Constrained Autonomous Satellite Docking via Differential Flatness and Model Predictive Control

Samira S. Farahani* Ivan Papusha† Catharine McGhan* Richard M. Murray†

***Abstract*— We investigate trajectory generation algorithms that allow a satellite to autonomously rendezvous and dock with a target satellite to perform maintenance tasks, or transport the target satellite to a new operational location. We propose different path planning strategies for each of the phases of rendezvous. In the first phase, the satellite navigates to a point in the Line of Sight (LOS) region of the target satellite. We show that the satellite's equations of motion are differentially flat in the relative coordinates, hence the rendezvous trajectory can be found efficiently in the flat output space without a need to integrate the full nonlinear dynamics. In the second phase, we use model predictive control (MPC) with linearized dynamics to navigate the spacecraft to the final docking location within a constrained approach envelope. We demonstrate feasibility of this study by simulating a sample docking mission.**

I. Introduction

Rendezvous and docking are used in several mission types, such as manned spaceflight, resupply, assembly, servicing and repair, and refueling [1], [2], [3], [4], [5], [6]. In all these missions, the concept of operations is very similar: the rendezvous phase refers to the approach of one spacecraft to another at a relative distance of 10 km to 100 m; the docking phase refers to the final approach maneuvers executed to engage docking ports, less than 100 m from the target spacecraft; and the docked phase refers to the control of the rigidly-attached spacecraft pair, such as orbital repositioning. In most cases, the target spacecraft is passively performing station-keeping while the chaser spacecraft actively handles the rendezvous and docking process.

Generally, the goal in a rendezvous and docking problem is to minimize fuel or propellant use, as fuel is a spacecraft's most valuable resource—since the initial fuel supply is constrained by launch vehicle lift capabilities, and satellites are generally not able to replenish fuel in orbit, the satellite's mission ends when the fuel is exhausted. Successful proximity operations also rely heavily on the availability and accuracy of relative measurements: radar and GPS can be used in the far-field, laser range finders are only valid within about 1 km relative distance to the target spacecraft, and stereo cameras and other visual navigation are generally only available within about 100 m of the target spacecraft. Various approaches, such as Kalman filtering and sensor fusion, have been proposed to deal with the measurement problem [7], [8], [9].

Our work addresses a portion of the benchmark rendezvous problem of relocating a broken spacecraft [10]. In this scenario, a defunct satellite or piece of space debris, referred to as the "target," occupies a given circular orbit slot. Such debris takes up valuable space, can collide with other objects, or can exit the orbital slot and become a hazard. To prevent this, a tug spacecraft, referred to as the "chaser," will autonomously dock to the target and act as a propulsion system that relocates the target to a new operational "graveyard" position.

Our focus in this paper is on autonomously navigating the chaser to the docking position. To this end, we split the rendezvous problem into two different phases, and use different control approaches in each phase: boundary value control in the far-field, and model predictive control (MPC) for close-in planning. During the first phase, trajectories outside the line of sight (LOS) cone are generated efficiently by exploiting the differential flatness of the relative frame equations of motion, and an effective lack of state constraints far from the final docking location. Until the chaser enters the LOS region, this initial phase (10 km to 100 m) of rendezvous corresponds to navigating the spacecraft to specified points along the trajectory.

In the second phase, once the spacecraft is inside the LOS region (100 m to 0 m), we use MPC to navigate the spacecraft to the final docking location. MPC has been successfully applied in the past to rendezvous and navigation of satellites [11], [12], [13], and is an appropriate control strategy in the final phase, because it allows the incorporation of constraints on the relative position, closing velocity, and thrust.

Our simulation results show that a combination of these approaches is effective and efficient. The rest of this paper is organized as follows. In Section II, we present the equations of motion and their transformation to Hill's frame. In Section III, we prove that the equations of motion are differentially flat, and discuss the boundary value problem for navigating the chaser to a location in the LOS region. In Section IV, we present the MPC optimization problem that navigates the chaser from a location in the LOS region to the docking position. We show simulation results in Section V, and conclude in Section VI.

II. Satellite Docking

A. Equations of motion

Let $(r, R) \in SE(3)$ be the inertial position and orientation of the target spacecraft frame, and $r_c \in \mathbb{R}^3$ the inertial position of the chaser. The rotation matrix $R \in SO(3)$ rotates vectors from the target frame to the inertial frame, and R^T rotates them back to the target frame. Assuming the target is

*Department of Computing and Mathematical Sciences, California Institute of Technology, Pasadena, CA.

†Department of Control and Dynamical Systems, California Institute of Technology, Pasadena, CA.

in a uniform circular orbit with radius $\|r\| = a$, the equations of motion in the inertial frame are:

$$\ddot{r} = -n^2 r \tag{1}$$

$$\ddot{r}_c = -\frac{\mu}{\|r_c\|^3} r_c + f_I/m_c \tag{2}$$

$$\dot{R} = R\omega^\wedge, \tag{3}$$

where $n = \sqrt{\mu/a^3}$ is the target angular velocity (mean motion), $\mu = GM$ is the standard gravitational parameter, m_c is the mass of the chaser spacecraft, $\omega = (0, 0, n) \in \mathbb{R}^3$ is the instantaneous body angular velocity of the target frame ($^\wedge$ denotes the cross-product matrix, see Section II-C), and $f_I \in \mathbb{R}^3$ are external inertial forces on the chaser.

B. Relative frame planning

It is convenient to work in relative coordinates. Define $\rho = R^T(r_c - r) = (x, y, z) \in \mathbb{R}^3$ as the relative position of the chaser in the target (Hill's) frame (see Fig. 1). Solving for r_c and substituting into (2), we obtain

$$\ddot{r} + (\ddot{R\rho}) = -\frac{\mu}{\|r + R\rho\|^3}(r + R\rho) + f_I/m_c. \tag{2'}$$

Together, equations (1), (2′), and (3) describe the (nonlinear) orbital dynamics of the target and chaser in terms of the target's inertial orientation (r, R) and the chaser's relative position ρ in the target's frame.

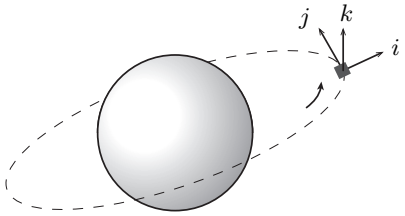

Fig. 1. Rotating reference (Hill's) frame centered at the target platform: radial direction i, parallel to velocity (in-track) j, and out of plane (cross-track) k.

The goal is to select forces, and indirectly thrusts, that ensure the chaser reaches the target, $\rho(t) \to 0$. Because the nonlinear motion is difficult to treat, the Clohessy–Wiltshire–Hill (CWH) linearization is typically used for the constrained docking portion of the proximity operations [14],

$$\begin{cases} \ddot{x} - 2n\dot{y} - 3n^2 x = f_x/m_c \\ \ddot{y} + 2n\dot{x} = f_y/m_c \\ \ddot{z} + n^2 z = f_z/m_c. \end{cases} \tag{4}$$

The CWH approximation holds well only when the target and chaser orbits are very close—over a few orbital periods with target separation no more than about 20 km (see Fig. 2).

As a result of this distance limitation, the CWH equations are only used in the terminal LOS-constrained portion of the docking process, where the state-constrained linear equations of motion allow direct application of model predictive control (MPC). A full nonlinear model is used outside of the LOS cone. We treat the nonlinear equations of motion efficiently using differential flatness, because MPC without modification is not a feasible approach while the chaser is far away from the target, due to the computational limitation caused by having a large prediction horizon. Note that unconstrained control techniques like LQR can only provide a baseline level of performance that, in general, violates the state and input constraints.

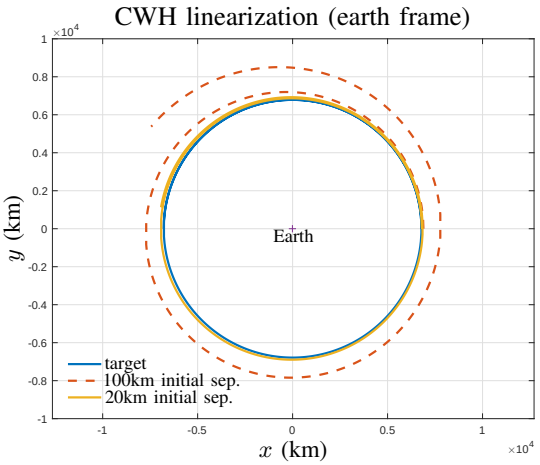

Fig. 2. For a target in circular low earth orbit (415 km) and initial separation of 100 km, the CWH linearization (dashed) departs from the nonlinear dynamics by more than 2000 km over just 1.5 orbital periods. By comparison, an initial target separation of 20 km shows much less departure.

C. Linearization

In this section we provide a modern derivation of the nonlinear dynamics and CWH equations (4) using notation familiar to robotics and manipulation (*e.g.*, [15]). Specifically, the linearization (4) is used for close-in (MPC-based) planning, while the full nonlinear model (6) is used in the far-field. Recall that the cross product matrix is defined as

$$\omega^\wedge \triangleq \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \in SO(3), \quad \omega \in \mathbb{R}^3.$$

Using $\dot{R} = R\omega^\wedge$ and $\ddot{R} = R(\omega^\wedge\omega^\wedge + \dot{\omega}^\wedge)$, we have

$$\begin{aligned} (\ddot{R\rho}) &= \ddot{R}\rho + 2\dot{R}\dot{\rho} + R\ddot{\rho} \\ &= R(\omega \times (\omega \times \rho) + \dot{\omega} \times \rho + 2\omega \times \dot{\rho} + \ddot{\rho}) \\ &= R(\omega \times (\omega \times \rho) + 2\omega \times \dot{\rho} + \ddot{\rho}), \end{aligned} \tag{5}$$

since $\dot{\omega} = 0$ for a uniform circular orbit. We can substitute (5) and (1) into (2′) and split out a factor of $n^2 = \mu/a^3$ to obtain

$$\begin{aligned} &-n^2 r + R(\omega \times (\omega \times \rho) + 2\omega \times \dot{\rho} + \ddot{\rho}) \\ &= -n^2 \left(1 + \frac{\|\rho\|^2}{a^2} + \frac{2r^T R\rho}{a^2}\right)^{-3/2} (r + R\rho) + f_I/m_c. \end{aligned}$$

Next, we multiply both sides by $R^{-1} = R^T$ and use the following substitutions for a circular orbit:

$$R^T r = (a, 0, 0), \quad \rho = (x, y, z), \quad \omega = (0, 0, n).$$

After some simplification, we arrive at the vector equation

$$\begin{bmatrix} \ddot{x} - 2n\dot{y} - n^2 x - n^2 a \\ \ddot{y} + 2n\dot{x} - n^2 y \\ \ddot{z} \end{bmatrix} = -n^2 \left(1 + \frac{x^2+y^2+z^2}{a^2} + \frac{2}{a}x\right)^{-3/2} \begin{bmatrix} x+a \\ y \\ z \end{bmatrix} + \frac{f}{m_c}, \tag{6}$$

where $f \triangleq R^T f_I$ is the target-referenced chaser control forces. Note that the left side is linear in (x, y, z). By linearizing the right side around $(x, y, z) = 0$, we obtain the CWH equations (4).

III. Boundary value planning

Differential flatness was originally studied in [16], extended in [17], and has been used to great success in nonlinear trajectory generation for a variety of mechanical systems, rigid body chains, robotic arms, and quadrotors [18], [19]. Examples of differentially flat systems include controllable linear systems, feedback linearizable systems, and fully actuated mechanical systems. In this work, we exploit the differential flatness of the nonlinear dynamics (6), where the flat output is the target-referenced coordinate $\rho = (x, y, z)$.

In general, a system with state $x \in \mathbb{R}^n$, control input $u \in \mathbb{R}^m$, and (possibly nonlinear) dynamics

$$\dot{x} = f(x, u) \tag{7}$$

is *differentially flat* if there exists an output $y \in \mathbb{R}^m$ of the form

$$y = y(x, u, \dot{u}, \ldots, u^{(p)}),$$

such that the state and control are functions of y and its derivatives,

$$x = x(y, \dot{y}, \ldots, y^{(q)}), \tag{8}$$

$$u = u(y, \dot{y}, \ldots, y^{(q)}). \tag{9}$$

The output y, called the *flat output*, plays a central role in trajectory planning because it determines both the state trajectory and the control input.

A. Boundary value problem

Consider the problem of finding a trajectory $x : [0, T] \to \mathbb{R}^n$ and an open loop control law $u : [0, T] \to \mathbb{R}^m$ satisfying the nonlinear dynamics (7) along with the initial and final constraints

$$x(0) = x_{\mathrm{i}}, \quad x(T) = x_{\mathrm{f}}, \tag{10}$$

where x_{i}, $x_{\mathrm{f}} \in \mathbb{R}^n$ are prescribed initial and final states, and T is a fixed final time. If the mapping (8) between the states x and $(y, \dot{y}, \ldots, y^{(q)})$ is bijective, then the state constraints (10) can be rewritten as

$$\begin{aligned} y(0) &= y_{\mathrm{i}}, & y(T) &= y_{\mathrm{f}}, \\ \dot{y}(0) &= \dot{y}_{\mathrm{i}}, & \dot{y}(T) &= \dot{y}_{\mathrm{f}}, \\ &\vdots & &\vdots \\ y^{(q)}(0) &= y_{\mathrm{i}}^{(q)} & y^{(q)}(T) &= y_{\mathrm{f}}^{(q)}, \end{aligned} \tag{11}$$

where $x_{\mathrm{i}} = x(y_{\mathrm{i}}, \dot{y}_{\mathrm{i}}, \ldots, y_{\mathrm{i}}^{(q)})$ and $x_{\mathrm{f}} = x(y_{\mathrm{f}}, \dot{y}_{\mathrm{f}}, \ldots, y_{\mathrm{f}}^{(q)})$.

B. Trajectory generation

It is common to write the flat output y trajectory as a linear combination

$$y(t) = \sum_{j=1}^{N} \alpha_j \phi_j(t), \tag{12}$$

where for each $j = 1, \ldots, N$, the basis functions $\phi_j : [0, T] \to \mathbb{R}^m$ are known, and the real coefficients $[\alpha_j]$ are to be determined. The boundary value constraints (11) are linear in the coefficient vector $(\alpha_1, \ldots, \alpha_N)$,

$$\begin{bmatrix} y(0) \\ \dot{y}(0) \\ \vdots \\ y^{(q)}(0) \end{bmatrix} = \begin{bmatrix} \phi_1(0) & \phi_2(0) & \cdots & \phi_N(0) \\ \dot{\phi}_1(0) & \dot{\phi}_2(0) & \cdots & \dot{\phi}_N(0) \\ \vdots & & \ddots & \\ \phi_1^{(q)}(0) & \phi_2^{(q)}(0) & \cdots & \phi_N^{(q)}(0) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix}, \tag{13}$$

$$\begin{bmatrix} y(T) \\ \dot{y}(T) \\ \vdots \\ y^{(q)}(T) \end{bmatrix} = \begin{bmatrix} \phi_1(T) & \phi_2(T) & \cdots & \phi_N(T) \\ \dot{\phi}_1(T) & \dot{\phi}_2(T) & \cdots & \dot{\phi}_N(T) \\ \vdots & & \ddots & \\ \phi_1^{(q)}(T) & \phi_2^{(q)}(T) & \cdots & \phi_N^{(q)}(T) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_N \end{bmatrix}. \tag{14}$$

Note that for a given coefficient vector $(\alpha_1, \ldots, \alpha_N)$, we can determine both the state trajectory $x(t)$ and the required control input $u(t)$ by taking successive derivatives of (12) and substituting them into (8) and (9). These derivatives are computed in closed form without numerical loss. Furthermore, any coefficient vector that satisfies the linear constraints (13) and (14), automatically satisfies the initial and final state constraints (10).

C. Fully actuated mechanical system

In the target-referenced coordinate ρ, the docking problem corresponds to a fully actuated mechanical system of the form

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + N(q) = \tau.$$

Note that this mechanical system is differentially flat with flat output $q \triangleq \rho = (x, y, z)$ and torques $\tau = f/m_c$, provided it is fully actuated, see [18, Table 2]. This assumption is satisfied for the docking problem, provided the chaser spacecraft can thrust in any direction, without any constraints on the thrust force magnitude. Given a trajectory $q(t)$, we determine the optimal inputs by solving the nonlinear equations (6) for f/m_c. In practice, constraints on the thrust force magnitude for far-field planning are incorporated, by proxy, by penalizing the final relative acceleration $\ddot{q}(T)$ and the trajectory coefficient norm.

D. Trajectory parameterization

We write the Hill's frame referenced trajectory $q(t) \in \mathbb{R}^3$ as a Fourier series around the orbital period n,

$$q_i(t) = \gamma_i + \sum_{j=1}^{N} \alpha_{ij} \cos(jnt) + \beta_{ij} \sin(jnt), \quad i = 1, 2, 3, \tag{15}$$

where N is the number of Fourier coefficients and $[\gamma_i]$, $[\alpha_{ij}]$, and $[\beta_{ij}]$ are constant coefficients to be determined. With initial and final conditions $q(0)$ and $q(T)$, feasible trajectories are given by coefficients that satisfy the linear equations (15). Note that time derivatives of $q(t)$ are also linear in the coefficients. While other basis functions can be used to parameterize the relative trajectory, we found that a Fourier series around the orbital period works well. Furthermore, actuator bandwidth is incorporated directly by limiting the number of high-frequency coefficients.

IV. Model Predictive Control

Using boundary value optimization, we obtain a trajectory to bring the chaser satellite to a point inside the LOS region. Then, we apply MPC to generate a trajectory that leads to the target or docking position. To obtain the controller using MPC, we discretize the continuous-time equations of motion obtained in (4). For simplicity, we present results for the in-plane 2D dynamical system, ignoring the out-of-plane z coordinate. Moreover, the objective function is defined as a quadratic objective function. Following the description in [11], the LQ MPC optimization problem with linear constraints at each time step k is defined as:

$$\begin{aligned}
\text{min. } & x(k+N_\text{p})^T P x(k+N_\text{p}) + \\
& \sum_{j=0}^{N_\text{p}-1} x(k+j)^T Q x(k+j) + u(k+j)^T R u(k+j) \\
\text{s.t. } & x(k+j+1) = Ax(k+j) + Bu(k+j), \\
& \qquad j = 0, \ldots, N_\text{p} - 1 && (16) \\
& |u_i(k+j)| \le u_\text{max},\ i = 1, 2,\ j = 0, \ldots, N_\text{p} - 1 && (17) \\
& |v_i(k+j)| \le v_\text{max},\ i = 1, 2,\ j = 0, \ldots, N_\text{p} - 1 && (18) \\
& A_\text{LOS} x(k+j) \le b_\text{LOS},\ \ j = 1, \ldots, N_\text{p}, && (19)
\end{aligned}$$

where N_p is the prediction horizon, and $P \in \mathbb{R}^{n \times n}$ is the solution of the Riccati equation related to the LQR problem, used as the terminal cost to enforce stability. The matrices Q and R are appropriately defined positive definite weighting matrices for the quadratic cost function. The constraint (16) refers to the state-space dynamics in which the A and B matrices come from discretizing the CWH equations (4); the constraint (17) defines bounds on each component of the input (thrust force); the constraint (18) defines bounds on the closing velocity; and (19) is the LOS cone constraint,

$$\begin{cases}
\dfrac{\sin(\varphi+\gamma)}{(r_p - r_\text{tol})\sin(\gamma)} x(k) - \dfrac{\cos(\varphi+\gamma)}{(r_p - r_\text{tol})\sin(\gamma)} y(k) \ge 1, \\
-\dfrac{\sin(\varphi-\gamma)}{(r_p - r_\text{tol})\sin(\gamma)} x(k) + \dfrac{\cos(\varphi-\gamma)}{(r_p - r_\text{tol})\sin(\gamma)} y(k) \ge 1, \\
\dfrac{\cos(\varphi)}{r_p \sin(\gamma)} x(k) + \dfrac{\sin(\varphi)}{r_p \sin(\gamma)} y(k) \ge 1,
\end{cases}$$

where φ is the angle between the platform docking port and the x-axis, γ denotes half of the LOS cone angle, r_p is the radius of the target platform (assuming it has a disk shape), and r_tol denotes the distance by which the vertex of the LOS cone is moved inside the platform (chosen to relax the LOS constraint). The first two constraints force the satellite to stay within the LOS cone and the last constraint ensures that the collision of the chaser with the target is avoided with the relaxed cone constraints. For more details regarding the LOS constraints, we refer the reader to [11].

The main reason for using MPC at this stage is to be able to enforce LOS, velocity, and thrust constraints. It is generally true that the longer the prediction horizon N_p, the better the closed-loop performance (though having a very large N_p also increases the computation time). In the next section, we compare MPC to LQR, and show examples where the chaser violates the LOS constraints even after tuning LQR to limit the input magnitude. However the MPC trajectory stays within the LOS cone as required. We also show that for $N_\text{p} = 30$ and a sampling time 0.5 sec, the obtained trajectory converges to the target's docking position.

V. Simulation results

In this section, we show the application of each control method to its associated approach phase, in order to generate a trajectory that leads the chaser satellite to the docking position on the target satellite. We used MATLAB R2014b to simulate the docking paths, and CVX and SDPT3 to solve the optimization problems [20], [21].

A. Phasing in from far away via differential flatness

Outside the LOS cone (> 100 m), the state is unconstrained, therefore we can exploit differential flatness. We can choose to minimize any convex function of the coefficients $[\gamma_i]$, $[\alpha_{ij}]$, and $[\beta_{ij}]$ that satisfy the linear equations (15) at the initial and final times. Two objectives that make sense are minimum final relative acceleration at time T, and minimum weighted coefficient norm. These are given by:

$$\begin{aligned}
J_\text{acc} &= \|\ddot{q}(T)\|_2^2, \\
J_\text{wnorm} &= \sum_{i=1}^{3} \Big(\sum_{j=1}^{N} (jn) \left(|\alpha_{ij}| + |\beta_{ij}| \right) \Big)^2.
\end{aligned}$$

We choose these objectives because total fuel use is not directly a convex function of the variables $[\gamma_i]$, $[\alpha_{ij}]$, and $[\beta_{ij}]$. As a result, fuel is minimized by proxy. For example, J_wnorm was chosen to be an upper bound on maximum relative velocity over the mission timeline, so that $\max_{t \in [0,T]} \|\dot{q}(t)\|_2^2 \le J_\text{wnorm}$. Other objectives are possible, *e.g.*, the fourth derivative (or *snap*), which has been successful in the context of quadrotor control [19]. Here, the final location of the chaser is a point in the LOS cone, which is the initial position of the MPC plan. The location we have chosen inside the LOS cone is $(0.1, -0.01)$ km and the magnitude of the final velocity is set to 1.5 m/s.

In general this method provides inputs suitable for non-impulsive engines, however, experiments show that as the Fourier basis becomes more expressive with larger N, the inputs become more impulsive, which is the expected optimal behavior. Fig. 3 shows the trajectories with $N = 1000$ basis elements, and control input magnitudes for two different mission timelines $T = 2160$ sec, and $T = 4320$ sec. A

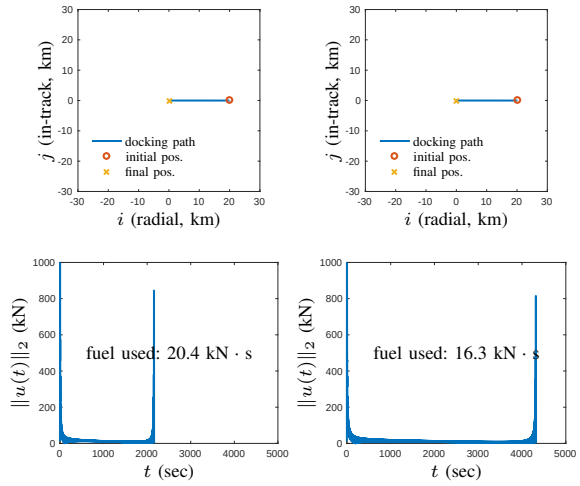

Fig. 3. Flatness-based docking with initial separation of 20km by minimizing maximum relative velocity J_{wnorm} with two different mission lengths: 2160 sec (left column) and 4320 sec (right column). Here, $N = 1000$ coefficients. The inputs become impulsive with higher N.

longer mission timeline results in less fuel use (area under the control magnitude curve), however, the control inputs are not constrained. Though we have not shown this here, it is possible to penalize the control inputs by a weighted objective on the input weights. Alternatively, the inputs can be clipped to their maximum as long as the impulse delivered is the same. We should mention that while the Fourier basis is adequate, a more careful selection of trajectory basis elements and objective weights tuned to a specific rendezvous application, actuator type, and bandwidth, are needed in a more realistic nonlinear path planning application.

B. Close-in docking via MPC

The parameters picked for this example are influenced by [11]. The mass of the chaser is $m_c = 500$ kg, target radius $r_p = 2.5$ m, LOS cone half angle $\gamma = 30°$, tolerance $r_{\text{tol}} = 0.5$ m, and target angular velocity $n = 1.107 \times 10^{-3}$ rad/s. The sampling time for the discrete model is 0.5 sec, and the weighting matrices in the objective function are:

$$Q = 3\begin{bmatrix} 0.1I_{2\times 2} & 0_{2\times 2} \\ 0_{2\times 2} & I_{2\times 2} \end{bmatrix}, \quad R = I_{2\times 2}.$$

The maximum thrust magnitude is $u_{\max} = 0.06$ kN, maximum closing velocity $v_{\max} = 1.5$ m/s, and the prediction horizon is $N_{\text{p}} = 30$. The process noise is i.i.d. Gaussian $w = (w_1, \ldots, w_{n_4}) \sim \mathcal{N}(0, I_{4\times 4})$.

To simulate a departure from the nominal trajectory in Fig. 3, we solve the MPC optimization problem (16)–(19), with initial state

$$x(0) = \begin{bmatrix} 0.1 \\ -0.01 \\ -v_{\max}\cos(\theta) \\ -v_{\max}\sin(\theta) \end{bmatrix}, \quad \theta = \tan^{-1}\frac{x_2(0)}{x_1(0)},$$

i.e., within the LOS cone, just over $\sim$ 100 m away, and pointed toward the target. The MPC maneuver simulation time is 111 sec. Note that the initial state in this phase corresponds to the final state obtained using the flatness approach in the previous section. We compare our results with a simple LQR controller.

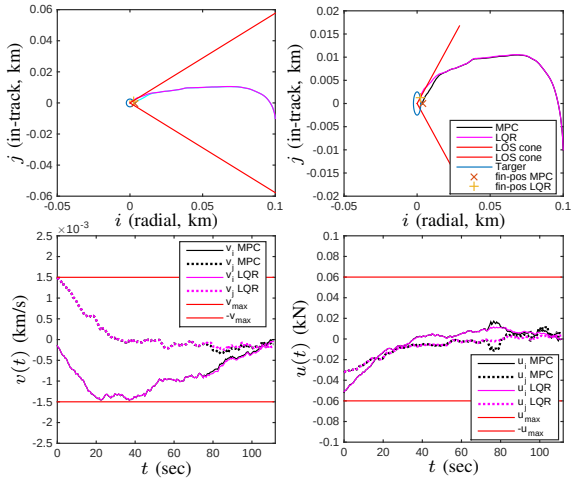

Fig. 4. Trajectory and velocity generated for the chaser and the optimal control sequence using LQR and MPC. The target spacecraft is shown as the center-circle, and the red lines delineate the LOS cone.

The obtained trajectory, relative velocity, and the optimal input sequences using MPC and a baseline LQR approach are shown in Fig. 4. The upper-right plot is a zoomed-in view of the final position of the chaser shown in the upper-left plot. As can be seen in this plot, the trajectory of the chaser using LQR violates the LOS constant while the trajectory provided by MPC remains within the LOS cone. The bottom-left plot shows that both components of the velocity stay within the given bound. The bottom-right plot shows that both components of the input vector u remain within the given bound. The baseline LQR controller has the same Q and R matrices as the MPC controller, but has an infinite horizon, and does not explicitly bound the control input or relative closing velocity.

We also illustrate in Fig. 5 that picking different initial positions for the chaser inside the LOS cone results in a successful trajectory plan using MPC, whereas for LQR, the chaser always ends up in a location outside the LOS cone. For this experiment, we picked 10 different initial states in the proximity of the LOS cone boundary (red lines in Fig. 5). The results of these two sections demonstrate that the combined control approaches, *i.e.*, the flatness approach together with MPC, result in a satisfactory trajectory for the rendezvous and docking mission.

VI. Conclusion

In this paper, we presented trajectory generation techniques for the rendezvous and docking of satellites. To autonomously navigate the chaser, we combined two different control strategies, which switch depending on the position of the chaser with respect to the target. In the first phase, where the chaser is still far away from the target (10 km to 100 m), we found the trajectory by solving a boundary

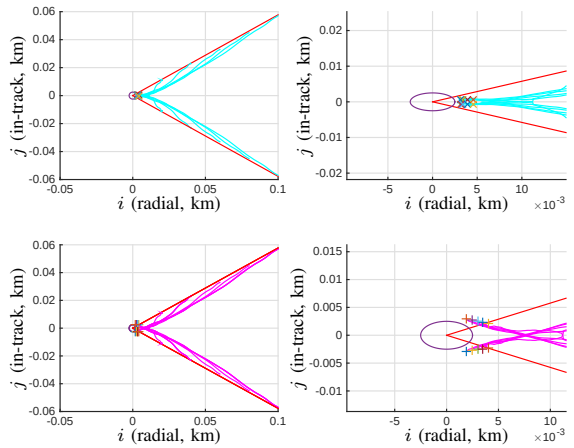

Fig. 5. Trajectories generated for the chaser using LQR and MPC with different initial positions. Target spacecraft is shown as the center-circle, and the red lines delineate the LOS cone. Left-hand side plots show the full approach (top-left: MPC, bottom-left: LQR), right-hand side plots are a zoomed-in view of the final position of the chaser relative to the target (top-right: MPC, bottom-right: LQR).

value problem. We showed that the equations of motion of the chaser satellite are differentially flat in the relative coordinates, and hence we do not need to integrate the nonlinear orbital dynamics. This is an advantage, because even for long time horizons, the full nonlinear trajectory can be obtained by solving a constrained least squares problem. Once the chaser enters the line of sight (LOS) region, we apply model predictive control (MPC) to find the final trajectory that leads the chaser to the docking location. In this close-in phase, we considered the state and input constraints.

Our future research is focused on designing efficient observers and investigating the robustness of the resulting trajectories. In this work, we did not consider measurement noise, and had a process noise model that does not incorporate, *e.g.*, the J_2 perturbation due to the Earth's flattening. Furthermore, our analysis was centered on a target in a uniform circular orbit, ignoring the more general case of an elliptical orbit. Also note that we did not directly constrain the thrust force in the first phase of motion outside of the LOS cone, though an appropriate choice of objective function that penalizes the trajectory coefficients is helpful. It is clear that the mission timeline and interface points between the boundary value problem and the MPC plan become variables over which a grid search must be performed. Future work should also investigate the effect of sensor or actuator failures on the chaser satellite.

Acknowledgments

This work was supported in part by the Boeing company and by the Engineering Resilient Space Systems group.

References

[1] K. Galabova, G. Bounova, O. de Weck, and D. Hastings, "Architecting a family of space tugs based on orbital transfer mission scenarios," in *AIAA SPACE 2003 Conference and Exposition*, Long Beach, California, Sept. 2003, p. 6368.

[2] J. Goodman, "History of space shuttle rendezvous and proximity operations," *Jounal of Spacecraft and Rockets*, vol. 43, no. 5, p. 944959, 2006.

[3] F. Tasker and C. G. Henshaw, "Managing contact dynamics for orbital robotic servicing missions," in *AIAA SPACE 2008 Conference and Exposition*, San Diego, California, Sept. 2008, p. 7908.

[4] M. Polites, "Technology of automated rendezvous and capture in space," *Jounal of Spacecraft and Rockets*, vol. 36, no. 2, p. 280291, 1999.

[5] D. Woffinden and D. Geller, "Navigating the road to autonomous orbital rendezvous," *Jounal of Spacecraft and Rockets*, vol. 44, no. 4, p. 898909, 2007.

[6] D. Zimpfer, P. Kachmar, and S. Tuohy, "Autonomous rendezvous, capture and in-space assembly:past, present and future," in *AIAA Space Exploration Conference*, Orlando, Florida, Jan. 2005, pp. 3433–3440.

[7] M. R. Leinz, C. T. Chen, M. W. Beaven, T. P. Weismuller, D. L. Caballero, W. B. Gaumer, P. W. Sabasteanski, P. A. Scott, and M. A. Lundgren, "Orbital express autonomous rendezvous and capture sensor system (ARCSS) flight test results," in *Proceedings of the AAS/AIAA Astrodynamics Specialists Conference*, Mackinac Island, Michigan, Aug. 2007, pp. AAS Paper 07–408.

[8] R. M. Pinson, R. T. Howard, and A. F. Heaton, "Orbital express advanced video guidance sensor: Ground testing, flight results and comparisons," in *Proceedings of the AAS/AIAA Astrodynamics Specialists Conference*, Honolulu, Hawaii, Aug. 2008, pp. AIAA Paper 08–7318.

[9] R. Zanetti, K. J. DeMars, and R. H. Bishop, "Underweighting nonlinear measurements," *Journal of Guidance, Control and Dynamics*, vol. 22, no. 5, p. 16701675, 2010.

[10] C. Jewinson and R. S. Erwin, "A spacecraft benchmark problem for hybrid control and estimation," in *IEEE Conference on Decision and Control (CDC), submitted*, 2016.

[11] S. D. Cairn, H. Park, and I. Kolmanovsky, "Model predictive control approach for guidance of spacecraft rendezvous and proximity maneuvering," *International Journal of Robust and Nonlinear Control*, vol. 22, p. 13981427, May 2012.

[12] M. Holzinger, J. DiMatteo, J. Schwartz, and M. Milam, "Passively safe receding horizon control for satellite proximity operations," in *IEEE Conference on Decision and Control (CDC)*, Cancun, Mexico, Dec. 2008, p. 34333440.

[13] A. Richards and J. How, "Performance evaluation of rendezvous using model predictive control," in *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Austin, Texas, Aug. 2003, p. AIAA Paper 20035507.

[14] W. H. Clohessy and R. S. Wiltshire, "Terminal guidance system for satellite rendezvous," *Journal of the Aerospace Sciences*, vol. 27, no. 9, pp. 653–658, 1960.

[15] R. M. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. CRC Press, 1994.

[16] M. Fliess, J. Levine, P. Martin, and P. Rouchon, "Flatness and defect of nonlinear systems: Introductory theory and examples," *International Journal of Control*, vol. 61, no. 6, pp. 1327–1361, 1995.

[17] M. van Nieuwstadt, M. Rathinam, and R. M. Murray, "Differential flatness and absolute equivalence of nonlinear control systems," *Journal of Control and Optimization*, vol. 36, no. 4, pp. 1225–1239, 1998.

[18] R. M. Murray, M. Rathinam, and W. Sluis, "Differential flatness of mechanical control systems: A catalog of prototype systems," in *ASME International Mechanical Engineering Congress and Exposition*, San Francisco, CA, Nov. 1995.

[19] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 2011, pp. 2520–2525.

[20] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[21] K. C. Toh, M. J. Todd, and R. H. Tutuncu, "SDPT3—a Matlab software package for semidefinite programming," pp. 545–581, 1999.