

## Model Predictive Control approach for guidance of spacecraft rendezvous and proximity maneuvering<sup>‡</sup>

S. Di Cairano<sup>1</sup>, H. Park<sup>2</sup> and I. Kolmanovsky<sup>2,\*,†</sup>

<sup>1</sup> *Mechatronics, Mitsubishi Electric Research Laboratories<sup>§</sup>, Cambridge, MA, USA*

<sup>2</sup> *Department of Aerospace Engineering, The University of Michigan, Ann Arbor, MI, USA*

### SUMMARY

Traditionally, rendezvous and proximity maneuvers have been performed using *open-loop* maneuver planning techniques and ad hoc error corrections. In this paper, a Model Predictive Control (MPC) approach is applied to spacecraft rendezvous and proximity maneuvering problems in the orbital plane. We demonstrate that various constraints arising in these maneuvers can be effectively handled with the MPC approach. These include constraints on thrust magnitude, constraints on spacecraft positioning within Line-of-Sight cone while approaching the docking port on a target platform, and constraints on approach velocity to match the velocity of the docking port. The two cases of a nonrotating and a rotating (tumbling) platform are treated separately, and trajectories are evaluated in terms of maneuver time and fuel consumption. For the case when the platform is not rotating and the docking port position is fixed with respect to the chosen frame, an explicit offline solution of the MPC optimization problem is shown to be possible; this explicit solution has a form of a piecewise affine control law suitable for online implementation without an on-board optimizer. In the case of a fast rotating platform, it is, however, shown that the prediction of the platform rotation is necessary to successfully accomplish the maneuvers and to reduce fuel consumption. Finally, the proposed approach is applied to debris avoidance maneuvers with the debris in the spacecraft rendezvous path. The significance of this paper is in demonstrating that Model Predictive Control can be an effective *feedback control* approach to satisfy various maneuver requirements, reduce fuel consumption, and provide robustness to disturbances. Copyright © 2012 John Wiley & Sons, Ltd.

Received 2 August 2011; Revised 12 February 2012; Accepted 15 March 2012

KEY WORDS: spacecraft rendezvous and proximity maneuvering; spacecraft guidance; spacecraft docking; constrained control; Model Predictive Control; debris avoidance

### 1. INTRODUCTION

Autonomous spacecraft rendezvous and proximity operations (RPO) are among the most important and difficult elements of modern spacecraft missions. Examples of RPO maneuvers (see [1–3] and references therein) include a transport vehicle approach and docking to the International Space Station, a capture and recovery of a tumbling out-of-control satellite and a fly-by or avoidance of a space object such as debris.

The requirements of RPO maneuvers invariably include the treatment of pointwise-in-time state and control constraints. Examples include the thrust magnitude constraints, constraints on the approaching spacecraft to maintain its position within a Line-of-Sight (LOS) cone emanating from

\*Correspondence to: Ilya Kolmanovsky, Department of Aerospace Engineering, The University of Michigan, Ann Arbor, Michigan, USA.

†E-mail: ilya@umich.edu

<sup>‡</sup>This manuscript is dedicated to Professor David W. Clarke of Oxford University, a gifted researcher and educator, and a role model for our community.

<sup>§</sup>This research was not sponsored by Mitsubishi Electric or its subsidiaries.

the docking port on the target platform (see [4–7] and references therein), and constraints on the terminal translational velocity of the spacecraft to match the velocity of the docking port for soft-docking [8]. The docking port may exhibit complicated motion if the target spacecraft is rotating or tumbling out of control (see [5, 9]). Collisions with debris emerging on the spacecraft path must be avoided during the maneuvers. In addition to satisfying constraints, fuel consumption and maneuver time must be minimized.

Spacecraft rendezvous control problems have received significant attention in the literature. See, for instance, [5, 7, 10–13] and references therein.

The main motivation for this paper is to demonstrate that Model Predictive Control (MPC) can be an attractive *feedback control* approach for RPO maneuvering, which has traditionally been performed via open-loop “ $\Delta v$ ” sequencing (see e.g., [7]). In recent years, some approaches for spacecraft rendezvous and docking based on variants of the MPC framework have been proposed. In [14] the authors proposed an MPC strategy with variable horizon that requires the solution of a mixed-integer linear program at every control cycle. Such a strategy is extended in [5] to generate failure-safe trajectories. The variable horizon approach is further extended in [15] in the so-called “rubber band” MPC, where the MPC controller is designed by inverse optimality by using techniques similar to [16] with a horizon that first maintains a constant number of moves (as in standard MPC) and then decreases as in variable horizon MPC. An application of MPC to spacecraft navigation in proximity of a space station is considered in [17], where an unconstrained MPC is proposed for guidance to the neighborhood of the space station, while the LOS between the station and the spacecraft sensors is maintained by a constrained spacecraft attitude controller, and a control allocation scheme commands the thrusters. In a similar context, in [18], a receding horizon controller requiring the solutions of nonconvex quadratically constrained quadratic programs has been proposed for passively safe proximity operations, where a statistical model of the uncertainty is used for improving robustness with respect to position uncertainty.

The available spacecraft computing power may vary depending on the spacecraft, from that being comparable with a personal computer for high end spacecraft to being significantly more restricted and even less than in automotive applications for low end spacecraft (nanosats or cubesats). In either cases, solutions that require lower computing effort are in demand for spacecraft applications. Any saved capacity can be used to deploy additional control, communications, and fault management functions and/or reduce electric power consumption that is a very significant concern in these applications.

Thus, motivated by considerations of computational feasibility for on-board implementation in orbiting spacecraft and differently from the previously mentioned literature, our approach to treating RPO problems is based on utilizing to a maximum extent the LQ MPC framework with constant horizon, real-valued optimization variables, and *dynamically reconfigurable* linear constraints. The MPC cost function is specified with stage and terminal costs defined on the basis of Lyapunov stability considerations. The online computations in the case of such an LQ MPC reduce to solving a quadratic program subject to linear constraints (see [19]), and in many cases, such as when approaching a nonrotating platform, this problem can be solved explicitly offline by using the parametric quadratic programming techniques developed in [20, 21]. With such an explicit MPC approach, the solution can simply be stored in the form of piecewise affine feedback law for online implementation. Preliminary results using this approach have been discussed by the authors in [22–24] and are here analyzed with more details and using problem specifications closer to the ones of real spacecraft applications.

For the general case of a rotating platform, the approach developed in this paper aims at overcoming a limitation of the standard MPC approach, namely the assumption of completely time-invariant plant and prediction model. Although not yet fully adaptive as a (constrained) Generalized Predictive Control [25, 26], where the entire plant model and optimal controller are identified online, the proposed MPC with dynamically reconfigurable constraints reduces the gap between MPC and Generalized Predictive Control, by allowing some degrees of adaption to modified external conditions (i.e., different docking port position and orientation in the considered application).

To illustrate our approach in detail, in the paper, we consider spacecraft maneuvering in *close proximity* to a disk-shaped target platform orbiting the Earth along a circular orbital track. Our

treatment is based on the assumption of point mass spacecraft, circular orbit, and in-orbital plane motion. These assumptions are reasonable for many maneuvers and can be relaxed. We first consider a scenario when the platform is nonrotating relative to its center of mass and then a scenario when the platform is rotating with known constant angular velocity relative to its center of mass. A Clohessy–Wiltshire–Hill (CWH) relative motion model is used by the MPC controller for repeated prediction and constrained optimization of spacecraft motion in response to the thrust sequence. Constraints on LOS cone positioning, terminal velocity for soft-docking, thrust magnitude, and debris avoidance are dynamically reconfigured and approximated by linear constraints, which are then enforced by the MPC controller.

The paper is organized as follows. In Section 2, we discuss spacecraft and target platform equations of motion, as well as various modeling details. The constraints in the RPO problem and their dynamic reconfiguration by linear constraints are the subject of Section 3. In Section 4, we develop MPC controllers for the cases without and with prediction of the platform motion. In Section 5, we examine the simulated trajectories for the case of a nonrotating platform, and we analyze the impact of the weights in the MPC cost function on fuel consumption-related metrics and time-to-dock. The robustness of the MPC controller is demonstrated by simulating the spacecraft motion as affected by unmeasured disturbances and comparing the closed-loop trajectories with the open-loop trajectories. The disturbances can occur because of air drag on a Low Earth orbit or errors in generating spacecraft thrust. An explicit MPC controller is also constructed in this section. For a rotating target platform, which is the case considered in Section 6, the trajectories, the fuel consumption, and the time-to-dock are compared for the implementation of the MPC controller with the prediction of the target's platform motion and without such a prediction. Finally, in Section 7, we demonstrate that our approach can be applied to the debris avoidance maneuvers. Concluding remarks are made at the end of the paper, in Section 8.

The main contributions of the paper are summarized as follows. First, we demonstrate that various constraints in the RPO problem can be handled using an LQ MPC approach coupled with dynamic reconfiguration of the (linear) constraints. This approach is feasible for implementation on-board of the spacecraft either through an online solution using a quadratic programming solver or in the case of nonrotating platform using an explicit MPC approach. Second, we demonstrate the capability of the spacecraft, controlled with MPC, to perform an approach of either a nonrotating platform or of a rotating platform and avoid debris on the spacecraft path. Fly-over imaging maneuvers involve controlling spacecraft motion on a periodic orbit around another object and can be handled using techniques developed here for approaching a rotating platform. Third, we demonstrate the robustness to unmeasured disturbances through the mechanism of systematic feedback corrections with MPC. Fourth, we demonstrate a direct connection between weights in the MPC cost function and fuel consumption and time-to-dock maneuver attributes. These results suggest that the currently employed open-loop guidance schemes coupled with ad hoc error correction procedures can be replaced in the future by a closed-loop guidance based on MPC that systematically compensates for disturbances and enforces constraints.

## 2. EQUATIONS OF MOTION

We consider autonomous rendezvous and docking maneuvers between a target platform and a spacecraft. The target platform is assumed to have a disk shape of radius  $r_p$  (m). If the platform does not have disk shape to begin with or for fly-over maneuvers, the platform can be over-bounded by a disk of a sufficiently large radius, see Figure 1. The center of mass of the platform is on a circular orbit around the Earth, and the orbital radius is  $R_0$  (m), see Figure 2. A docking port is located on the platform surface. The platform rotates at a constant angular velocity  $\omega_p \geq 0$  (rad/s) around its center of mass. The spacecraft is represented by a point mass, and it has to approach the target platform for docking to the port.

We confine the motion of the target and of the spacecraft to the orbital  $x - y$  plane, where  $y$  corresponds to the along the orbital track direction and  $x$  corresponds to the radial direction along the radius-vector from the center of the Earth to the target platform. The disturbances, for instance, because of air drag, solar pressure, and nonspherical gravity perturbation ( $J_2$ ) effects [27], are

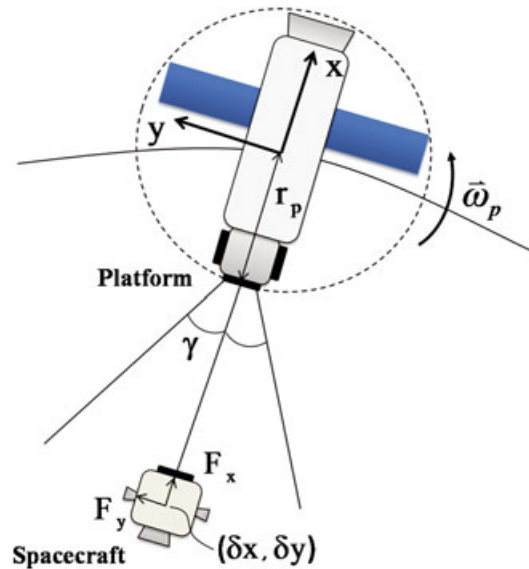


Figure 1. Schematics of spacecraft and target platform with Line-of-Sight cone.

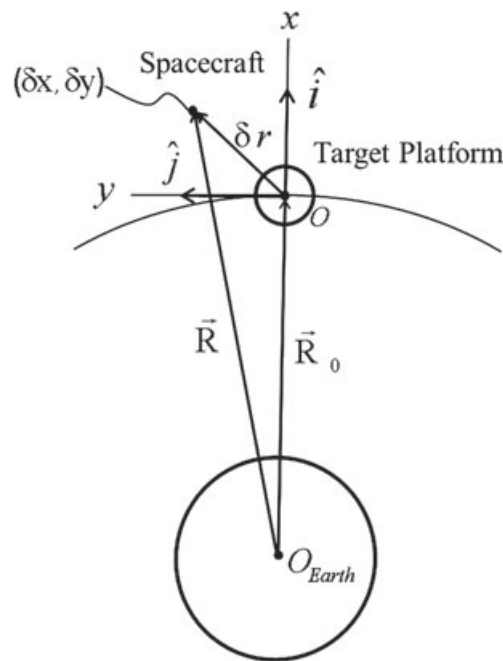


Figure 2. Relative frame centered at the target platform and relative coordinates.

neglected in the model formulation because their effects during the short time of the maneuver can be compensated by the MPC feedback, as we will show later via simulations.

The treatment of planar spacecraft motion is consistent with requirements of typical rendezvous and docking maneuvers [7]. The out-of-plane relative dynamics are decoupled from the planar dynamics and are stable, and hence are neglected, here.

The spacecraft translational motion is actuated by thrusters. We assume that thrusters can be operated to generate prescribed propulsive forces in  $x$  and  $y$  directions and that the thrust magnitude is limited. The prescribed thrust forces can be physically realized by control allocation to appropriate thruster on–off times, see [7, 17]. For a single main thruster spacecraft configuration,

we assume that the spacecraft orientation is changed appropriately by the attitude control system to realize the prescribed thrust vector. The MPC feedback can be relied upon to compensate for thrust vector direction and magnitude errors, as it will be shown later in simulations.

To express the motion of the spacecraft relative to the target platform, we use the CWH equations [10, 27]. Because the target platform is in a circular orbit around the Earth of radius  $R_0$  (m), the orbital rate is  $n = \sqrt{\frac{\mu}{R_0^3}}$  (rad/s), where  $\mu$  ( $\text{m}^3/\text{s}^2$ ) is the gravitational constant of the Earth. The reference Hill's frame is located at the target platform center of mass; hence, it rotates with orbital rate  $n$  with respect to the inertial reference frame that is located at the center of the Earth. The position vector to the target's center of mass from the center of the Earth is expressed as  $\vec{R}_0 = R_0 \hat{i}$ . The relative position vector of the spacecraft with respect to the platform is expressed as  $\delta \vec{r} = \delta x \hat{i} + \delta y \hat{j}$ , where  $\delta x, \delta y$  (m) are the components of the position vector of the spacecraft relative to the platform center. The position vector of the spacecraft with respect to the center of the Earth is thus given by  $\vec{R} = \vec{R}_0 + \delta \vec{r} = (R_0 + \delta x) \hat{i} + \delta y \hat{j}$ . The equations of motion for the spacecraft are nonlinear and can be expressed in vector form as

$$\ddot{\vec{R}} = -\mu \frac{\vec{R}}{R^3} + \frac{1}{m_c} \vec{F}, \quad (1)$$

where  $\vec{F}$  denotes the vector of forces applied to the spacecraft and  $m_c$  (kg) is the mass of the spacecraft. Given that  $R = \sqrt{(R_0 + \delta x)^2 + \delta y^2}$  [m], we obtain

$$\ddot{\vec{R}} = (\delta \ddot{x} - 2n\delta \dot{y} - n^2(R_0 + \delta x)) \hat{i} + (\delta \ddot{y} + 2n\delta \dot{x} - n^2\delta y) \hat{j}.$$

For  $\delta r \ll R$ , the CWH equations [10, 27] approximate the relative motion dynamics as

$$\begin{aligned} \delta \ddot{x} - 3n^2\delta x - 2n\delta \dot{y} &= \frac{F_x}{m_c} = u_x, \\ \delta \ddot{y} + 2n\delta \dot{x} &= \frac{F_y}{m_c} = u_y, \end{aligned} \quad (2)$$

where  $u_x, u_y$  ( $\text{m/s}^2$ ) are acceleration components of the spacecraft in  $x$  and  $y$  directions, induced by the thrust forces  $F_x, F_y$  (N), respectively. The spacecraft accelerations are subsequently treated as control signals as it is common in these applications [7].

Model (2) can be formulated as

$$\dot{X} = AX + BU, \quad (3)$$

where  $X \in \mathbb{R}^4$  is the state vector,  $U \in \mathbb{R}^2$  is the control vector, and

$$X = \begin{pmatrix} \delta x \\ \delta y \\ \delta \dot{x} \\ \delta \dot{y} \end{pmatrix}, \quad A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 2n \\ 0 & 0 & -2n & 0 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad U = \begin{pmatrix} u_x \\ u_y \end{pmatrix}.$$

For the development of the MPC controller, continuous-time spacecraft model (3) is discretized in time with sampling period  $T_s$  (s) leading to the discrete-time model

$$X(k+1) = A_d X(k) + B_d U(k), \quad (4)$$

where  $X(k) \in \mathbb{R}^4$  and  $U(k) \in \mathbb{R}^2$  denote, respectively, the state and input vectors at the sampling instant  $k \in \mathbb{Z}_{0+}$ .

The coordinates of the docking port in Hill's frame at time instant  $k$  are associated to the state variables  $r_x(k), r_y(k)$  (m) and have dynamics

$$\begin{aligned} r_x(k+1) &= \cos(\omega_p T_s) r_x(k) - \sin(\omega_p T_s) r_y(k), \\ r_y(k+1) &= \sin(\omega_p T_s) r_x(k) + \cos(\omega_p T_s) r_y(k), \end{aligned} \quad (5)$$



where  $\omega_p$  (rad/s) is the angular velocity of the platform about its center of mass. If the target platform is not rotating, then  $\omega_p = 0$  and  $r_x(k+1) = r_x(k)$ ,  $r_y(k+1) = r_y(k)$ . In the rotating platform case, (5) enables to use MPC for docking to a moving port. The relative coordinates of the spacecraft with respect to the docking port are defined as

$$\begin{aligned}\sigma_x(k+1) &= \delta x(k) - r_x(k), \\ \sigma_y(k+1) &= \delta y(k) - r_y(k).\end{aligned}\quad (6)$$

The state vector, augmented with the position of the docking port and the relative coordinates from (6), has the following form:

$$\bar{X} = \begin{pmatrix} \delta x & \delta y & \delta \dot{x} & \delta \dot{y} & r_x & r_y & \sigma_x & \sigma_y \end{pmatrix}^T.$$

From (4)–(6), we can formulate the system model as

$$\bar{X}(k+1) = \bar{A}\bar{X}(k) + \bar{B}\bar{U}(k), \quad (7)$$

with appropriately defined  $\bar{A}$ ,  $\bar{B}$ , and  $\bar{U} = [U^T \ s]^T$ , where  $s$  is an auxiliary slack variable used in the definition of the constraints, as explained next.

### 3. CONSTRAINT MODELING AND DYNAMIC RECONFIGURATION

For computational efficiency reasons, we base our approach to RPO maneuvering on the application of an LQ MPC with linear inequality constraints. Various constraints in the RPO control problem and the procedure to handle them by dynamically reconfigurable linear constraints are now discussed.

#### 3.1. Thrust constraints

For a single thruster spacecraft, the constraint on the maximum thrust magnitude has the following form:

$$u_x^2 + u_y^2 \leq u_{\max}^2. \quad (8)$$

This constraint is nonlinear, but in the LQ MPC, we can only enforce linear input constraints. By constraining

$$-\frac{u_{\max}}{\sqrt{2}} \leq u_x(k) \leq \frac{u_{\max}}{\sqrt{2}}, \quad -\frac{u_{\max}}{\sqrt{2}} \leq u_y(k) \leq \frac{u_{\max}}{\sqrt{2}}, \quad (9)$$

constraint (8) can be conservatively enforced. To avoid unnecessary control authority reduction, we impose the thrust magnitude constraints in the form

$$-u_{\max} \leq u_x(k) \leq u_{\max}, \quad -u_{\max} \leq u_y(k) \leq u_{\max}, \quad (10)$$

and if  $u_x(k)^2 + u_y(k)^2 > u_{\max}^2$  occurs for some  $k$ , we modify the computed  $u_x(k)$  and  $u_y(k)$  by directionality preserving scaling [7],

$$u'_x(k) = \frac{u_x(k)}{\sqrt{u_x(k)^2 + u_y(k)^2}} u_{\max}, \quad u'_y(k) = \frac{u_y(k)}{\sqrt{u_x(k)^2 + u_y(k)^2}} u_{\max}. \quad (11)$$

#### Remark 1

The approach mentioned earlier can be generalized to enforcing constraint

$$-\bar{\gamma}u_{\max} \leq u_x(k) \leq \bar{\gamma}u_{\max}, \quad -\bar{\gamma}u_{\max} \leq u_y(k) \leq \bar{\gamma}u_{\max}, \quad (12)$$

where  $\bar{\gamma} \in [1/\sqrt{2}, 1]$  is a parameter, chosen offline, that trades off conservativeness of the constraints and reliability of the trajectory prediction. In fact, for  $\bar{\gamma} = 1/\sqrt{2}$ , (12) reduces to (9), which is more conservative than (10), hence, limiting the performance, but it ensures that the acceleration for the

planned trajectory can always be achieved. Instead for  $\bar{\gamma} = 1$ , (12) reduces to (10), which is less conservative, yet it may occasionally happen that the acceleration for the planned trajectory cannot be actually achieved. For  $\bar{\gamma} \in (1/\sqrt{2}, 1)$ , the intermediate trade-offs are obtained.

**Remark 2**

Constraint (8) is a convex quadratic constraint. Hence, if added to the quadratic program that is generated by LQ MPC, it results in a convex Quadratically Constrained Quadratic Program (QCQP). The QCQPs are indeed more complex than linearly constrained QPs, but specific algorithms that exploit their structure to provide faster solution are becoming available [28]. Alternatively, the problem can be formulated and solved as a Second Order Cone Program [29], for which the solution usually requires more time and resources than what is typically available for this application. Because here we focus on obtaining a controller that can execute even with limited computational resources, we choose to apply approximation (12), solve the resulting QP, and perform scaling (11), if needed.

### 3.2. Line-of-Sight constraints

The LOS constraints confine the spacecraft to the intersection of the LOS cone, with vertex moved slightly inside the platform and a half-plane. See lines **a**, **b**, and **c** in Figure 3. Let  $\gamma$  denote the half of the LOS cone angle, and let  $r_{\text{tol}}[m]$  denote the distance by which the vertex of LOS cone is moved inside the platform. The value  $r_{\text{tol}} > 0$ , which is chosen offline, slightly relaxes the LOS constraints, corresponding to **a** and **b**, becoming borderline feasible as the spacecraft approaches the docking port. The constraint corresponding to the half-plane **c**, defined by a tangent line to the platform at the position of the docking port, ensures that collisions of the spacecraft with the target platform are avoided with the relaxed cone constraints.

The LOS constraints are mathematically defined by

$$\begin{cases} \mathbf{a}: \frac{\sin(\varphi(k)+\gamma)}{(r_p-r_{\text{tol}})\sin\gamma}\delta x(k) - \frac{\cos(\varphi(k)+\gamma)}{(r_p-r_{\text{tol}})\sin\gamma}\delta y(k) \geq 1, \\ \mathbf{b}: -\frac{\sin(\varphi(k)-\gamma)}{(r_p-r_{\text{tol}})\sin\gamma}\delta x(k) + \frac{\cos(\varphi(k)-\gamma)}{(r_p-r_{\text{tol}})\sin\gamma}\delta y(k) \geq 1, \\ \mathbf{c}: \frac{\cos\varphi(k)}{r_p\sin\gamma}\delta x(k) + \frac{\sin\varphi(k)}{r_p\sin\gamma}\delta y(k) \geq 1, \end{cases} \quad (13)$$

where  $\varphi(k)$  is the angle between the platform docking port and the  $x$ -axis at the time instant  $k$ .

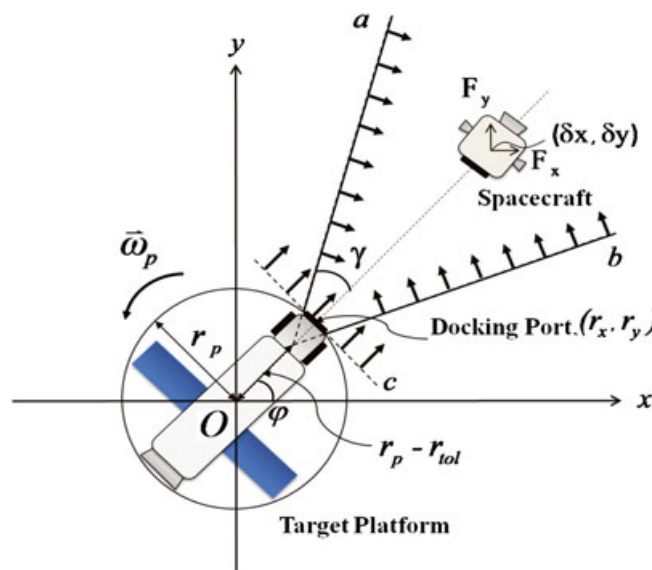


Figure 3. Geometric representation of the Line-of-Sight constraints.

For the case where the platform is not rotating and  $\varphi(k)$  is constant, that is,  $\varphi(k) = \varphi$ , the LOS constraints are linear inequalities in  $\delta x(k)$  and  $\delta y(k)$ . For the case where the platform rotates, that is,  $\varphi(k)$  changes in time, we will consider and compare two approaches for the treatment of LOS constraints (13) over the prediction horizon of the MPC problem. In the first approach,  $\varphi(k)$  is assumed to remain constant over the prediction horizon, and the constraints remain frozen. In the second approach (see Section 4.2), we will approximately predict the changes in the LOS constraint because of the platform rotation.

### 3.3. Soft-docking constraint

The soft-docking constraint ensures that the relative velocity of the spacecraft, once it approaches the docking port, is close to the docking port velocity. This ensures that the spacecraft can follow the port and avoid excessive mechanical shock when docking occurs.

Because the soft-docking constraint is a terminal constraint, to handle it by using the conventional MPC formulation, we consider a related pointwise-in-time constraint, requiring that the 1-norm of the spacecraft velocity relative to the docking port is bounded by an affine function of the 1-norm of the distance of the spacecraft relative to the docking port. A similar approach was used in our previous work [30] to handle the soft-landing constraints for an electromagnetic actuator.

In this paper we use the notation  $a(j|k)$  to indicate the value of a variable  $a$  predicted  $j$  steps ahead from step  $k$ . Let  $\sigma_x(j|k)$ ,  $\sigma_y(j|k)$  be the predicted values of the spacecraft position in  $x$  and  $y$  directions relative to the docking port  $j$  steps ahead, given that  $k$  is the current time instant at which the computations are performed. Similarly, let the predicted relative velocities be denoted by  $\delta \dot{x}(j|k)$  and  $\delta \dot{y}(j|k)$  (m/s). The docking port velocities can be predicted by  $v_{p_x}(j|k) = -\omega_p r_y(j|k)$  and  $v_{p_y}(j|k) = \omega_p r_x(j|k)$ , which we use to enforce over the MPC prediction horizon the constraint

$$|\sigma_x(j|k)| + |\sigma_y(j|k)| \geq \eta \{ |\delta \dot{x}(j|k) - v_{p_x}(j|k)| + |\delta \dot{y}(j|k) - v_{p_y}(j|k)| - s(j|k) \} - \beta. \quad (14)$$

Here,  $\eta > 0$  and  $\beta > 0$  are constant parameters that define the shape of the feasible set in the position-velocity space. The variable  $s(j|k)$  is a slack variable that was introduced as a component of  $\bar{U}$  in (7), and is used to avoid infeasibility of constraint (14).

To handle constraint (14), which is pointwise-in-time but still ‘mildly’ nonlinear, we replace it by a related linear constraint on the basis of the assumption that over the prediction horizon, the signs of  $\sigma_x(j|k)$ ,  $\sigma_y(j|k)$ ,  $\delta \dot{x}(j|k) - v_{p_x}(j|k)$  and  $\delta \dot{y}(j|k) - v_{p_y}(j|k)$  do not change. This leads to a dynamically reconfigurable constraint of the form

$$\begin{aligned} \text{sgn}(\delta x(k))(\sigma_x(j|k)) + \text{sgn}(\delta y(k))(\sigma_y(j|k)) &\geq \eta \{ \text{sgn}(\delta \dot{x}(k) - v_{p_x}(k))(\delta \dot{x}(j|k) - v_{p_x}(j|k)) \\ &\quad + \text{sgn}(\delta \dot{y}(k) - v_{p_y}(k))(\delta \dot{y}(j|k) - v_{p_y}(j|k)) - s(j|k) \} - \beta, \end{aligned} \quad (15)$$

where  $\text{sgn}(\cdot)$  indicates the well-known *sign* function. The mismatch between the predicted trajectory based on this simplifying assumption and the actual spacecraft trajectory is compensated because of MPC recomputing the solution at every time instant and updating the constraint representation in real-time.

With a limited loss of performance, constraint (15) can be further simplified to

$$\begin{aligned} \zeta(k) &\geq \eta \{ \text{sgn}(\delta \dot{x}(k) - v_{p_x}(k))(\delta \dot{x}(j|k) - v_{p_x}(j|k)) \\ &\quad + \text{sgn}(\delta \dot{y}(k) - v_{p_y}(k))(\delta \dot{y}(j|k) - v_{p_y}(j|k)) - s(j|k) \} - \beta, \end{aligned} \quad (16)$$

where

$$\begin{aligned} \zeta(k) &\triangleq |\delta x(k) - r_x(k)| + |\delta y(k) - r_y(k)| \\ &= |\sigma_x(k)| + |\sigma_y(k)|. \end{aligned} \quad (17)$$

The approach taken here to approximately handle the soft-docking constraint (and related approach for debris avoidance in Section 7) simplifies the optimization problem to a level that enables its treatment by computationally effective, conventional MPC techniques based on linear



models with linear constraints. Our subsequent simulation results indicate that this approach does not compromise the response properties and enforces satisfactorily the constraints.

#### 4. MODEL PREDICTIVE CONTROLLER DESIGN

With the dynamic model and constraints defined in Section 3, we design the MPC controller for the cases of nonrotating and rotating platforms.

To define the cost function for the MPC controller, we first compute the value function of the infinite horizon unconstrained LQ problem for stabilizing the relative position and velocity of the spacecraft to the origin, that is,

$$\min_{U(\cdot)} J = \sum_{k=0}^{\infty} X(k)^T Q X(k) + U(k)^T R U(k), \quad (18)$$

where

$$Q = \begin{pmatrix} Q_{11} & 0 & 0 & 0 \\ 0 & Q_{22} & 0 & 0 \\ 0 & 0 & Q_{33} & 0 \\ 0 & 0 & 0 & Q_{44} \end{pmatrix} = \begin{pmatrix} Q_1 & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & Q_2 \end{pmatrix}, \quad R = \begin{pmatrix} R_{11} & R_{12} \\ R_{21} & R_{22} \end{pmatrix},$$

$U(\cdot) = \{U(0), U(1), \dots\}$  and  $\mathbf{0}_{n \times m}$  denotes an  $n \times m$  zero matrix. In (18),  $Q$  is a positive-definite state weighting matrix, and  $R$  is a positive-definite control weighting matrix. Let  $P$  denote the solution of the Riccati equation for solving problem (18) and  $K$  the corresponding LQR feedback gain,

$$P = \begin{pmatrix} P_{11} & P_{12} & P_{13} & P_{14} \\ P_{21} & P_{22} & P_{23} & P_{24} \\ P_{31} & P_{32} & P_{33} & P_{34} \\ P_{41} & P_{42} & P_{43} & P_{44} \end{pmatrix} = \begin{pmatrix} P_1 & P_2 \\ P_3 & P_4 \end{pmatrix}, \quad K = \begin{pmatrix} K_{11} & K_{12} & K_{13} & K_{14} \\ K_{21} & K_{22} & K_{23} & K_{24} \end{pmatrix},$$

so that the value function is  $v(X(0)) = X(0)^T P X(0)$  for the LQ problem. We use  $P$  in the terminal cost of the MPC problem, consistently with the classical MPC theory (see [16, 31, 32]) as a stability enforcing mechanism.

##### 4.1. Model Predictive Controller without prediction of platform motion

For the case of an MPC controller that does not use the prediction of the platform motion over the horizon, the prediction model is based on (7) and (13) with  $\omega_p = 0$  and  $\varphi(k) = \varphi$ . We can express the prediction model in the form

$$\bar{X}(j+1|k) = \bar{A}\bar{X}(j|k) + \bar{B}\bar{U}(j|k), \quad (19a)$$

$$\bar{Y}(j|k) = \bar{C}\bar{X}(j|k) + \bar{D}\bar{U}(j|k), \quad (19b)$$

where (19b) represents the constrained output because of (7) and (14). These output constraints are imposed as

$$\bar{Y}(j|k) \geq \bar{Y}_{\min}(k), \quad (20)$$

where  $\bar{Y}_{\min}(k) = (1 \ 1 \ 1 \ -\beta - \zeta(k))^T$ . The matrices  $\bar{A}$ ,  $\bar{B}$ ,  $\bar{C}$ , and  $\bar{D}$  in (19) are expressed as

$$\bar{A} = \begin{pmatrix} A_d & \mathbf{0}_{4 \times 4} \\ \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} & \Omega & \mathbf{0}_{2 \times 2} \\ I & \mathbf{0}_{2 \times 2} & -I & \mathbf{0}_{2 \times 2} \end{pmatrix}, \quad \bar{B} = \begin{pmatrix} B_d & \mathbf{0}_{4 \times 1} \\ \mathbf{0}_{4 \times 2} & \mathbf{0}_{4 \times 1} \end{pmatrix},$$

$$\bar{C} = \begin{pmatrix} \bar{C}_{11} & \bar{C}_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\ \bar{C}_{21} & \bar{C}_{22} & 0 & 0 & 0 & 0 & 0 & 0 \\ \bar{C}_{31} & \bar{C}_{32} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \bar{C}_{43} & \bar{C}_{44} & \bar{C}_{45} & \bar{C}_{46} & 0 & 0 \end{pmatrix}, \quad (21)$$

$$\bar{D} = \begin{pmatrix} \mathbf{0}_{3 \times 2} & \mathbf{0}_{3 \times 1} \\ \mathbf{0}_{1 \times 2} & \eta \end{pmatrix},$$

where

$$\Omega = \begin{pmatrix} \cos(\omega_p T_s) & -\sin(\omega_p T_s) \\ \sin(\omega_p T_s) & \cos(\omega_p T_s) \end{pmatrix}, \quad I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

and by assuming  $\varphi(j|k) = \varphi(k) = \varphi$ ,

$$\begin{aligned} \bar{C}_{11} &= \frac{\sin(\varphi + \gamma)}{(r_p - r_{\text{tol}}) \sin \gamma}, \quad \bar{C}_{12} = -\frac{\cos(\varphi + \gamma)}{(r_p - r_{\text{tol}}) \sin \gamma}, \quad \bar{C}_{21} = -\frac{\sin(\varphi - \gamma)}{(r_p - r_{\text{tol}}) \sin \gamma}, \\ \bar{C}_{22} &= \frac{\cos(\varphi - \gamma)}{(r_p - r_{\text{tol}}) \sin \gamma}, \quad \bar{C}_{31} = \frac{\cos \varphi}{r_p \sin \gamma}, \quad \bar{C}_{32} = \frac{\sin \varphi}{r_p \sin \gamma}. \end{aligned}$$

Here,  $\bar{C}_{11}, \bar{C}_{12}, \bar{C}_{21}, \bar{C}_{22}, \bar{C}_{31}$ , and  $\bar{C}_{32}$  are used to represent LOS constraints (13). The elements  $\bar{C}_{43}, \bar{C}_{44}, \bar{C}_{45}$ , and  $\bar{C}_{46}$  of  $\bar{C}$  are used to model (16), which approximates soft-docking constraint (14). Note that if  $\delta \dot{x}(k) - v_{p_x}(k) \geq 0$ , then  $\bar{C}_{43} = -\eta, \bar{C}_{46} = \eta \omega_p$ ; otherwise,  $\bar{C}_{43} = \eta, \bar{C}_{46} = -\eta \omega_p$ . If  $\delta \dot{y}(k) - v_{p_y}(k) \geq 0$ , then  $\bar{C}_{44} = -\eta, \bar{C}_{45} = -\eta \omega_p$ ; otherwise,  $\bar{C}_{44} = \eta, \bar{C}_{45} = \eta \omega_p$ .

At every time instant  $k$ , the MPC controller determines the control action on the basis of the solution of the following optimization problem

$$\min_{\bar{U}(k)} \bar{X}(N_J|k)^T \bar{P} \bar{X}(N_J|k) + \sum_{j=0}^{N_J-1} \bar{X}(j|k)^T \bar{Q} \bar{X}(j|k) + \bar{U}(j|k)^T \bar{R} \bar{U}(j|k), \quad (22a)$$

$$\text{s.t. } \bar{X}(j+1|k) = \bar{A} \bar{X}(j|k) + \bar{B} \bar{U}(j|k), \quad (22b)$$

$$\bar{Y}(j|k) = \bar{C} \bar{X}(j|k) + \bar{D} \bar{U}(j|k), \quad (22c)$$

$$\bar{X}(0|k) = \bar{X}(k), \quad (22d)$$

$$\bar{U}(j|k) = \bar{K} \bar{X}(j|k), \quad j = N_U + 1, \dots, N_J - 1, \quad (22e)$$

$$\bar{Y}(j|k) \geq \bar{Y}_{\min}(k), \quad j = 0, \dots, N_C, \quad (22f)$$

$$\bar{U}(j|k) \geq \bar{U}_{\min}, \quad j = 0, \dots, N_U, \quad (22g)$$

$$\bar{U}(j|k) \leq \bar{U}_{\max}, \quad j = 0, \dots, N_U, \quad (22h)$$

where  $\bar{U}(k) = \{\bar{U}(0|k), \dots, \bar{U}(N_U|k)\}$ ,  $N_J$  denotes the prediction horizon,  $N_U$  denotes the control horizon, and  $N_C$  denotes the constraint horizon. Smaller values of  $N_U$  and  $N_C$  tend to reduce the complexity of optimal control problem (22) and hence computational requirement of the platform where the MPC controller is executed. The input constraints in (22) are defined by (10). The matrices  $\bar{P}, \bar{Q}, \bar{R}$ , and  $\bar{K}$  are constructed from  $Q$  and  $R$  in (18) and the solution,  $P$ , of the Riccati equation and the LQR gain  $\bar{K}$  for (18) as

$$\bar{P} = \begin{pmatrix} \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & P_4 & \mathbf{0}_{2 \times 2} & P_3 \\ \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & P_2 & \mathbf{0}_{2 \times 2} & P_1 \end{pmatrix}, \quad \bar{Q} = \begin{pmatrix} \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & Q_2 & \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} & Q_1 \end{pmatrix},$$

$$\bar{R} = \begin{pmatrix} R & \mathbf{0}_{2 \times 1} \\ \mathbf{0}_{1 \times 2} & \rho \end{pmatrix}, \quad \bar{K} = \begin{pmatrix} \mathbf{0}_{2 \times 2} & K_{13} & K_{14} & \mathbf{0}_{2 \times 2} & K_{11} & K_{12} \\ \mathbf{0}_{2 \times 2} & K_{23} & K_{24} & \mathbf{0}_{2 \times 2} & K_{21} & K_{22} \end{pmatrix}.$$

In (22),  $\rho > 0$  is a large weight on the slack variable causing this to be zero whenever feasible.

At every control cycle, from the measured/estimated state  $\bar{X}(k)$ , the MPC controller solves (22) with respect to the finite sequence of control actions,  $\bar{U}(k)$ , and applies the first element of the optimal sequence  $\bar{U}^*(k)$  to the plant,  $\bar{U}(k) = \bar{U}^*(0|k)$ . The MPC feedback law is defined implicitly as the solution of constrained optimization problem (22); however, because of the structure of (22), this feedback law is a static (nonlinear) function of the current state,  $\bar{U}(k) = \bar{U}_{\text{MPC}}(\bar{X}(k))$ .

#### 4.2. Model Predictive Controller with prediction of platform motion

Even if the constraints in (22) change with time, as the initial state of the finite horizon optimal control problem  $\bar{X}$  changes, the bounds are assumed constant in prediction. In this section, we propose a method to incorporate a prediction of LOS constraints (13) changes because of the rotation of the docking port for the MPC optimization problem. To continue exploiting an LQ MPC framework, we employ the approximations on the basis of the Taylor series expansion

$$\varphi(j|k) \simeq \varphi(k) + \dot{\varphi}(k)jT_s = \varphi(k) + \bar{\varphi}(j, k, T_s), \quad (23)$$

$$\begin{aligned} \sin(\varphi(j|k) + \gamma) &\simeq \sin(\varphi(k) + \gamma) + \cos(\varphi(k) + \gamma)\bar{\varphi}(j, k, T_s), \\ \cos(\varphi(j|k) + \gamma) &\simeq \cos(\varphi(k) + \gamma) - \sin(\varphi(k) + \gamma)\bar{\varphi}(j, k, T_s), \end{aligned} \quad (24)$$

where  $k$  denotes the current time instant and  $j \in \mathbb{Z}_{0+}$  is a future time instant with respect to  $k$ . By substituting (23) and (24) into the LOS constraints (13), we can obtain the following LOS constraints for prediction of the platform motion:

$$\begin{cases} \mathbf{a}' : & L_1\delta x(j|k) - L_2\delta y(j|k) + \{L_2\delta x(k) + L_1\delta y(k)\}(\varphi(j|k) - \varphi(k)) \geq 1, \\ \mathbf{b}' : & -L_3\delta x(j|k) + L_4\delta y(j|k) - \{L_4\delta x(k) + L_3\delta y(k)\}(\varphi(j|k) - \varphi(k)) \geq 1, \\ \mathbf{c}' : & L_5\delta x(j|k) + L_6\delta y(j|k) - \{L_6\delta x(k) + L_5\delta y(k)\}(\varphi(j|k) - \varphi(k)) \geq 1, \end{cases} \quad (25)$$

where

$$\begin{aligned} L_1 &= \frac{\sin(\varphi(k) + \gamma)}{(r_p - r_{\text{tol}}) \sin \gamma}, \quad L_2 = \frac{\cos(\varphi(k) + \gamma)}{(r_p - r_{\text{tol}}) \sin \gamma}, \quad L_3 = \frac{\sin(\varphi(k) - \gamma)}{(r_p - r_{\text{tol}}) \sin \gamma}, \\ L_4 &= \frac{\cos(\varphi(k) - \gamma)}{(r_p - r_{\text{tol}}) \sin \gamma}, \quad L_5 = \frac{\cos(\varphi(k))}{r_p \sin \gamma}, \quad L_6 = \frac{\sin(\varphi(k))}{r_p \sin \gamma}. \end{aligned}$$

In addition, to predict the future position of the LOS cone, we introduce the auxiliary state vector,

$$Z(j|k) = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} = \begin{pmatrix} \varphi(j+1|k) \\ \varphi(j|k) \end{pmatrix} = \begin{pmatrix} \varphi(j|k) + \omega_p T_s \\ \varphi(j|k) \end{pmatrix}, \quad (26)$$

with dynamics defined by

$$Z(j+1|k) = \begin{pmatrix} 2 & -1 \\ 1 & 0 \end{pmatrix} Z(j|k) = \Theta Z(j|k). \quad (27)$$

Considering (7), (25), and (27), the augmented prediction model has state vector

$$\tilde{X} = (\delta x \quad \delta y \quad \delta \dot{x} \quad \delta \dot{y} \quad r_x \quad r_y \quad \sigma_x \quad \sigma_y \quad z_1 \quad z_2)^T \quad (28)$$

and dynamics formulated as a linear system subject to time-varying constraints

$$\tilde{X}(j+1|k) = \tilde{A}\tilde{X}(j|k) + \tilde{B}\tilde{U}(j|k), \quad (29a)$$

$$\tilde{Y}(j|k) = \tilde{C}\tilde{X}(j|k) + \tilde{D}\tilde{U}(j|k), \quad (29b)$$

where

$$\tilde{A} = \begin{pmatrix} \bar{A} & \mathbf{0}_{8 \times 2} \\ \mathbf{0}_{2 \times 8} & \Theta \end{pmatrix}, \quad \tilde{B} = \begin{pmatrix} \bar{B} \\ \mathbf{0}_{2 \times 3} \end{pmatrix}, \quad \tilde{D} = \bar{D}, \quad \tilde{U} = \bar{U},$$

$$\tilde{C} = \begin{pmatrix} \tilde{C}_{11} & \tilde{C}_{12} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \tilde{C}_{\text{LOS1}} \\ \tilde{C}_{21} & \tilde{C}_{22} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \tilde{C}_{\text{LOS2}} \\ \tilde{C}_{31} & \tilde{C}_{32} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \tilde{C}_{\text{LOS3}} \\ 0 & 0 & \tilde{C}_{43} & \tilde{C}_{44} & \tilde{C}_{45} & \tilde{C}_{46} & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$\tilde{C}_{\text{LOS1}} = L_2 \delta x(k) + L_1 \delta y(k),$$

$$\tilde{C}_{\text{LOS2}} = -L_4 \delta x(k) - L_3 \delta y(k),$$

$$\tilde{C}_{\text{LOS3}} = -L_6 \delta x(k) - L_5 \delta y(k),$$

so that we have the auxiliary state vector and terms for the LOS constraints in the new model.

The output constraints (cf. (20)) are

$$\tilde{Y}(j|k) \geq \tilde{Y}_{\min}(k), \quad \tilde{Y}_{\min}(k) = \tilde{Y}_{\min}(k). \quad (30)$$

Thus, for the case where prediction of the port motion is performed, the MPC optimal control problem is formulated from (28)–(30) as

$$\min_{\tilde{U}(k)} \tilde{X}(N_J|k)^T \tilde{P} \tilde{X}(N_J|k) + \sum_{j=0}^{N_J-1} \tilde{X}(j|k)^T \tilde{Q} \tilde{X}(j|k) + \tilde{U}(j|k)^T \tilde{R} \tilde{U}(j|k), \quad (31a)$$

$$\text{s.t. } \tilde{X}(j+1|k) = \tilde{A} \tilde{X}(j|k) + \tilde{B} \tilde{U}(j|k), \quad (31b)$$

$$\tilde{Y}(j|k) = \tilde{C} \tilde{X}(j|k) + \tilde{D} \tilde{U}(j|k), \quad (31c)$$

$$\tilde{X}(0|k) = \tilde{X}(k), \quad (31d)$$

$$\tilde{U}(j|k) = \tilde{K} \tilde{X}(j|k), \quad j = N_U + 1, \dots, N_J - 1, \quad (31e)$$

$$\tilde{Y}(j|k) \geq \tilde{Y}_{\min}(k), \quad j = 0, \dots, N_C, \quad (31f)$$

$$\tilde{U}(j|k) \geq \tilde{U}_{\min}, \quad j = 0, \dots, N_U, \quad (31g)$$

$$\tilde{U}(j|k) \leq \tilde{U}_{\max}, \quad j = 0, \dots, N_U, \quad (31h)$$

where  $\tilde{U}(k) = \{\tilde{U}(0|k), \dots, \tilde{U}(N_U|k)\}$ ,

$$\tilde{P} = \begin{pmatrix} \tilde{P} & \mathbf{0}_{8 \times 2} \\ \mathbf{0}_{2 \times 8} & \mathbf{0}_{2 \times 2} \end{pmatrix}, \quad \tilde{Q} = \begin{pmatrix} \tilde{Q} & \mathbf{0}_{8 \times 2} \\ \mathbf{0}_{2 \times 8} & \mathbf{0}_{2 \times 2} \end{pmatrix}, \quad \tilde{R} = \tilde{R}, \quad \tilde{K} = \begin{pmatrix} \tilde{K} & \mathbf{0}_{2 \times 2} \end{pmatrix},$$

and the input constraints in (31) are defined by (10).

## 5. SIMULATED APPROACH OF A NONROTATING PLATFORM

In the simulations, we have used the following parameters representative of spacecraft maneuvering in close proximity of a nonrotating target platform: the radius of the target  $r_p$  is 2.5 m, the half angle of the LOS cone  $\gamma$  is 10 deg, the tolerance  $r_{\text{tol}}$  is 0.5 m, and the orbital rate  $n$  is  $1.107 \times 10^{-3}$  rad/s (corresponding to the orbit of 500 km above the Earth). The controller sampling period is  $T_s = 0.5$  s. The total maneuver simulation time is 100 s. In (14),  $\eta = 1$ , and  $\beta = \frac{2.5}{10}$ . The slack variable weight is set as  $\rho = 10^{10}$ . The weighting matrices are chosen in the form

$$Q = 3 \times 10^3 \begin{pmatrix} 10^2 I & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 2} & I \end{pmatrix}, \quad R = 10^2 I.$$

The value of  $R$  was subsequently modified to study the sensitivity of fuel consumption and time-to-dock. In the simulations, we use  $u_{\max} = 0.2$  m/s<sup>2</sup> for the input constraints.

For all simulations, the prediction horizon of the MPC problem was set as  $N_J = 40$ , the constraint horizon for both input and output constraints was set as  $N_C = 5$ , and the control horizon was set as  $N_U = 5$ . These values were determined by tuning closed-loop response by using simulations.

### 5.1. Radial approach

In the radial approach, the spacecraft approaches the platform along the radial line from the center of the Earth to the center of the target. To simulate the radial approach, we choose the initial location for the spacecraft as  $(\delta x_0, \delta y_0) = (100, -10)$  (m), which is in the range of admissible initial conditions for RPO maneuvers and the initial position of the docking port as  $(r_{x0}, r_{y0}) = (2.5, 0)$  (m). The closed-loop responses are shown in Figure 4. In this and other plots of spacecraft trajectory on the  $x-y$  plane, the LOS constraints are shown by red dashed lines. The soft-docking, the LOS cone, and thrust magnitude constraints are enforced by the MPC controller, and the spacecraft successfully completes the maneuver.

### 5.2. In-track approach

In the in-track approach, the spacecraft approaches the platform in the direction along the orbital track. To simulate the in-track approach, the initial location of the spacecraft is chosen as  $(\delta x_0, \delta y_0) = (-10, 100)$  (m) and the initial position of the docking port as  $(r_{x0}, r_{y0}) = (0, 2.5)$  (m). The closed-loop responses are shown in Figure 5.

### 5.3. Trajectories from different initial locations

The initial location of the spacecraft is now varied within the LOS cone. The starting points consist of points on the boundaries of the LOS cone and points in the interior of LOS cone. The results are shown in Figure 6 for the radial approach and for the in-track approach. The trajectories near both boundaries have similar curvature in both cases.

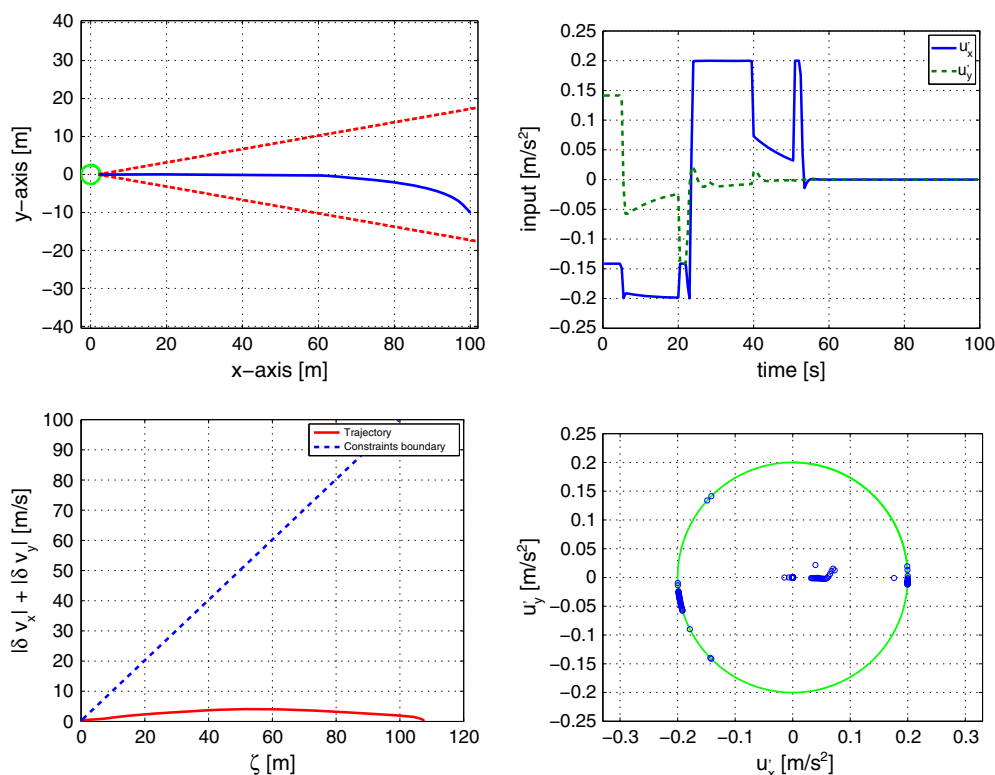


Figure 4. Radial approach to a nonrotating platform with the Model Predictive Controller. Top row: trajectory on the  $x-y$  plane (left) and control accelerations  $u'_x$  and  $u'_y$  (right). Bottom row: relative velocity 1-norm versus relative position 1-norm (left) and the plot of  $u'_y$  versus  $u'_x$  with magnitude saturation (right).



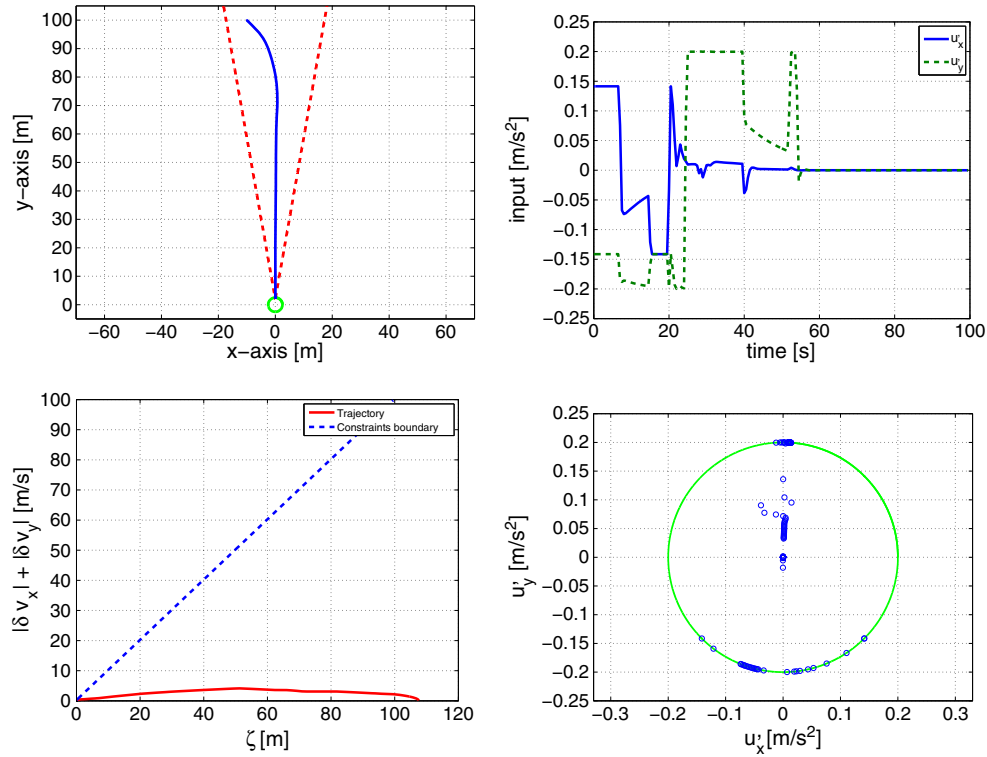


Figure 5. In-track approach to a nonrotating platform with the Model Predictive Controller. Top row: trajectory on the  $x$ - $y$  plane (left) and control accelerations  $u'_x$  and  $u'_y$  (right). Bottom row: relative velocity 1-norm versus relative position 1-norm (left) and the plot of  $u'_y$  versus  $u'_x$  with magnitude saturation (right).

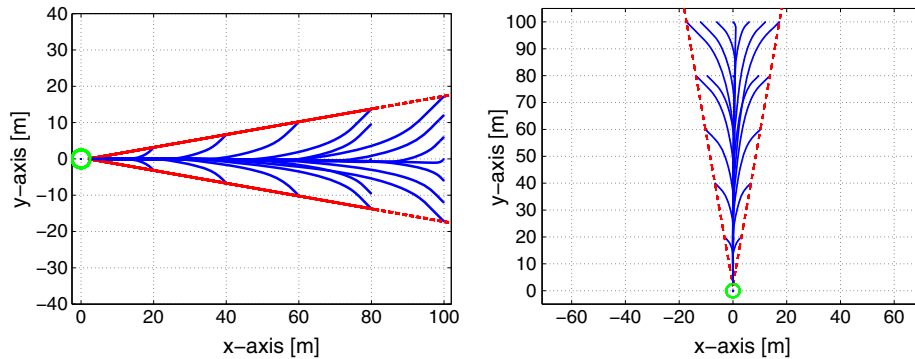


Figure 6. Trajectories from different initial spacecraft locations for a nonrotating platform for radial approach (left) and for in-track approach (right).

#### 5.4. Analysis of fuel consumption-related metrics

We consider three fuel consumption-related metrics

$$J_1 = \sum_{k=0}^{T_d} |u_x(k)| + |u_y(k)|,$$

$$J_2 = \sum_{k=0}^{T_d} (u_x(k))^2 + (u_y(k))^2,$$

$$J_3 = \sum_{k=0}^{T_d} \sqrt{(u_x(k))^2 + (u_y(k))^2}, \quad (32)$$

where  $T_d = \text{ceil}(t_d/T_s)$ ,  $\text{ceil}$  is the rounding to the closest larger integer, and  $t_d$  is the ‘time-to-dock’ in seconds, that is, the time it takes from the initial condition to achieve docking. In the simulation, docking is considered to occur when the spacecraft reaches a distance of 0.1 m from the docking port.

The metric  $J_1$  is relevant to the fuel consumption of the spacecraft that has sets of orthogonal thrusters that can be simultaneously fired. The metric  $J_2$  represents a quadratic penalty on the control effort and is closely related to the control cost in the MPC cost function but not directly to the fuel consumption. Finally, the metric  $J_3$  is relevant to the spacecraft with a single thruster that is reoriented (e.g., by controlling the spacecraft attitude) as necessary to realize the desired propulsive force. In what follows, we highlight the relationship between these metrics for the MPC maneuvers on the basis of the results of our simulations.

Tables I–III summarize the results in the radial approach for different values of the matrix  $R$  scaled by a parameter  $\alpha$ , which is varied between  $10^2$  and  $10^9$ , that is,

$$R = \alpha I. \quad (33)$$

Note that when  $\alpha$  ranges between  $10^2$  and  $10^7$ ,  $J_1$ ,  $J_2$ ,  $J_3$ , and  $t_d$  are essentially constant. The  $J_i$ ,  $i = 1, 2, 3$ , all decrease, and  $t_d$  increases for  $10^7 \leq \alpha \leq 2.5 \times 10^8$ . For  $\alpha \geq 2.5 \times 10^8$ , the trajectories start to behave differently. They are faster in the initial phase, approach the platform a bit higher on the  $y$ -axis than the docking port, and then proceed (‘slide’) to the docking port. The  $J_i$ ,  $i = 1, 2, 3$ , continue to decrease as  $\alpha$  increases; however,  $t_d$  also decreases (from 74.5 s at  $\alpha = 2 \times 10^8$  to 65.5 s at  $\alpha = 10^9$ ).

Overall, the three different metrics,  $J_1$ ,  $J_2$ , and  $J_3$ , appear to exhibit similar trends that are opposite to that of  $t_d$ , except for very large values of  $\alpha$ , when, however, the controller tends to be scarcely

Table I. Metrics in radial approach when  $\alpha$  varies between  $10^2$  and  $10^7$ .

$\alpha$	$10^2$	$10^3$	$10^4$	$10^5$	$10^6$	$10^7$
$J_1$	20.43	20.44	20.43	20.34	20.63	20.63
$J_2$	3.37	3.37	3.37	3.35	3.36	3.31
$J_3$	17.72	17.72	17.74	17.62	17.70	17.76
$t_d$	53.00	53.00	53.00	52.50	52.50	54.00

Table II. Metrics in radial approach when  $\alpha$  varies between  $10^7$  and  $2.5 \times 10^8$ .

$\alpha$	$10^7$	$2.5 \times 10^7$	$5 \times 10^7$	$7.5 \times 10^7$	$10^8$	$2.5 \times 10^8$
$J_1$	20.63	20.59	20.17	20.13	19.96	19.53
$J_2$	3.31	3.34	3.27	3.31	3.28	3.22
$J_3$	17.76	17.91	17.67	17.76	17.66	17.39
$t_d$	54.00	57.50	63.50	67.00	69.50	74.50

Table III. Metrics in radial approach when  $\alpha$  varies between  $2.5 \times 10^8$  and  $10^9$ .

$\alpha$	$2.5 \times 10^8$	$5 \times 10^8$	$7.5 \times 10^8$	$10^9$
$J_1$	19.53	19.19	18.94	18.77
$J_2$	3.22	3.14	3.08	3.03
$J_3$	17.39	17.12	16.91	16.77
$t_d$	74.50	74.50	71.50	65.50

robust to external disturbances. These results indicate that the quadratic type MPC cost function that does not directly account for the fuel consumption yet allows to formulate the controller as an LQ MPC has a direct influence on the final attributes (fuel consumption and time-to-dock). The cost function weights can be used as tuning knobs to adjust the fuel consumption versus the time-to-dock performance of the spacecraft. Figure 7 illustrates the influence of  $\alpha$  on the spacecraft trajectory during radial approach as  $\alpha$  varies as in Tables I–III.

### 5.5. Compensation of the disturbances

As opposed to robust MPC (see e.g., [33–35] and the references therein), the MPC design that is proposed here does not provide an analytically quantifiable robustness. However, it possesses the intrinsic robustness of feedback control, while resulting in a simpler algorithm that does not need information on the ranges and type of disturbances, which, for the application at hand, may be difficult to obtain.

In this section, we demonstrate the intrinsic robustness of the MPC controller to unmeasured disturbances. In orbit, the disturbances can occur because of thrust errors, air drag in Low Earth Orbit, or solar pressure in Geostationary Orbit. The simulations are performed here for a nonrotating target platform.

The air drag can be represented by a constant *in-track* disturbance acceleration [7] with magnitude

$$w = \frac{F_y}{m_c} = -\frac{\rho}{2} n^2 R_0^2 \left( \frac{1}{B_c} - \frac{1}{B_p} \right),$$

where  $B_c$  is the ballistic coefficient of the spacecraft,  $B_p$  is the ballistic coefficient of the platform, and  $\rho$  is average air density at spacecraft altitude. Equations (3) with the disturbance acceleration added take the following form:

$$\bar{X}(k+1) = \bar{A}\bar{X}(k) + \bar{B}\bar{U}(k) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ T_s \end{bmatrix} w. \quad (34)$$

Motivated by this air drag model, we assume an aggressive scenario when the disturbance is acting along the orbital track and had large constant magnitude of 10% of thrust, that is,  $w = 0.1 \times u_{\max}$ .

We first simulate the open-loop spacecraft motion with the control inputs specified as functions of time and under the effects of the disturbance. These control inputs are the same as in Figures 4 and 5 and successfully complete the maneuver without a disturbance. The closed-loop trajectories are also simulated, where the control input is recomputed by the MPC at every step by using the current state as an initial condition and without knowledge of the disturbance. The results of the

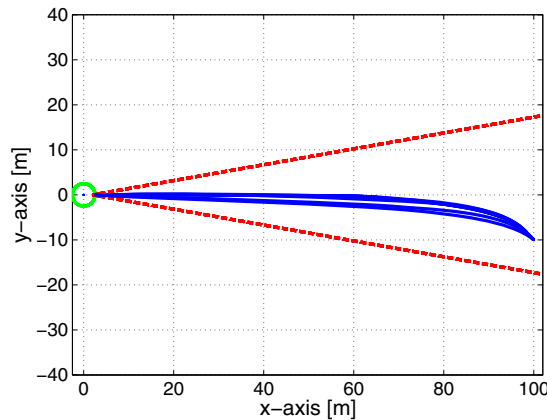


Figure 7. Trajectories when  $\alpha$  varies between  $10^2$  and  $10^9$ .

open-loop and closed-loop maneuvers for radial and in-track approach by the spacecraft affected by the disturbance are shown in Figures 8 and 9. With the open-loop control, the spacecraft fails to complete the maneuver because of the disturbances. On the other hand, the MPC controller is able to successfully guide the spacecraft despite these disturbances: The final error is about 1.2 cm for both radial and in-track maneuvers.

Figures 10 and 11 illustrate additional responses to disturbances for the radial and in-track approaches, respectively. We consider the cases of (i) constant disturbance vector with components of magnitude  $(0.1 \times u_{\max})$  on both  $x$  and  $y$  axis and (ii) random disturbance in acceleration actuation amplitude and direction. In the second case, the disturbance simulates errors in thrust direction, for instance because of some error in the attitude control, and amplitude, for instance because of realization of the continuous thrust through thrust pulses [7]. In this case, the actuated spacecraft acceleration is

$$u_r(k) = \text{sat}_2^{u_{\max}}(R_{\theta(k)}U^*(0|k)),$$

$$u(k) = \text{sat}_2^{u_{\max}}((1 + u_d(k))u_r(k)),$$

where  $\text{sat}_2^{u_{\max}}$  denotes the saturation in 2-norm by directionality preserving scaling (11),  $R_{\theta} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}$  is the matrix producing a rotation of angle  $\theta$ , and  $\theta(k) \in [-\frac{\pi}{6}, \frac{\pi}{6}]$ ,  $u_d(k) \in [-0.15, 0.15]$  are independent, uniformly distributed discrete-time random variables whose values change every 5 s. In radial and in-track approaches, the MPC controller is able to successfully compensate the effect of these disturbances. The trajectories obtained for 10 simulations with random disturbances on both radial and in-track approaches are shown in Figure 12. Finally, in our simulations, actuation disturbances of up to  $\pm 25\%$  magnitude and  $\pm 45$  deg direction appear to be tolerable for the proposed control strategy.

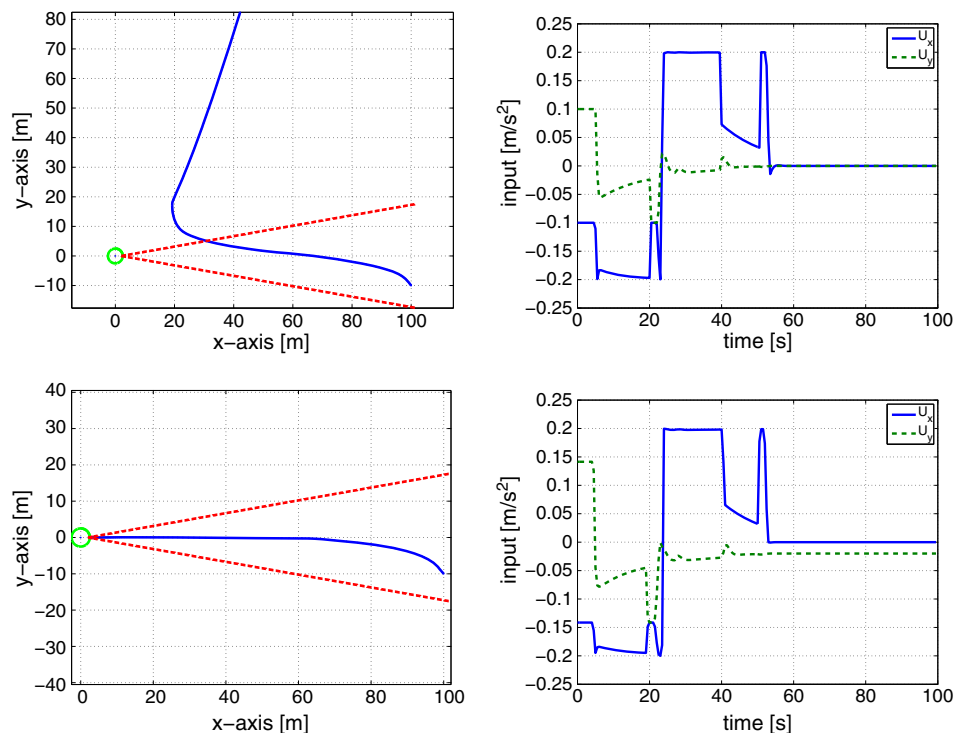


Figure 8. Radial approach subject to disturbances. Top row: open-loop trajectory (left) and open-loop control accelerations (right). Bottom row: closed-loop trajectory (left) and closed-loop control accelerations (right).

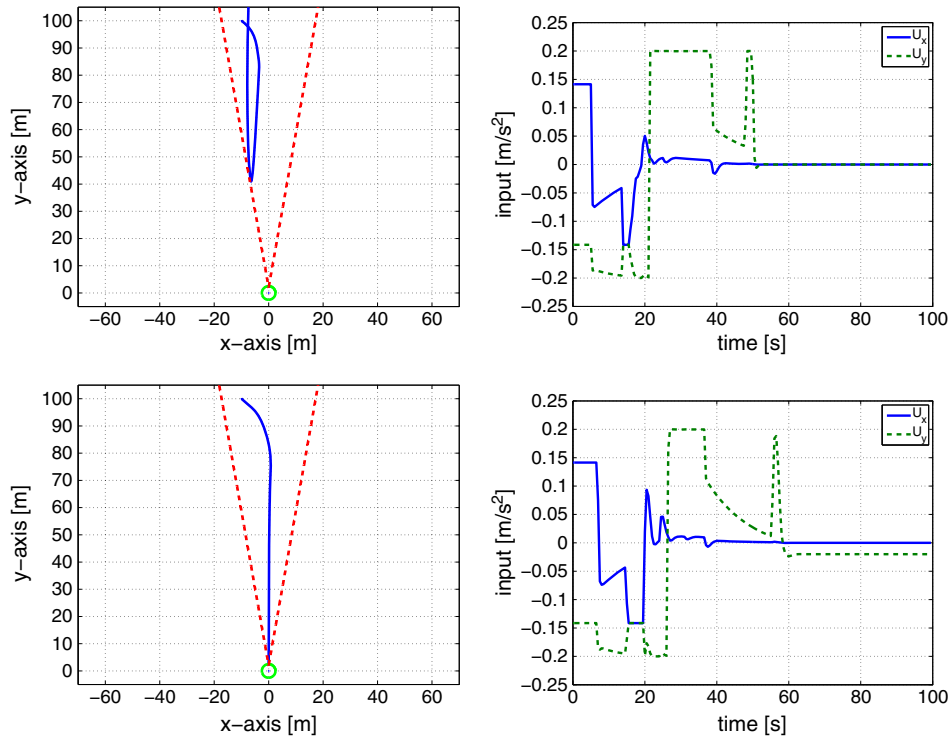


Figure 9. In-track approach subject to disturbances. Top row: open-loop trajectory (left) and open-loop control accelerations (right). Bottom row: closed-loop trajectory (left) and closed-loop control accelerations (right).

### 5.6. Explicit Model Predictive Control

When MPC is applied to linear systems with linear constraints and quadratic cost function, the control law can be explicitly computed by multiparametric quadratic programming (see [20, 21]). The MPC control law,  $\bar{U}_{\text{MPC}}(\bar{X})$ , is a piecewise affine state feedback, specified through the state-space partitioning into polyhedral regions and an affine state feedback law assigned to each region. With the use of the explicit MPC control law, the need to perform online optimization and to validate and embed the QP solver in the control software is avoided.

For the case where the target platform does not rotate and the angle of the LOS cone is known and fixed, we can design an explicit MPC controller. Because the coefficients of  $\bar{C}$  in Equation (20) change discretely depending on the current state, a slightly modified synthesis procedure is required, similarly to [36].

First, we enumerate the possible values of  $\bar{C}$  in (19), caused by the changing signs of the current state in (15) (or (16)), hence, obtaining the set of matrices,  $\mathcal{C} = \{\bar{C}_h\}_{h=1}^q$ . Then, we compute the control laws  $\gamma_{\text{MPC}}(i, \bar{X})$ ,  $i = 1, \dots, q$ , by applying multiparametric programming to the MPC problem, where  $\bar{C} = \bar{C}_i$  is a constant. Because each of the corresponding MPC optimization problems is a standard quadratic program, each feedback law has a piecewise affine form

$$\begin{aligned} \gamma_{\text{MPC}}(i, \bar{X}) &= F_j^i \bar{X} + G_j^i \\ j : H_j^i \bar{X} &\leq K_j^i, \end{aligned} \quad (35)$$

where  $j \in \mathcal{J}_i$ ,  $\mathcal{J}_i = \{1, \dots, q_i\}$  and  $q_i$  is the number of regions of the MPC law associated to the case  $\bar{C} = \bar{C}_i$ . Note also that the switching conditions on the  $\bar{C}$  coefficients are independent of the current control input and can be encoded by linear inequalities, such that  $\bar{C} = \bar{C}_i$  if and only if



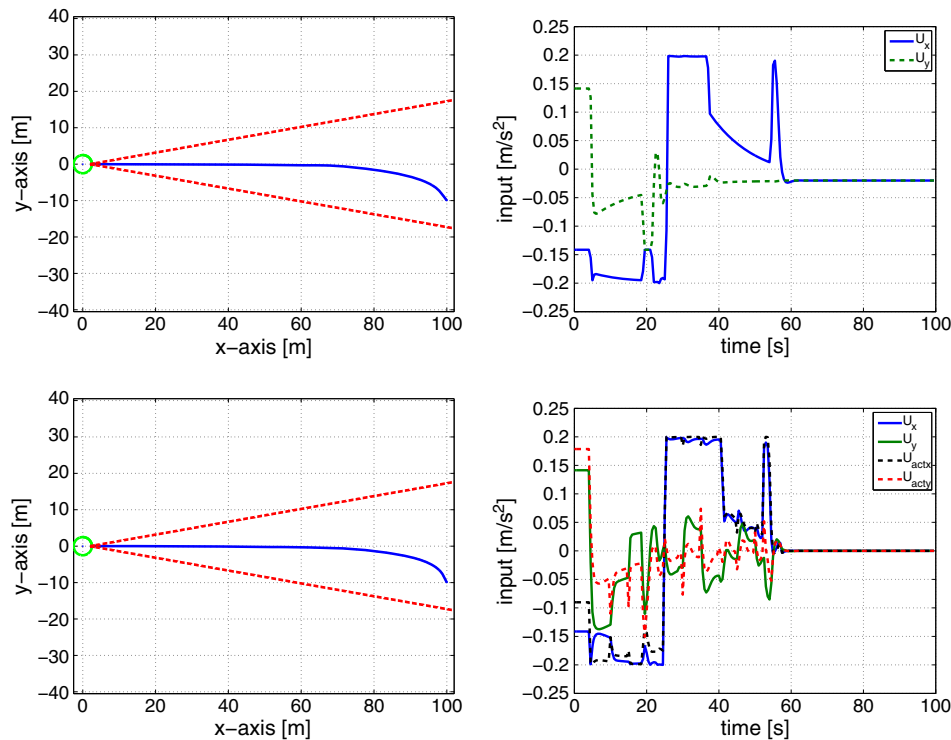


Figure 10. Radial approach subject to disturbances. Top row: constant disturbance in  $x$ - $y$  direction. Closed-loop trajectory (left) and closed-loop control accelerations (right). Bottom row: random magnitude disturbance in  $x$ - $y$  direction. Closed-loop trajectory (left) and closed-loop control accelerations (solid) and actuated accelerations (dash) (right).

$H_i \bar{X} \leq K_i$ . Thus, the  $q$  control laws can be merged into a *single* piecewise affine function

$$u = F_j^i \bar{X}(k) + G_j^i, \quad (36a)$$

$$i, j : H_i \bar{X}(k) \leq K_i, \quad (36b)$$

$$H_j^i \bar{X}(k) \leq K_j^i, \quad (36c)$$

where (36c) selects the control law that is active, (36b) selects the control law region that is currently active, and (36a) is evaluated to obtain the control input.

To construct the explicit solution, we choose the implementation of MPC controller based on (14) that gives  $q = 16$ , and hence, 16 control laws are merged in (36). The 16 piecewise affine feedback control laws are obtained by the 16 choices for the signs<sup>†</sup> of  $\bar{C}_{43}$ ,  $\bar{C}_{44}$ ,  $\bar{C}_{45}$ , and  $\bar{C}_{46}$ .

For our MPC controller, the total number of polyhedral regions in (36) for the case where all four components of the matrix  $\bar{C}$  switch (this is 16 control laws case) is 3068. Figure 13 shows the cross-section of the regions in  $x - y$  plane, computed for zero velocity and docking port located at  $(2.5, 0)$ . The number of regions appears to be acceptable and suggests that explicit MPC solutions can be feasible for automation of spacecraft rendezvous and proximity maneuvers. Furthermore, the complexity of the polyhedral partitioning can be reduced by eliminating small regions and expanding the neighboring ones. In fact, it was verified that about a third of the regions have Chebyshev radius smaller than  $10^{-3}$  and can be eliminated. The elimination of these regions induces minimal perturbations to the closed-loop system, and these perturbations are restricted only to area of the removed regions. Also, in general, the controller uses most frequently (i.e., more than 99% of times) a much smaller subset of regions, usually 10%–20% of the total, which can be identified

<sup>†</sup>Here, we apply a slightly modified definition of  $\text{sgn}(\cdot)$ , where  $\text{sgn}(0) = 1$ .

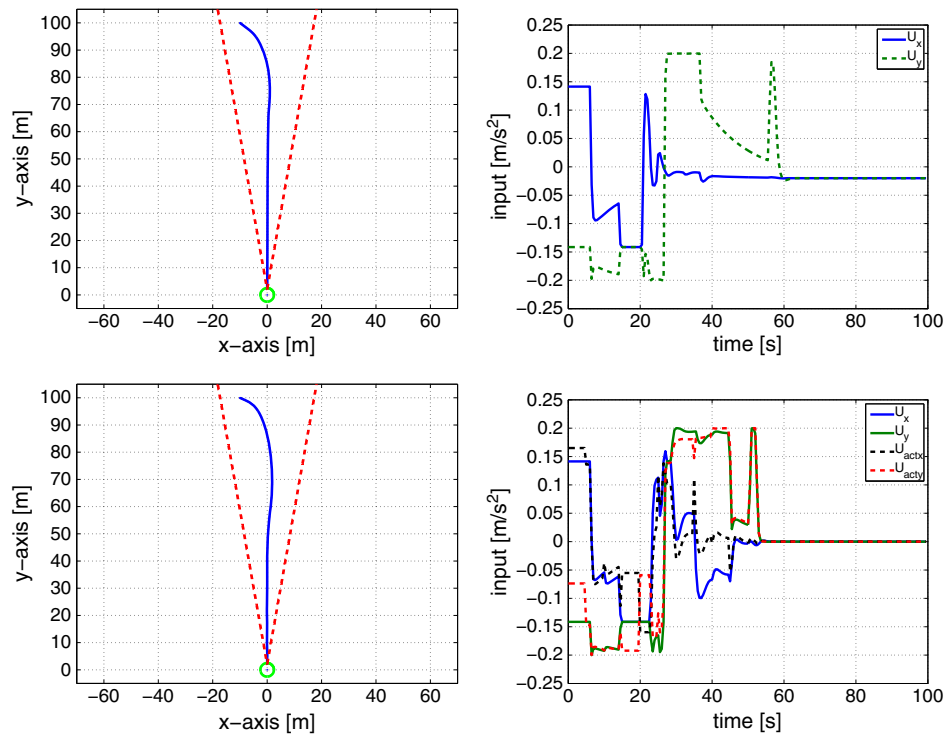


Figure 11. In-track approach subject to disturbances. Top row: constant disturbance in  $x$ - $y$  direction. Closed-loop trajectory (left) and closed-loop control accelerations (right). Bottom row: random magnitude disturbance in  $x$ - $y$  direction. Closed-loop trajectory (left) and closed-loop control accelerations (solid) and actuated accelerations (dash) (right).

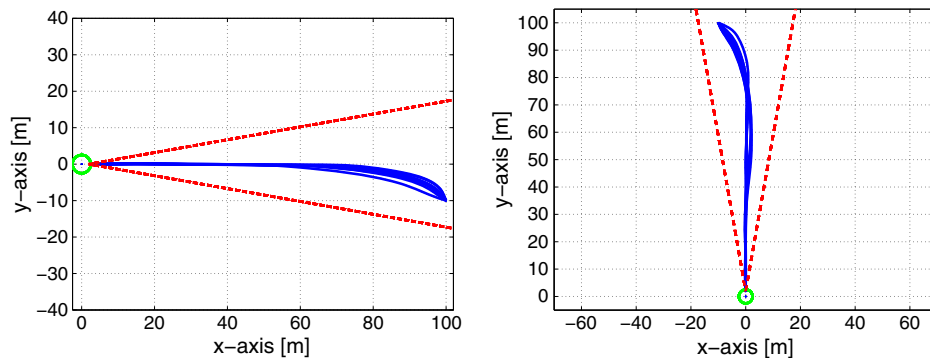


Figure 12. Repeated simulations with random disturbances on thrust actuation. Closed-loop radial approach trajectories (left) and closed-loop in-track approach trajectories (right).

by extensive simulations. With these approaches, the controller data memory requirements can be reduced to fit the platform computational resources.

If instead we use simplified constraints (16), we can reduce the effort for computation and storage to  $q = 4$  piecewise affine feedback control laws. A minor drawback with this approach is that the parameter vector in the multiparametric programming algorithm (see [20]) needs to include the variable  $\zeta(k)$  defined in (17) and hence the feedback law has one extra dimension.

### Remark 3

It is important to briefly recall the benefits that can be obtained by deploying the explicit MPC feedback law in complex embedded control systems [37,38], rather than the QP-based algorithm.

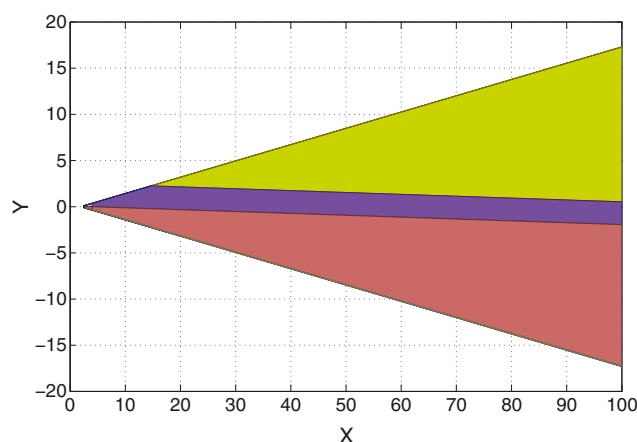


Figure 13. Regions of the explicit Model Predictive Controller.

From a control design perspective, the evaluation of the explicit MPC feedback law allows to characterize the nominal closed-loop dynamics and hence allows for a more detailed analysis of the closed-loop system properties [39]. From a platform requirements perspective, the explicit feedback is evaluated only through basic operations, that is, sums, products, and comparisons to 0, which only requires standard hardware circuits and which can be implemented even in fixed-point numbers. Also, explicit MPC, while requiring a possibly consistent amount of storage memory, requires minimum amount of RAM, as opposed to high-performance QP algorithms, where it shall be noted that more and more, the amount of memory is the limiting constraint in complex embedded control systems [39,40]. Finally, from a verification perspective, it is much simpler to verify a lookup table of gains and to bound the worst-case execution time, instead of a QP algorithm that executes from an infinite set of initial conditions. At the same time, the implementation with the on-board QP solver, if feasible from computations and verification standpoint, can improve the reconfigurability of the MPC controller to changes in parameters.

## 6. SIMULATED APPROACH OF A ROTATING PLATFORM

The same parameters as in Section 5 were used to simulate the approach to a rotating platform. In (14),  $\beta = \frac{2.5+\omega_p}{10}$  was made dependent on  $\omega_p$  to relax the constraint in the case of a faster rotating platform.

In this section, we consider the case when the spacecraft approaches the platform rotating in the counterclockwise direction. We examine two cases. In the first case, while the platform rotates, the changes in LOS cone constraints due to rotation are not accounted for in prediction. In other words, the MPC controller assumes that the constraints remain frozen as they are at the current time instant. In the second case, the evolution of LOS constraints due to platform rotation is (approximately) predicted, as described in Section 4.2.

The initial position of the docking port is  $(r_{x0}, r_{y0}) = (2.5, 0)$  (m), and the spacecraft starts from a representative initial condition  $(\delta x_0, \delta y_0) = (50, 5)$  (m) in the interior of the initial LOS cone. Such an initial condition and LOS cone position correspond to a radial approach if the platform were not rotating. We simulate the maneuvers for a lower platform angular rate of  $\omega_p = 0.6$  deg/s and a higher platform angular rate of  $\omega_p = 2.25$  deg/s.

For  $\omega_p = 0.6$  deg/s, Figures 14 and 15 show that maneuvers can be successfully completed regardless of whether the prediction of platform motion and of the LOS cone is performed or not.

Table IV compares the fuel consumption metrics and the docking time for the maneuvers with and without platform motion prediction as the control weighting  $\alpha$  varies. For lower control weighting values, prediction reduces fuel consumption (about 15% for  $\alpha = 10^2$ ). The time-to-dock is

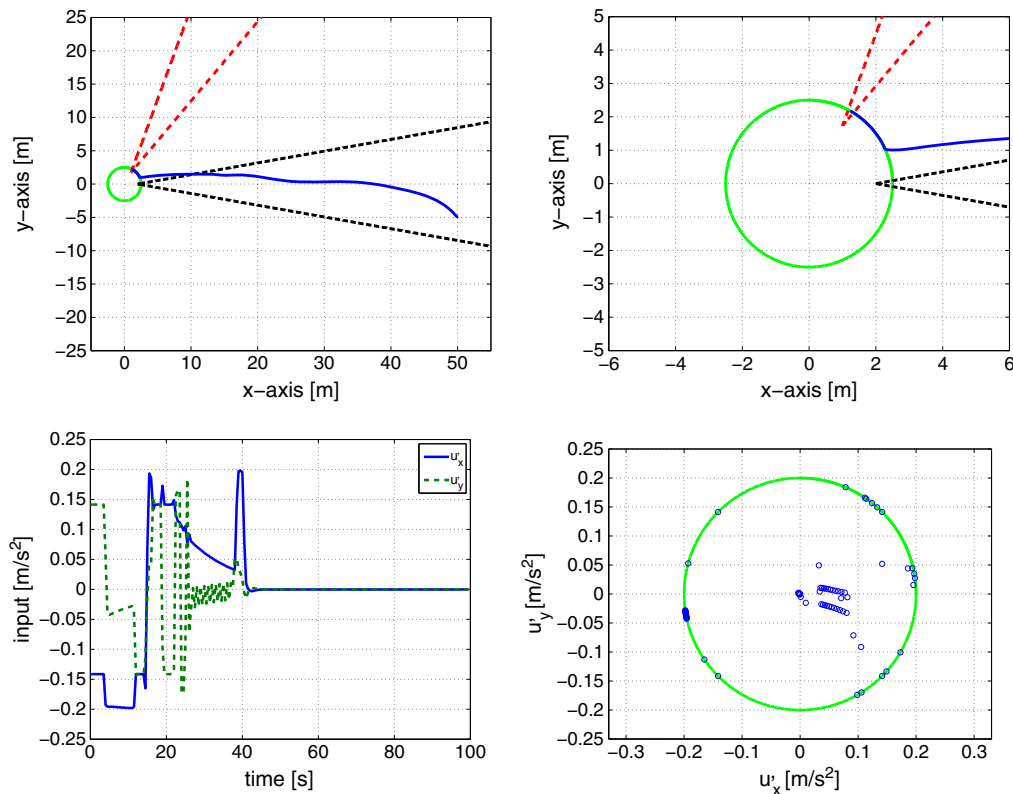


Figure 14. Radial approach to a platform rotating at  $\omega_p = 0.6$  deg/s without prediction of platform motion. Top row: trajectory on the  $x$ - $y$  plane (left) and zoomed-in trajectory (right). Bottom row: control accelerations  $u'_x$  and  $u'_y$  (left) and the plot of  $u'_y$  versus  $u'_x$  with magnitude saturation (right). Initial position of the Line-of-Sight cone is designated by black dashed lines and final position by the red dashed lines.

approximately the same for the two controllers. The fuel consumption difference is eroded as control weighting increase, and both controllers start to strongly emphasize small control effort and low fuel consumption. We have also found that this difference is also dependent on the initial position of the spacecraft within LOS cone and may be eroded when the spacecraft is close to 'active' constraint boundary. Qualitatively, similar conclusions about the fuel consumption benefit of predicting the platform motion are obtained at angular rates lower and slightly higher than 0.6 deg/s.

Figure 16 compares the spacecraft trajectories when prediction of the platform motion is employed versus when it is not employed for higher  $\omega_p = 2.25$  deg/s,  $\alpha = 10^2$ , and  $(\delta x_0, \delta y_0) = (50, 5)$  (m). The spacecraft is able to successfully perform the maneuver with the prediction of platform motion, but it is not able to keep up and eventually violates the constraints (e.g., collides with the platform at time 62 s) when such prediction is not employed. Note that the control inputs remain saturated at the limits in the case of no prediction, as the controller attempts to keep up with the constraints.

Figure 17 illustrates the responses for the case of the radial approach and the in-track approach with  $\omega_p = 0.6$ , and  $\omega_p = 2.25$  while  $\alpha = 1 \times 10^2$  when the initial conditions of the spacecraft vary. The spacecraft is able to successfully complete the maneuvers with the platform motion prediction for both lower and higher angular rates. We note that for  $\omega_p > 2.25$  deg, the spacecraft may fail to complete the maneuvers even with the platform motion prediction as actuator authority may be insufficient to keep up with a rapidly rotating platform in this case.

To summarize, our simulation results here and in [23, 24] suggest that incorporating the prediction of the platform motion and changes in LOS cone can result in more fuel efficient maneuvers and in being able to complete the maneuvers for higher rotational rate of the platform. These benefits

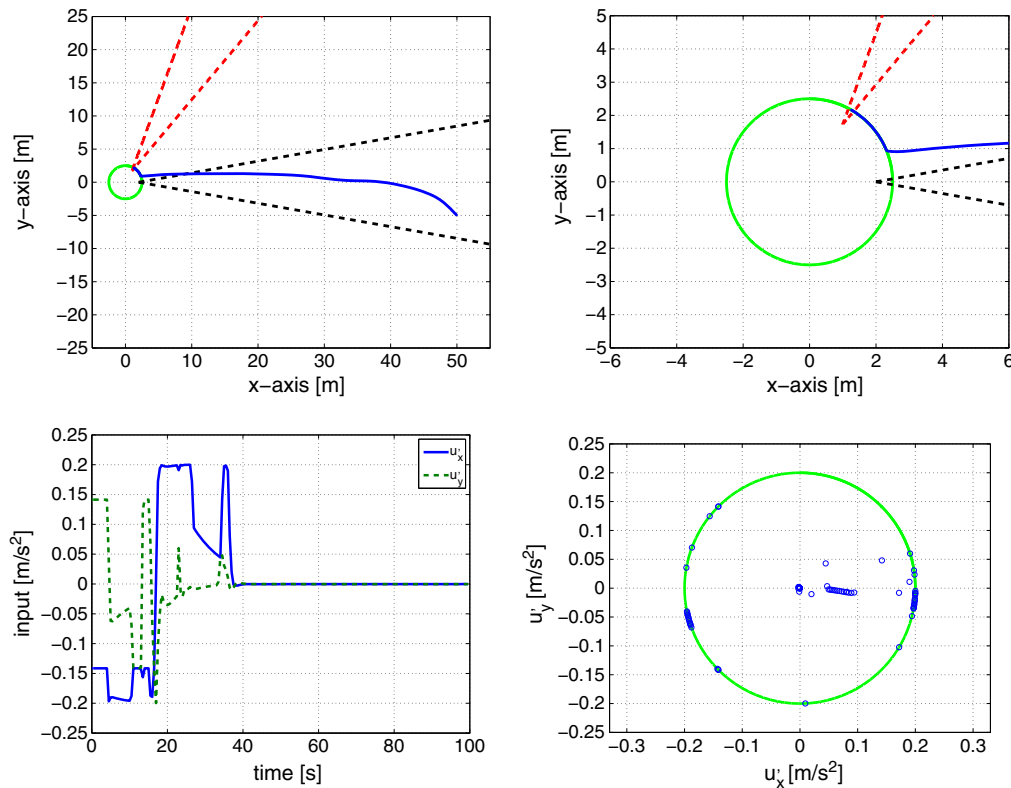


Figure 15. Radial approach platform rotating at  $\omega_p = 0.6$  deg/s with prediction of platform motion. Top row: trajectory on the  $x$ - $y$  plane (left) and zoomed-in trajectory (right). Bottom row: control accelerations  $u'_x$  and  $u'_y$  (left) and the plot of  $u'_y$  versus  $u'_x$  with magnitude saturation (right). Initial position of the Line-of-Sight cone is designated by black dashed lines and final position by the red dashed lines.

Table IV. Fuel consumption-related metrics and docking time versus  $\alpha$ .

$\omega_p = 0.6$ deg/s	$\alpha$	$10^2$	$10^3$	$10^4$	$10^5$	$10^6$	$10^7$	$5 \times 10^7$	$10^8$	$5 \times 10^8$
Nonpredicted constraints	$J_1$	17.44	18.16	15.61	17.74	15.66	14.28	13.49	13.16	12.55
	$J_2$	2.36	2.50	2.14	2.41	2.15	1.88	1.79	1.76	1.63
	$J_3$	13.33	13.94	12.18	13.59	12.29	11.51	11.20	11.10	10.64
	$t_d$ (s)	40.50	40.50	40.50	40.50	40.50	42.00	51.00	57.50	63.50
Predicted constraints	$J_1$	14.35	14.34	14.28	14.38	14.04	13.65	13.12	12.94	12.13
	$J_2$	1.93	1.93	1.92	1.93	1.87	1.76	1.72	1.70	1.54
	$J_3$	11.57	11.57	11.54	11.58	11.41	11.17	11.00	10.91	10.27
	$t_d$ (s)	40.50	40.50	40.50	40.50	41.00	42.00	51.50	57.50	63.50

are more pronounced for medium range of  $\omega_p$ , and are eroded for very low values of  $\omega_p$  and as  $\omega_p$  increases to larger values, which exceed the actuators capabilities.

## 7. COLLISION AVOIDANCE MANEUVERS

In this section, we consider the additional objective of avoiding debris on the spacecraft rendezvous path. There are more than 22,000 debris of 10 cm and longer orbiting the Earth today, and this number is growing. Collision with orbital debris is a serious threat that can damage the spacecraft. Several collision risk assessment methods have been developed, see for example, [41–45] and references therein, along with debris collision avoidance strategies, see for example, [41, 46, 47] and



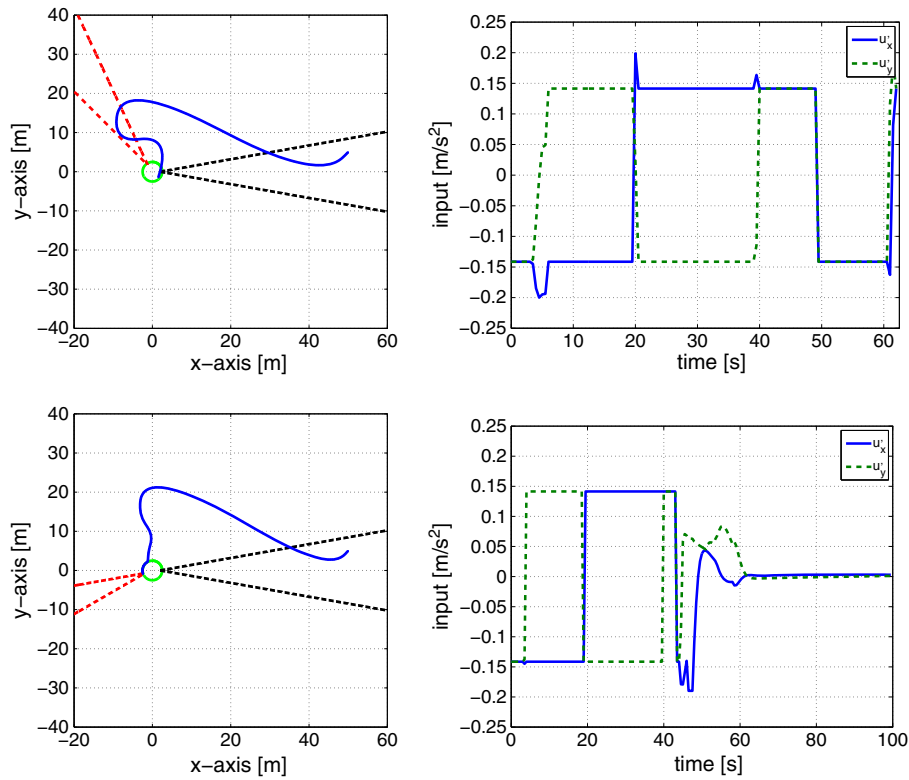


Figure 16. Radial approach to a platform rotating at with  $\omega_p = 2.25$  deg/s. Top row: plots without prediction of platform motion. Trajectory on the  $x$ - $y$  plane (left) and control accelerations  $u'_x$  and  $u'_y$  (right). Bottom row: plots with prediction of platform motion. Trajectory (left) and control accelerations (right). Initial position of the Line-of-Sight cone is designated by black dashed lines and final position by the red dashed lines.

references therein. In [46, 48], for instance, collision avoidance strategies for polyhedral objects as obstacles have been developed on the basis of mixed-integer linear programming.

To incorporate debris avoidance in our MPC approach, we assume that the debris can be covered by a virtual disk of radius  $r_d$  centered at  $(d_x, d_y)$  (m). See Figure 18.

### 7.1. Model Predictive Controller design for debris avoidance

Our approach to debris avoidance is based on covering the debris by a disk and assuming that this 'virtual' disk slowly rotates with angular rate  $\omega_d$  (rad/s). Referring to Figure 18, we impose the constraint forcing the spacecraft to remain in a specified half-plane relative to a tangent line to the disk. As the tangent line rotates with the disk, the constraint is dynamically reconfigured and varies in time. For simplicity, we assume here that the docking port does not rotate, and it is at the origin of the reference frame, that is,  $(r_x, r_y) = (0, 0)$ ,  $(\sigma_x, \sigma_y) = (\delta x, \delta y)$ .

At the activation of the constraint, the disk tangent line is perpendicular to the line between the spacecraft location,  $(\delta x(0), \delta y(0))$  (m), and the center of the disk,  $(d_x, d_y)$ . The angle  $\varphi_d(0)$  is defined as the angle between the  $x$ -axis and the normal to the tangent line so that  $\varphi_d(0) = \tan^{-1} \left( \frac{\delta y(0) - d_y}{\delta x(0) - d_x} \right)$ . Then,  $\varphi_d(k+1) = \varphi_d(k) + \omega_d k T_s$ , and the debris avoidance constraint is given by

$$\frac{\cos \varphi_d(k)}{r_d} (\delta x(k) - d_x) + \frac{\sin \varphi_d(k)}{r_d} (\delta y(k) - d_y) \geq 1. \quad (37)$$

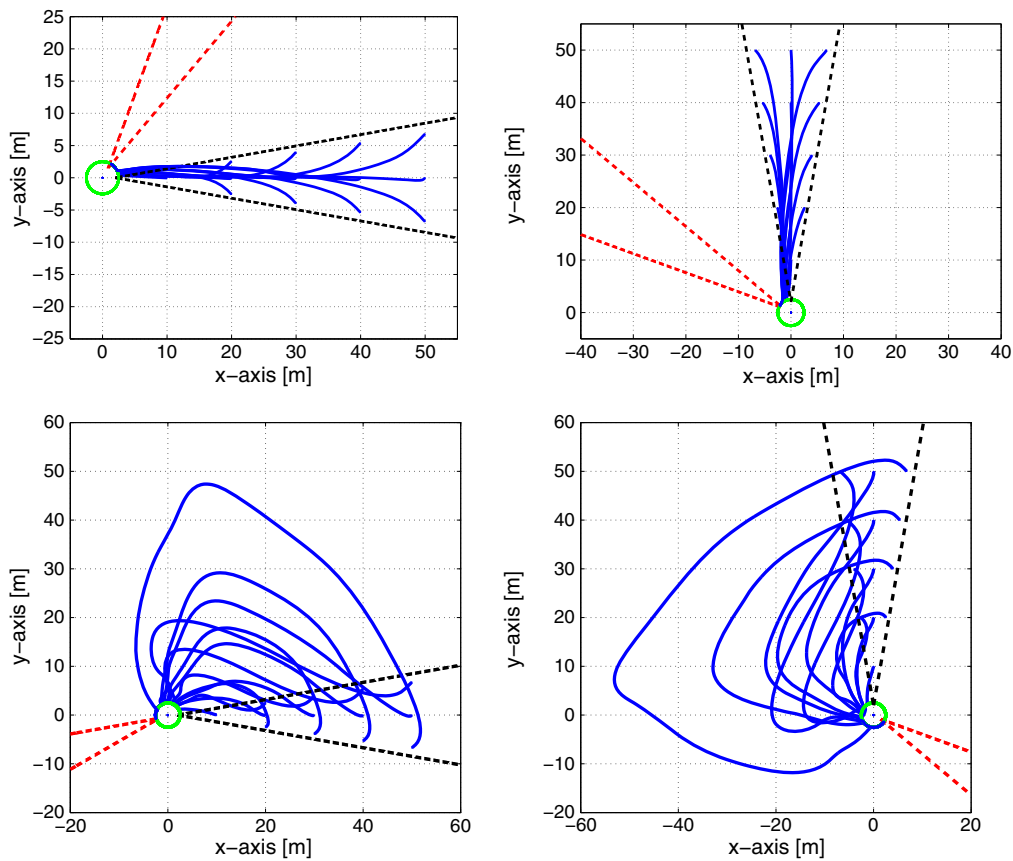


Figure 17. Trajectory on the  $x$ - $y$  plane starting from various initial positions in a rotating platform with prediction of platform motion. Top row:  $\omega_p = 0.6$  deg/s. Radial approach (left) and in-track approach (right). Bottom row:  $\omega_p = 2.25$  deg/s. Radial approach (left) and in-track approach (right). Initial position of the Line-of-Sight cone is designated by black dashed lines and final position by the red dashed lines.

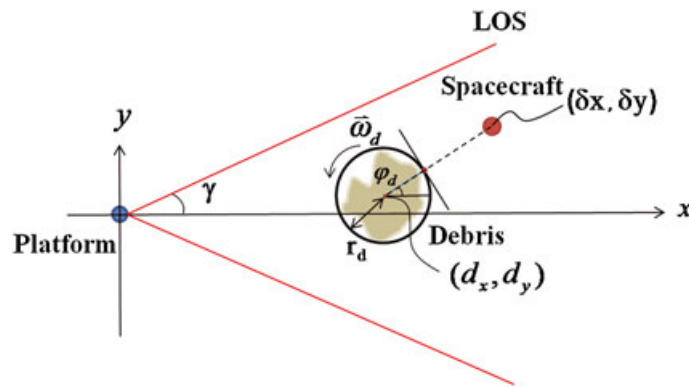


Figure 18. Schematics of the approach used to achieve debris avoidance.

A similar approach to the one in Section 6 is applied so that (37) is approximated in prediction by

$$L_7(\delta x(j|k) - d_x) + L_8(\delta y(j|k) - d_y) + (-L_8(\delta x(k) - d_x) + L_7(\delta y(k) - d_y))(\varphi_d(j|k) - \varphi_d(k)) \geq 1, \quad (38)$$

where

$$L_7 = \frac{\cos \varphi_d(k)}{r_d}, \quad L_8 = \frac{\sin \varphi_d(k)}{r_d},$$

$$\varphi_d(j|k) \simeq \varphi_d(k) + \dot{\varphi}_d(k)jT_s,$$

and  $j \in \mathbb{Z}_{0+}$  denotes a future instant with respect to  $k$ . Debris constraint (38) is deactivated once  $\varphi_d$  becomes equal to  $\varphi_d(0) + \pi$  so that the constraint does not interfere with the spacecraft motion after it passes the debris.

Similarly to (28) and (29), the state vector for debris avoidance maneuver has the form

$$\hat{X} = (\delta x \quad \delta y \quad \delta \dot{x} \quad \delta \dot{y} \quad d_x \quad d_y \quad r_{d_x} \quad r_{d_y} \quad z_1 \quad z_2)^T,$$

and the model is represented by

$$\hat{X}(j+1|k) = \hat{A}\hat{X}(j|k) + \hat{B}\hat{U}(j|k), \quad (39a)$$

$$\hat{Y}(j|k) = \hat{C}\hat{X}(j|k) + \hat{D}\hat{U}(j|k), \quad (39b)$$

where

$$\hat{A} = \begin{pmatrix} A_d & \mathbf{0}_{4 \times 2} & \mathbf{0}_{4 \times 2} & \mathbf{0}_{4 \times 2} \\ \mathbf{0}_{2 \times 4} & I & \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 4} & \mathbf{0}_{2 \times 2} & \Omega_d & \mathbf{0}_{2 \times 2} \\ \mathbf{0}_{2 \times 4} & \mathbf{0}_{2 \times 2} & \mathbf{0}_{2 \times 2} & \Theta \end{pmatrix}, \quad \hat{B} = \bar{B}, \quad \hat{D} = \bar{D},$$

$\hat{U} = \bar{U}$ ,  $\Theta$  is defined in (27),

$$\hat{C} = \begin{pmatrix} L_7 & L_8 & 0 & 0 & -L_7 & -L_8 & 0 & 0 & 0 & \hat{C}_1 \\ \sin \gamma & -\cos \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \sin \gamma & \cos \gamma & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hat{C}_2 & \hat{C}_3 & \hat{C}_4 & \hat{C}_5 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

and

$$\Omega_d = \begin{pmatrix} \cos(\omega_d T_s) & -\sin(\omega_d T_s) \\ \sin(\omega_d T_s) & \cos(\omega_d T_s) \end{pmatrix},$$

$$\hat{C}_1 = L_8(\delta x(k) - d_x) + L_7(\delta y(k) - d_y),$$

$$\hat{C}_2 = \text{sgn}(\delta x(k)), \quad \hat{C}_3 = \text{sgn}(\delta y(k)),$$

$$\hat{C}_4 = -\eta \text{sgn}(\delta \dot{x}(k)), \quad \hat{C}_5 = -\eta \text{sgn}(\delta \dot{y}(k)),$$

and where the constraint

$$\hat{Y}(j|k) \geq \hat{Y}_{\min}, \quad \hat{Y}_{\min} = (1 \quad 0 \quad 0 \quad -\beta)^T, \quad (40)$$

is imposed on the system output, representing the LOS constraints, the soft-docking constraint with respect to the platform, and the debris avoidance constraint.

Thus, for the debris avoidance case, the MPC optimal control problem is formulated as

$$\min_{\hat{\mathbf{U}}(k)} \hat{\mathbf{X}}(N_J|k)^T \hat{\mathbf{P}} \hat{\mathbf{X}}(N_J|k) + \sum_{j=0}^{N_J-1} \hat{\mathbf{X}}(j|k)^T \hat{\mathbf{Q}} \hat{\mathbf{X}}(j|k) + \hat{\mathbf{U}}(j|k)^T \hat{\mathbf{R}} \hat{\mathbf{U}}(j|k), \quad (41a)$$

$$\text{s.t. } \hat{\mathbf{X}}(j+1|k) = \hat{\mathbf{A}} \hat{\mathbf{X}}(j|k) + \hat{\mathbf{B}} \hat{\mathbf{U}}(j|k), \quad (41b)$$

$$\hat{\mathbf{Y}}(j|k) = \hat{\mathbf{C}} \hat{\mathbf{X}}(j|k) + \hat{\mathbf{D}} \hat{\mathbf{U}}(j|k), \quad (41c)$$

$$\hat{\mathbf{X}}(0|k) = \hat{\mathbf{X}}(k), \quad (41d)$$

$$\hat{\mathbf{U}}(j|k) = \hat{\mathbf{K}} \hat{\mathbf{X}}(j|k), \quad j = N_U + 1, \dots, N_J - 1, \quad (41e)$$

$$\hat{\mathbf{Y}}(j|k) \geq \hat{\mathbf{Y}}_{\min}(k), \quad j = 0, \dots, N_C, \quad (41f)$$

$$\hat{\mathbf{U}}(j|k) \geq \hat{\mathbf{U}}_{\min}, \quad j = 0, \dots, N_U, \quad (41g)$$

$$\hat{\mathbf{U}}(j|k) \leq \hat{\mathbf{U}}_{\max}, \quad j = 0, \dots, N_U, \quad (41h)$$

where  $\hat{\mathbf{U}}(k) = \{\hat{\mathbf{U}}(0|k), \dots, \hat{\mathbf{U}}(N_U|k)\}$ ,

$$\hat{\mathbf{P}} = \begin{pmatrix} \mathbf{P} & \mathbf{0}_{4 \times 6} \\ \mathbf{0}_{6 \times 4} & \mathbf{0}_{6 \times 6} \end{pmatrix}, \quad \hat{\mathbf{Q}} = \begin{pmatrix} \mathbf{Q} & \mathbf{0}_{4 \times 6} \\ \mathbf{0}_{6 \times 4} & \mathbf{0}_{6 \times 6} \end{pmatrix}, \quad \hat{\mathbf{R}} = \bar{\mathbf{R}}, \quad \hat{\mathbf{K}} = \begin{pmatrix} \mathbf{K} & \mathbf{0}_{2 \times 6} \end{pmatrix},$$

and the input constraints in (41) are defined by (10).

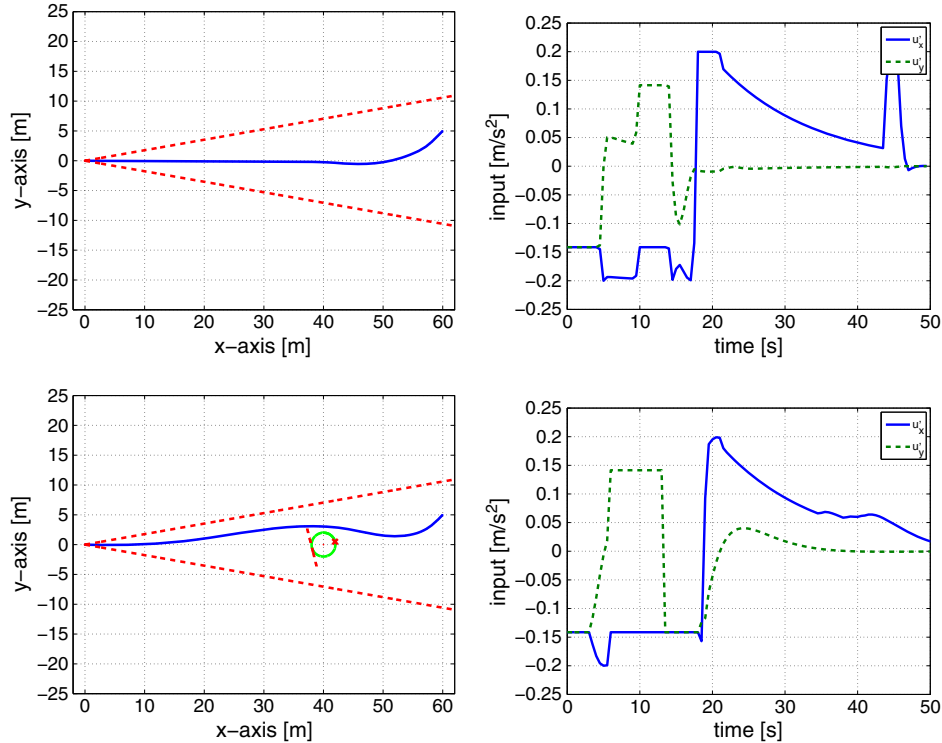


Figure 19. Comparison of the maneuvers. Top row: trajectory on the  $x$ - $y$  plane without debris (left) and control accelerations  $u'_x$  and  $u'_y$  without debris (right). Bottom row: trajectory on the  $x$ - $y$  plane with debris (left) and control accelerations  $u'_x$  and  $u'_y$  (right) with debris.

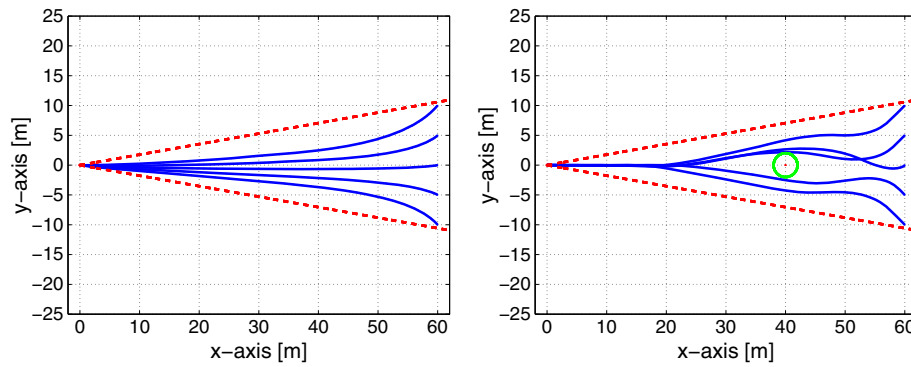


Figure 20. Trajectories from various initial positions without debris (left) and with debris (right).

Table V. Costs and docking time in maneuvers with and without debris avoidance.

	$J_1$	$J_2$	$J_3$	$t_d$
Without debris	17.88	3.03	14.77	45.50
With debris	21.03	3.53	16.18	53.00

## 7.2. Simulation results

Simulation results are now presented for  $r_d = 2$  m. Other parameters are the same as the ones in Section 6. The debris is located at  $(d_x, d_y) = (40, 0)$  (m), and the initial location of the spacecraft is  $(\delta x_0, \delta y_0) = (60, 5)$  (m). The docking port is located at the origin of the reference frame and is represented here as a point mass. The angular rate of the ‘virtual’ disk,  $\omega_d$ , was varied between 5 deg and 15 deg, and the best value of  $\omega_d = 12$  deg/s was determined in simulations on the basis of a trade-off between maneuver feasibility (in terms of satisfying the imposed constraints) and speed. Figure 19 compares the trajectories of the spacecraft when there is no debris and when the spacecraft performs a rendezvous while avoiding debris. Note that the debris is placed at a location that makes infeasible the trajectory of the no-debris case. The ‘x’ symbol represents the initial point of activation of the tangent line constraint. The constraint is deactivated after the disk covering the debris has rotated  $\pi$  rad. The line tangent to the virtual disk in Figure 19 is in the position after the disk rotated  $\pi$  rad, and the debris avoidance constraint is deactivated. Figure 20 illustrates debris avoidance maneuvers for various initial positions of the spacecraft.

We also report the three fuel consumption-related metrics in (32) and the time-to-dock for the case with and without the debris in Table V. The debris avoidance results in increased fuel consumption and longer maneuver time.

## 8. CONCLUSIONS

In this paper, we have considered rendezvous and proximity maneuvering between a spacecraft and (a potentially rotating/tumbling) target platform. MPC was utilized to enforce velocity of approach (soft-docking), LOS cone, and thrust magnitude constraints. The problem was treated in an LQ MPC framework without introducing integer variables so that the finite horizon optimization problem can be solved by QP algorithms, which are known to be efficient and computationally affordable. To achieve this, we have employed *dynamically reconfigurable* linear constraints. We have demonstrated through simulations that the MPC controller is capable of planning efficient maneuvers, while enforcing all the imposed constraints, and of compensating for the effects of significant disturbances such as those that may be caused by thrust errors, air drag, or solar pressure. In addition, we have analyzed the dependence of fuel consumption and ‘time-to-dock’ on the weights of the MPC cost function by using three different metrics. The weights of the MPC cost function, for



example, the scaling factor on the control weighting matrix, have been shown to have a direct influence over the trade-off between the fuel consumption and the docking time. For the case when the platform is not rotating and the LOS cone angle is fixed and known, an explicit MPC solution can be generated offline for on-board use, hence, avoiding the need for the deployment and validation of the QP solver in the spacecraft software. For the case when the target platform is rotating, our results suggest that predicting the motion of the docking port and of the associated changes in the LOS constraint permits to perform maneuvers initiated when the spacecraft is further away from the platform and when the platform is rotating/tumbling at a higher rate. We have also shown that by incorporating such a prediction into the MPC problem formulation, the fuel consumption is reduced. Finally, we have demonstrated that related ideas can be applied for debris avoidance during the rendezvous maneuver. The extensions to more general in and out-of-plane spacecraft motion and to three dimensional spacecraft and platform rotations will be pursued in future works.

## REFERENCES

1. Goodman J. History of space shuttle rendezvous and proximity operations. *Journal of Spacecraft and Rockets* 2006; **43**(5):944–959.
2. Polites M. Technology of automated rendezvous and capture in space. *Journal of Spacecraft and Rockets* 1999; **36**(2):280–291.
3. Woffinden D, Geller D. Navigating the road to autonomous orbital rendezvous. *Journal of Spacecraft and Rockets* 2007; **44**(4):898–909.
4. Aldrin E, Jr. Line-of-sight guidance techniques for manned orbital rendezvous. *Sc.D. Thesis*, Massachusetts Institute of Technology, USA, 1963.
5. Breger L, How J. Safe trajectories for autonomous rendezvous of spacecraft. *Journal of Guidance, Control, and Dynamics* 2008; **5**:31.
6. Clark F, Spehar P, Brazzel J, Hinkel H. Laser-based relative navigation and guidance for space shuttle proximity operation. *Advances in the Astronautical Sciences* 2003; **113**:732–737.
7. Fehse W. *Automated Rendezvous and Docking of Spacecraft*. Cambridge Aerospace Series: Cambridge, UK, 2003.
8. Bryson A. Linear feedback solutions for minimum effort interception, rendezvous, and soft landing. *AIAA Journal* 1965; **3**(8):1542–1544.
9. Coverstone-Carroll V. Detumbling and reorienting underactuated rigid spacecraft. *Journal of Guidance, Control, and Dynamics* 1996; **3**:19.
10. Clohessy W, Wiltshire R. Terminal guidance system for satellite rendezvous. *Journal of Aerospace Sciences* 1960; **27**(9):653–658.
11. Luo Y, Tang G. Spacecraft optimal rendezvous controller design using simulated annealing. *Aerospace Science and Technology* 2005; **9**:732–737.
12. Miele A, Weeks M, Circia M. Optimal trajectories for spacecraft rendezvous. *Journal of Optimization Theory and Applications* 2007; **132**(3):353–376.
13. Kong E, Nolet S, Miller D. Autonomous docking algorithm development and experimentation using the SPHERES testbed. *Proceedings of SPIE Defense and Security Symposium*, Orlando, Florida, 2004.
14. Richards A, How J. Performance evaluation of rendezvous using model predictive control. *AIAA Guidance, Navigation, and Control Conference and Exhibit, AIAA Paper 2003-5507*, Austin, Texas, 2003; 1–9.
15. Hartley E. Model predictive control for spacecraft rendezvous. *Ph.D. Dissertation*, University of Cambridge, UK, 2010.
16. Di Cairano S, Bemporad A. Model predictive control tuning by controller matching. *IEEE Transactions on Automatic Control* 2010; **55**(1):185–190.
17. Singh L, Bortolami S, Page L. Optimal guidance and thruster control in orbital approach and rendezvous for docking using model predictive control. *AIAA Guidance, Navigation, and Control Conference and Exhibit, AIAA Paper 2010-7754*, Toronto, Canada, 2010; 1–15.
18. Holzinger M, DiMatteo J, Schwartz J, Milam M. Passively safe receding horizon control for satellite proximity operations. *47th IEEE Conference on Decision and Control*, Cancun, Mexico, 2008; 3433–3440.
19. Nocedal J, Wright S. *Numerical Optimization*. Springer: New York, 2006.
20. Bemporad A, Morari M, Dua V, Pistikopoulos E. The explicit linear quadratic regulator for constrained systems. *Automatica* 2002; **38**(1):3–20.
21. Borrelli F, Bemporad A, Morari M. Model predictive control base on linear programming - the explicit solutions. *IEEE Transactions on Automatic Control* 2002; **47**(12):1974–1985.
22. Park H, Di Cairano S, Kolmanovsky I. Model Predictive Control of spacecraft docking with a non-rotating platform. *IFAC 18th World Congress*, Milano, Italy, 2011; 8485–8490.
23. Park H, Di Cairano S, Kolmanovsky I. Model Predictive Control for spacecraft rendezvous and docking with a rotating/tumbling platform and for debris avoidance. *American Control Conference*, San Francisco, California, 2011; 1922–1927.

24. Park H, Di Cairano S, Kolmanovsky I. Linear quadratic Model Predictive Control approach to spacecraft rendezvous and docking. *21st AAS/AIAA Space Flight Mechanics Meeting*, New Orleans, Louisiana, 2011.
25. Clarke D, Mohtadi C, Tuffs P. Generalized predictive control—part I the basic algorithm. *Automatica* 1987; **23**(2):137–148.
26. Clarke D, Mohtadi C, Tuffs P. Generalized predictive control—part II extensions and interpretations. *Automatica* 1987; **23**(2):149–160.
27. Wie B. *Spacecraft dynamics and control*, 2010. 2nd Edition AIAA.
28. IBM ILOG CPLEX V12.1: User's manual for CPLEX, 2009. International Business Machines Corporation.
29. Boyd S, Vandenberghe L. *Convex Optimization*. Cambridge University Press: Cambridge, UK, 2004.
30. Di Cairano S, Bemporad A, Kolmanovsky I, Hrovat D. Model predictive control of magnetically actuated mass spring damper for automotive applications. *International Journal of Control* 2007; **80**(11):1701–1716.
31. Maciejowski J. *Predictive Control with Constraints*. Prentice Hall: London, 2000.
32. Mayne DQ, Rawlings JB, Rao CV, Scokaert POM. Constrained model predictive control: stability and optimality. *Automatica* 2000; **36**(6):789–814.
33. Mayne DQ, Seron MM, Raković SV. Robust model predictive control of constrained linear systems with bounded disturbances. *Automatica* 2005; **41**(2):219–224.
34. Kothare MV, Balakrishnan V, Morari M. Robust constrained model predictive control using linear matrix inequalities. *Automatica* 1996; **32**(10):1361–1379.
35. Bemporad A, Morari M. Robust model predictive control: a survey. In *Robustness in Identification and Control*, Vol. 245, Garulli A, Tesi A (eds), Lect. Notes in Control and Inf. Sciences: Springer, London, 1999; 207–226.
36. Di Cairano S, Tseng H, Bernardini D, Bemporad A. Steering vehicle control by switched model predictive control. *6th IFAC Symposium on Advances in Automotive Control*, Munich, Germany, 2010; 1–6.
37. Di Cairano S, Yanakiev D, Bemporad A, Kolmanovsky I, Hrovat D. An MPC design flow for automotive control and applications to idle speed regulation. *47th IEEE Conference on Decision and Control*, Cancun, Mexico, 2008; 5686–5691.
38. Di Cairano S, Tseng H. Driver-assist steering by active front steering and differential braking: design, implementation and experimental evaluation of a switched model predictive control approach. *49th IEEE Conference on Decision and Control*, Atlanta, Georgia, 2010; 2886–2891.
39. Di Cairano S, Yanakiev D, Bemporad A, Kolmanovsky IV, Hrovat D. Model predictive idle speed control: design, analysis, and experimental evaluation. *IEEE Transactions on Control Systems Technology* 2011; **20**(1):84–97.
40. Di Cairano S, Liang W, Kolmanovsky I, Kuang M, Phillips A. Engine power smoothing energy management strategy for a series hybrid electric vehicle. *American Control Conference*, San Francisco, CA, 2011; 2101–2106.
41. Foster J, Stansbery E. Fundamentals of debris collision avoidance. *54th International Astronautical Congress*, Bremen, Germany, 2003; 2558–2566.
42. Foster J, Frisbee J. Comparison of the exclusion volume and probability threshold methods for debris avoidance for the STS Orbiter and International Space Station, 2007. NASA Technical Paper.
43. Legendre P, Deguine B, Garmier R, Revelin B. Two line element accuracy assessment based on a mixture of Gaussian laws. *AIAA/AAS Specialists Conference, AIAA Paper 2006-6518*, Colorado, USA, 2006; 1–13.
44. Patera R. General method for calculating satellite collision probability. *Journal of Guidance, Control, and Dynamics* 2001; **24**(4):716–722.
45. Pelet L, Legendre P, Delavault S, Martin T. Detection of orbital debris collision risks for the automated transfer vehicle. *20th International Symposium on Space Flight Dynamics*, Maryland, USA, 2007.
46. Richards A, Schouwenaars T, How J, Feron E. Spacecraft trajectory planning with avoidance constraints using mixed-integer linear programming. *Journal of Guidance, Control, and Dynamics* 2002; **25**(4):755–764.
47. Sanchez-Ortiz N, Bello-Mora M, Klinkrad H. Collision avoidance manoeuvres during spacecraft mission lifetime: analysis of operational missions. *55th International Astronautical Congress, IAC Paper IAC-04-IAA.5.12.5.10*, Vancouver, Canada, 2004; 1–12.
48. Maia M, Galvao R. On the use of mixed-integer linear programming for predictive control with avoidance constraints. *International Journal of Robust and Nonlinear Control* 2009; **19**(7):822–828.