{n,k_0:k}^{(w-1)}, \mathbf{u}_{k_0:k}^{(w-1)}) \right] \mathbf{u}_{k_0:k}^{(w)} \end{aligned} \quad (55)$$

where the initial condition $\mathbf{x}_{k_0}^{(w)}$ is fixed as given in (4). The variables $\mathbf{b}_k^{(w)}$ and $\mathbf{C}_k^{(w)}$ are functions of solutions from previous iterations, therefore $\mathbf{x}_k^{(w)}$ is an affine function of $\mathbf{u}_k^{(w)}$. Since $\mathcal{F}_x(\mathbf{x}_k^{(w)})$ in (16) is a convex function, a convex function of an affine function is convex, so the entire cost function can be expressed as a convex function of $\mathbf{u}_k^{(w)}$. \square

Example 3 (Convexification of Collision Avoidance Constraint). [4] To use convex programming to solve the trajectory optimization (Problem 2), the nonconvex collision avoidance constraint must be converted to a convex constraint. The best convex approximations of the collision avoidance constraints will be affine constraints. In other words, the 2-D circle (\mathbb{S}^1) or 3-D sphere (\mathbb{S}^2) which defines the prohibited region is replaced by a line (2-D) or plane (3-D) which is tangent to the circle or sphere and perpendicular to the line segment connecting the nominal positions ($\bar{\mathbf{x}}_j$) of the agents or obstacles.

$$(\bar{\mathbf{x}}_{j,k} - \bar{\mathbf{x}}_{i,k})^T G^T G (\mathbf{x}_{j,k} - \bar{\mathbf{x}}_{i,k}) \geq R_{\text{col}} \|G(\bar{\mathbf{x}}_{j,k} - \bar{\mathbf{x}}_{i,k})\|_2 \quad k = k_0, \dots, T, \quad i \quad (57)$$

The addition of the barrier $\sum_{j=k_0}^{k-1} \mathcal{L}_{i,j} \|\mathbf{u}_j - \bar{\mathbf{u}}_j\|$ around the inequality constraints (18) is imposed to ensure conformity to the nonlinear dynamics, but it reduces the feasible set such that the number of iterations required to find a convergent solution is significantly higher than the previous SCP implementation when used with tough convex or convexified constraints like Example 3. To reduce the computational burden, a combined approach (Method 3) was formulated wherein the original SCP method is used until the solutions violate the constraints of the nonlinear problem, then the SCPn method is used. This method is only advisable when the inequality constraints are lenient enough that the added $\mathcal{L}_{i,j}$ term is not needed for several iterations in the trajectory generation process. The relationship between the components of the M-SCPn method can be seen in Figure 3.

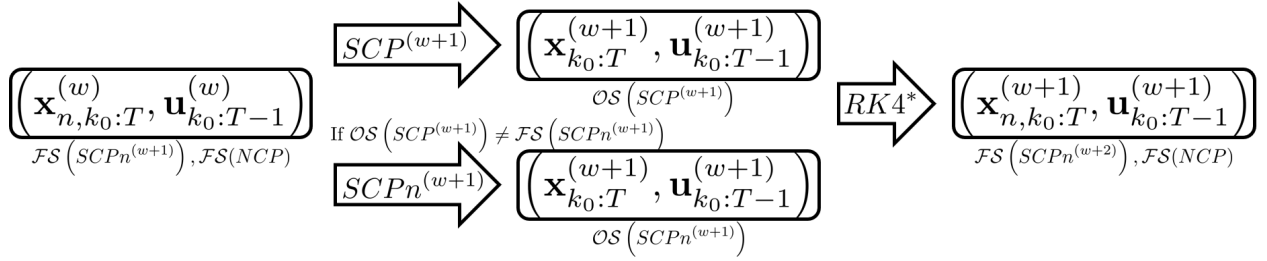


Fig. 3 Relationship between M-SCPn components

Method 3 Modified Sequential Convex Programming with Nonlinear Dynamic Correction (M-SCPn Method)

- 1: $\mathbf{x}_k^{(0)} :=$ the solution to Problem 3 $\forall k$
 - 2: $\bar{\mathbf{x}}_k := \mathbf{x}_k^{(0)}, \forall k$
 - 3: $w := 1$
 - 4: **while** $\|\mathbf{x}_k^{(w)} - \mathbf{x}_{n,k}^{(w-1)}\| < \epsilon \forall k$ **do**
 - 5: $\mathbf{x}_k^{(w)}, \mathbf{u}_k^{(w)} :=$ the solution to Problem 3 (Sequential Convex Program), $\forall k$
 - 6: **if** (18) and (19) do not hold **then**
 - 7: $\mathbf{x}_k^{(w)}, \mathbf{u}_k^{(w)} :=$ the solution to Problem 4 (Nonlinear Corrected Sequential Convex Program), $\forall k$
 - 8: **end if**
 - 9: $\mathbf{x}_{n,k}^{(w)} := \mathbf{f}(\mathbf{x}_0, \mathbf{u}_k^{(w)}), \forall k$
 - 10: $\bar{\mathbf{x}}_k := \mathbf{x}_{n,k}^{(w)}, \forall k$
 - 11: $w := w + 1$
 - 12: **end while**
 - 13: $\mathbf{x}_{n,k}^{(w-1)}$ is the approximate solution to Problem 2
-

Next, we will prove the stability of the combined method.

Theorem 4 (Convergence of M-SCPn). *If $(\mathbf{x}_{n,k}^{(w)}, \mathbf{u}_k^{(w)})$ is a feasible solution to Problem 2 for some w , then the M-SCPn Method (Algorithm 3) will converge to a steady-state solution $(\mathbf{x}_{n,k}^\infty, \mathbf{u}_k^\infty)$ which is feasible to the nonconvex problem.*

Proof. There are two possible ways the M-SCPn method can execute. Either SCP returns a solution that is feasible to SCPn, or SCPn executes and returns the solution. In either case, the resulting solution is feasible to SCPn then numerically integrated. From Theorem 2 we see that a solution of this sort must also be feasible to the nonconvex problem, Problem 2. From Theorem 3 we know that the feasible set of the NCP is closed and bounded and thus the solutions of

M-SCPn form a sequence which converges to a point $(\mathbf{x}_k^{(\infty)}, \mathbf{u}_k^{(\infty)})$. To prove that the cost converges, we must first show that it decreases over M-SCPn iterations. There are four possible logical paths through the method that must be addressed:

- 1) $SCP^{(w)}$ solution is the nominal trajectory for $SCP^{(w+1)}$ (proven in prior work [5])
- 2) $SCP^{(w)}$ solution is the nominal trajectory for $SCPn^{(w+1)}$
- 3) $SCPn^{(w)}$ solution is the nominal trajectory for $SCP^{(w+1)}$
- 4) $SCPn^{(w)}$ solution is the nominal trajectory for $SCPn^{(w+1)}$ (proven in Theorem 2)

Note that each nominal trajectory is numerically integrated and the proof in prior work remains valid for this logical path because the numerical integration does not affect the input trajectory.

For Case 2, the numerically integrated $SCP^{(w)}$ solution is used as the nominal trajectory for $SCPn^{(w+1)}$, so substituting in to (17), (18), and (19) shows that the SCPn constraints simplify to the NCP constraints which are satisfied by $SCP^{(w)}$. Therefore, $SCP^{(w)}$ is a feasible solution of $SCPn^{(w+1)}$. Since an optimal solution must have lower cost than a feasible solution, the cost must decrease in Case 2.

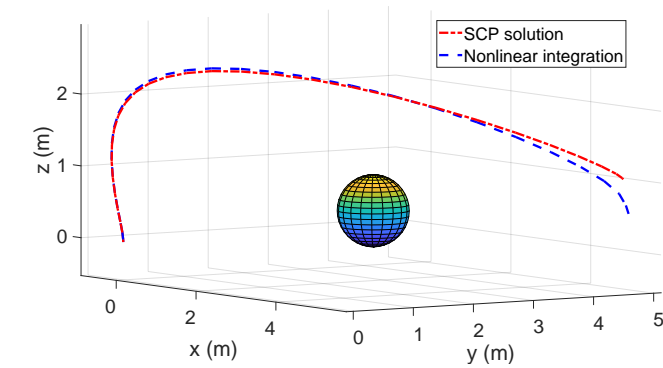
For Case 3, $SCPn^{(w)}$ is feasible to the NCP and the inequality constraints are the same in NCP and SCP. The dynamics constraint is satisfied because substituting the $SCPn^{(w)}$ trajectory into the linearized dynamics equation for $SCP^{(w+1)}$ simplifies to the nonlinear dynamics equation. So $SCPn^{(w)}$ is a feasible solution of $SCP^{(w+1)}$. Since an optimal solution must have lower cost than a feasible solution, the cost must also decrease in Case 3.

Since the cost $J(\mathbf{u}^{(w)})$ is continuous and decreasing over a closed and bounded set, it converges by the Weierstrass theorem [13] to $J(\mathbf{u}^{(\infty)})$ as established in Theorem 2. □

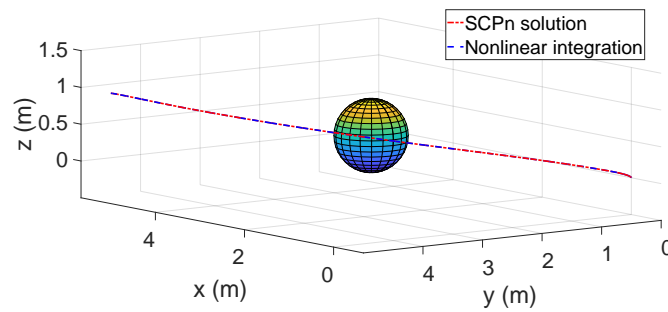
V. Simulation Results

Simulations were performed to validate the SCPn and M-SCPn methods compared to SCP and the true nonlinear trajectory. All simulations were performed using quadrotor dynamics in an environment with a single obstacle using the collision avoidance constraint described in Example 3 and initial and terminal constraints. Simulations used CVX, a MATLAB-based convex optimization solver [14, 15]. The nonlinear trajectories were found by numerically integrating the nonlinear dynamics, as seen in 2, using the optimal control trajectory found in SCP, SCPn, and M-SCPn. Each simulation was given the initial position of 0, 0, 0 and the terminal position of 5, 5, 1 with a 1-m diameter sphere in the middle of the straight-line path. These simulations were repeated with varying convergence tolerances and initial nominal trajectories to determine sensitivity to these parameters.

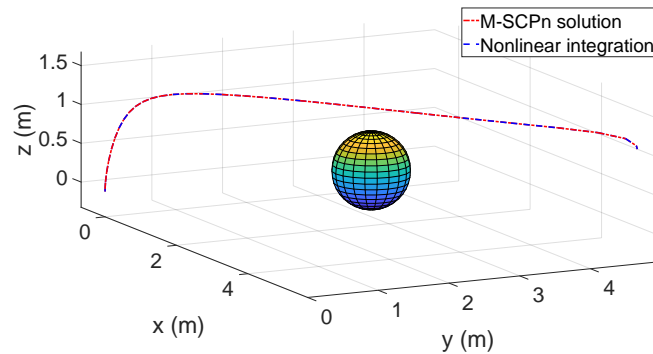
Results of the simulations are shown in Figure 4. Figure 4a shows that the SCP trajectory diverges from the nonlinear dynamics over the course of the trajectory to such an extent that the terminal constraint is not satisfied in the nonlinear dynamics. Figure 4b shows the trajectory for both the SCPn and nonlinear cases swooping out of the page around the sphere. The M-SCPn trajectory shown in Figure 4c looks much closer to the SCP trajectory than the SCPn trajectory, but does not diverge from the nonlinear dynamics. Both SCPn and M-SCPn trajectories satisfy the nonlinear dynamics for the full trajectory including the terminal position.



(a) SCP Optimal vs Nonlinear Trajectory



(b) SCPn Optimal vs Nonlinear Trajectory



(c) M-SCPn Optimal vs Nonlinear Trajectory

Fig. 4 Quadrotor Trajectories computed using SCP (a), SCPn (b), and M-SCPn (c) compared against the nonlinear trajectory using the respective control trajectory. The SCP trajectory clearly does not follow the nonlinear dynamics, but the SCPn and M-SCPn trajectories do.

To test the performance of the methods, the convergence tolerance and the initial nominal trajectories were varied. The convergence tolerance specifies at what point the method is considered converged based on the difference between the state and control trajectories. The different nominal trajectories used were determined by the number of iterations used to generate the nominal trajectories, with more iterations equating to better initial guesses. Initial results from these sensitivity simulations indicated that SCPn is quite sensitive to the initial nominal trajectory given and could not perform unless the seed trajectory was very consistent with the nonlinear dynamics. In general, SCPn needed the most iterations to converge except when given the best initial nominal trajectory. For the smallest two tolerances, both SCPn and M-SCPn could not converge before the maximum number of iterations was reached. As expected, SCP has the lowest computation time, with M-SCPn coming in second for most combinations of initial nominal trajectories and tolerances. However, SCPn does significantly outperform M-SCPn when the initial nominal trajectory is favorable.

More study is needed to draw definitive trends.

VI. Conclusion

Two methods were presented which are capable of extending sequential convex programming for use with nonlinear dynamics. The methods, SCPn and M-SCPn, numerically integrate the control trajectory resulting from the optimization to obtain a corrected nominal trajectory, which the next iteration of the optimization then linearized the dynamics around. This way, the linearization and discretization errors do not compound in each successive iteration, keeping the obtained trajectory close to the nonlinear dynamic trajectory. The convergence and optimality of the SCPn method was shown. Simulation results using highly nonlinear quadrotor dynamics show that both SCPn and M-SCPn solve the terminal constraint failure of SCP and yields trajectories which adhere to the nonlinear dynamics, but suffer a performance loss when compared to SCP due to the extra computation needed to enforce the nonlinear dynamics. SCPn performed significantly better when given a well-chosen initial nominal trajectory which nearly satisfied the nonlinear dynamics, without satisfying the other constraints. More theoretical study and simulation is needed to certify the sensitivity results and determine the true performance of both methods. Further work remains to reduce the computational burden and implement the method on an agile quadrotor platform.

Acknowledgements

The work of Rebecca Foust was supported by a NASA Space Technology Research Fellowship and in part by the Jet Propulsion Laboratory (JPL). Government sponsorship is acknowledged. The authors also thank Amir Rahmani and Christian Chilan for their technical input.

References

- [1] Fahroo, F., and Ross, I. M., "Advances in pseudospectral methods for optimal control," *AIAA guidance, navigation and control conference and exhibit*, 2008, p. 7309.
- [2] Rao, A. V., "A survey of numerical methods for optimal control," *Advances in the Astronautical Sciences*, Vol. 135, No. 1, 2009, pp. 497–528.
- [3] Richards, A., Schouwenaars, T., How, J. P., and Feron, E., "Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 4, 2002, pp. 755–764.
- [4] Morgan, D., Chung, S.-J., and Hadaegh, F. Y., "Model Predictive Control of Swarms of Spacecraft Using Sequential Convex Programming," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 6, 2014, pp. 1725–1740.
- [5] Morgan, D., Subramanian, G. P., Chung, S.-J., and Hadaegh, F. Y., "Swarm Assignment and Trajectory Optimization Using Variable-Swarm, Distributed Auction Assignment and Sequential Convex Programming," *The International Journal of Robotics Research*, 2016.
- [6] Foust, R., Chung, S.-J., and Hadaegh, F. Y., "Real-Time Optimal Control And Target Assignment For Autonomous In-Orbit Satellite Assembly From A Modular Heterogeneous Swarm," *26th AAS/AIAA Space Flight Mechanics Meeting, Napa, CA*, February 14-18, 2016.
- [7] Foust, R., Chung, S.-J., and Hadaegh, F., "Autonomous In-Orbit Satellite Assembly from a Modular Heterogeneous Swarm using Sequential Convex Programming," *AIAA/AAS Astrodynamics Specialist Conference, AIAA SPACE and Astronautics Forum and Exposition, Long Beach, CA*, Sept. 13-16, 2016.
- [8] Bandyopadhyay, S., Baldini, F., Foust, R., Chung, S.-J., Rahmani, A., de la Croix, J.-P., and Hadaegh, F. Y., "Distributed Fast Motion Planning for Spacecraft Swarms in Cluttered Environments Using Spherical Expansions and Sequence of Convex Optimization Problems," *Proc. 9th International Workshop on Satellite Constellations and Formation Flying (IWSCFF)*, Boulder, Colorado, June 19-21, 2017.
- [9] Gong, Q., Fahroo, F., and Ross, I. M., "Spectral algorithm for pseudospectral methods in optimal control," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 3, 2008, pp. 460–471.
- [10] Ross, I. M., and Fahroo, F., "Legendre Pseudospectral Approximations of Optimal Control Problems," *Lecture Notes in Control and Information Systems*, Vol. 295, 2003, pp. 327–342.

- [11] Garg, D., Patterson, M., Hager, W. W., Rao, A. V., Benson, D. A., and Huntington, G. T., “A unified framework for the numerical solution of optimal control problems using pseudospectral methods,” *Automatica*, Vol. 46, No. 11, 2010, pp. 1843–1851.
- [12] Khalil, H., *Nonlinear Systems*, 3rd ed., Prentice Hall, 2002.
- [13] Bertsekas, D. P., *Nonlinear programming*, Athena scientific, 1999.
- [14] Grant, M., and Boyd, S., “CVX: Matlab Software for Disciplined Convex Programming, version 2.1,” <http://cvxr.com/cvx>, Mar. 2014.
- [15] Grant, M., and Boyd, S., “Graph implementations for nonsmooth convex programs,” *Recent Advances in Learning and Control*, edited by V. Blondel, S. Boyd, and H. Kimura, Lecture Notes in Control and Information Sciences, Springer-Verlag Limited, 2008, pp. 95–110. http://stanford.edu/~boyd/graph_dcp.html.