CEAS-GNC-2024-013

EuroGNC Navigation Guidance Control CEAS Bristol, UK June 11th-13th 2024

Mixed Integer Model Predictive Control for a free-floating platform with binary and continuous actuation

Franek Stark Master Student, Robotics and Autonomous Systems, Universität zu Lübeck, 23562 Lübeck, Germany and Robotics Innovation Center, Deutsches Forschungszentrum für Künstliche Intelligenz GmbH, 28359 Bremen, Germany. franek.stark@dfki.de

Shubham Vyas Reasercher, Robotics Innovation Center, Deutsches Forschungszentrum für Künstliche Intelligenz GmbH, 28359 Bremen, Germany and AG Robotik, Universität Bremen, 28359 Bremen, Germany shubham.vyas@dfki.de

Georg Schildbach Professor, Institute for Electrical Engineering in Medicine, Universität zu Lübeck, 23562 Lübeck, Germany georg.schildbach@uni-luebeck.de

Frank Kirchner Professor, Executive Director and Head of Robotics Innovation Center, Deutsches Forschungszentrum für Künstliche Intelligenz GmbH, 28359 Bremen, Germany and AG Robotik, Universität Bremen, 28359 Bremen, Germany frank.kirchner@dfki.de

ABSTRACT

This work develops a novel Mixed Integer Model Predictive Control (MIMPC) for European Space Agency (ESA)'s 3-dof free-floating platform which is actuated using on/off-thrusters that are subject to activation time constraints and a Reaction Wheel (RW). It compares a penalty-term, a Linear Complementarity Constraints (LCC), and a Mixed Integer (MI) based formulation to transcribe the on/off thrusters within the optimization problem. A set of linear constraints is presented to enforce the thruster time constraints. Analyses show that under the activation time constraints and real-time requirements, only the MI formulation provides a functional MPC controller. Hence, an MIMPC which directly controls the system's eight thrusters and RW is developed. Simulated results show that the controller can (sub-) optimally control and stabilize the system in real time for a short enough prediction horizon. By including the thruster's timing and on/off constraints, the controller is able to exploit the system's structure to provide efficient control.

Keywords: Model Predictive Control; Mixed Integer; Binary thrusters; Dwell time constraints; Space robots

Nomenclature

θ	=	Platform orientation (yaw) in world coordinates
$\dot{\theta}$	=	Platform angular velocity in world coordinates
x, y	=	Platform position in world coordinates
$\dot{x}, \dot{y}$	=	Platform velocity in world coordinates
ω_{RW}	=	Reaction Wheel (RW) rotational speed
$\mathbf{I}^{n \times n}$	=	Identity matrix of size $n \times n$

$\mathbf{0}^{n\times m}$	=	Zero matrix of size $n \times m$
$\mathrm{s}\theta, \mathrm{c}\theta$	=	Sine and cosine of θ
$\mathbf{x}, \mathbf{u}$	=	State and input vector of the system
$\hat{\mathbf{x}}$	=	Target state
F_n	=	Nominal thrust force applied by a single thruster
r	=	Platform radius
m, I_S	=	Platform overall mass and inertia on the z-axis.
I_RW	=	RW's inertia on the z-axis.
$\mathbf{x}_{j\|t}, \mathbf{u}_{j\|t}$	=	State and input prediction of time step j, predicted at time step t
N	=	Prediction Horizon
Δt	=	Discretization time of the system dynamics within the controller
$\mathbf{U}_t, \mathbf{X}_t$	=	Set of all predicted states and inputs at time step t
$\mathbf{u}_{\mathrm{bin},t}$	=	Set of all predicted binary inputs at time step t
$\mathbf{Q}, \mathbf{W}$	=	Diagonal cost matrices for state and input cost terms.
t_s	=	Solver time to find the solution
$\hat{t}_s$	=	Maximal or desired solver time to find the solution, and hence also control period
$[[0, N]]$	=	Closed integer interval from 1 to N, i.e. $\{0, \dots, N\}$
$[[0, N))$	=	Right-open integer interval from 1 to N, i.e. $\{0, \dots, N - 1\}$

Acronyms

ESA European Space Agency. 1, 3, 16

LCC Linear Complementarity Constraints. 1, 3, 6–8, 10, 13
LCP Linear Complementarity Program. 3
LCQP Linear Complementarity Quadratic Program. 9
LP Linear Program. 6

MI Mixed Integer. 1, 6
MILP Mixed Integer Linear Program. 4, 7, 9–11, 13
MIMPC Mixed Integer Model Predictive Control. 1, 4, 11–13
MIP Mixed Integer Program. 3, 7, 9, 10
MPC Model Predictive Control. 3, 4, 6, 7, 9–11
MPCC Mathematical Program with Complementarity Constraints. 3, 4, 7–11, 13

NLP Non-Linear Program. 8

OCP Optimal Control Problem. 3, 6, 9–11
ORGL Orbital Robotics and GNC Lab. 3, 16

p-p peak-to-peak amplitude. 13
PWM pulse-width modulation. 3
PWPF pulse-width pulse-frequency. 3

QP Quadratic Program. 3, 7, 9–11, 13

REACSA REcap-ACrobat-SAtsim. 3–5, 9, 11, 12
RMS root-mean-square. 12, 13
RW Reaction Wheel. 1–5, 7, 9, 11–13

TVLQR Time-Varying Linear Quadratic Regulator. 3, 13

1 Introduction

Spacecraft rendezvous maneuvers are a pressing topic due to current issues such as space debris removal [1, 2]. To carry out such maneuvers and test the associated sub-assemblies on earth, air-bearing floating platforms are used [3]. European Space Agency (ESA)'s Orbital Robotics and GNC Lab (ORGL) features a 9 m × 5 m flat-floor for air-bearing satellite testbeds [4]. On top, the 200 kg heavy air-bearing platform REcap-ACrobat-SAtsim (REACSA) simulates a satellite [5]. REACSA is equipped with eight cold gas thrusters and a Reaction Wheel (RW) to be able to control its linear and angular acceleration. Like any system that uses thrusters and RWs, this experimental platform is subject to certain input constraints. Besides the RW's maximum applicable torque, its minimum and maximum angular velocity, the thrusters can only be on/off (binary) actuated. Thus, the thrusters either deliver full or no thrust. Moreover, due to physical limitations, a safe operation enforces certain activation time constraints: A thruster must remain switched on for a minimum time $t_{\text{on, min}}$ to build up repeatable and reliable thrust. Also, it must be turned off before a maximum activation time $t_{\text{on, max}}$ in order to comply with the pneumatic system and to prevent icing due to expanding gas. After an activation phase, a thruster has an additional off period $t_{\text{off, min}}$, while the buffer tank fills up again and the thruster nozzles warm up.

As in other approaches for control in space [6], the current controller for REACSA is a Time-Varying Linear Quadratic Regulator (TVLQR) with preliminary trajectory optimization. However, it does not know about the system constraints and hence, in reality, performs poorly [7]. To achieve optimal control while respecting system constraints, Model Predictive Control (MPC) has already achieved good results in spacecraft rendezvous application [8]. By explicitly modeling REACSA's thruster constraints in the Optimal Control Problem (OCP), the aim is to obtain a thrust-efficient control. Existing works on MPC for satellite-like platforms with on/off thrusters can be divided into different classes.

Continuous MPCs model the inputs with a continuous function and convert them in an additional, posterior step into binary values. Different works for combined attitude and position control of similar platforms like REACSA perform the conversion by using a Delta-Sigma Modulator [9], pulse-width modulation (PWM) [10] or simple trigger logic [11]. The MPC for only attitude control proposed by [12] gets a little closer to direct binary control by outputting the continuous PWM parameters. However, by assuming one firing cycle per prediction step it can not flexibly allocate binary thruster values. Another example is the attitude controller for the Brazilian Multimission Platform Satellite, presented in [13]. It converts the continuous inputs using a pulse-width pulse-frequency (PWPF) modulator.

Since optimal control by exploiting perfect thruster allocation that considers the timing constraints is desired, this work focuses on the second class: formulations that know about the binary inputs. This leads to the general class of MPCs with binary variables. Enforcing an input to be a binary variable makes the controller's underlying optimization problem significantly more complex since a non-convexity is introduced. Other works tackle this kind of optimization problem using three different methods:

1) A widely used approach is to define respective variables as integer variables and formulate the problem as a Mixed Integer Program (MIP).
2) A quadratic cost term penalizes non-binary values, while the binary variables are assumed to be continuous, and the problem is a Quadratic Program (QP) [14].
3) A constraint, called Linear Complementarity Constraints (LCC), enforces binary values [15]. The problem becomes a Mathematical Program with Complementarity Constraints (MPCC).

While for the MIP an appropriate solver is necessary, the QP can be solved by any (non-convex) QP solver. An MPCC can be solved with any non-linear solver [16], whereby there exist specialized solvers and techniques similar to MIP [17]. The LCCs appeared first in a Linear Complementarity Program (LCP) [18]. MPCCs have already been shown to work for binary decisions in the area of contact-implicit trajectory optimization [19]. However, binary decision problems are NP-hard and solvability in real-time depends strongly on the problem [20]. For the MIP formulation, there exist several works in similar context. In [21] an attitude Mixed Integer Linear Program (MILP) based control for entry vehicles with on/off thrusters is presented. It does not consider any timing constraints and no statement about the real-time capability is made. Timing constraints are considered in [22]: an attitude control with on/off thrusters under minimum on and off time constraints, assuming equal minimum on and off time. To be used as a real-time Mixed Integer Model Predictive Control (MIMPC), it is designed as an explicit MPC. In addition to that, in [23] an MILP based MPC whose objective is to minimize the thrust firings is presented. For a short prediction horizon, state-of-the-art solvers provide an approximate solution fast enough. However, they consider no thruster time constraints. A similar MILP approach with on/off thrusters that have a minimum impulse bit is developed in [24]. It is able to run in real-time on a modern computer. Although all these works explicitly model the on/off constraints of the thrusters thus belonging to the second class, they only consider attitude control. Only in [25] position is considered by reducing oscillations of a cable-driven platform with cold gas thrusters. However, no thruster time limits are considered, and the implementation is not real-time capable. Various works have successfully developed MIMPCs for satellites, using the integer variables for collision avoidance [26, 27]. For the penalty-term and MPCC formulation, to the authors' knowledge, there are no results for a similar application.

The literature refers to the minimum thruster on and off times as *minimum dwell time constraints* [28]. Only a few of the works from the space context listed above develop an MPC that takes these constraints into account. None of these works combine all three: minimum on, minimum off, and maximum on time constraints. An important fact to consider for this system is that with on/off engines it is impossible to stabilize the system to a steady state [29]. Instead, these systems are kept in a limit cycle around the target [21–23, 28].

The scope of this work is a comparison of the different formulations that enforce binary inputs. It compares which is the most appropriate in terms of real-time capability, while also respecting the timing constraints. Based on these results, it develops a novel real-time capable MILP based suboptimal MPC for free-floating platforms with on/off thrusters under timing constraints, using REACSA as an example. For that, a novel thruster timing constraint formulation is introduced. The paper is organized as follows: Section 2 introduces the two system models used in this work and studies REACSA's limit cycle. The *decoupled model* simplifies REACSA to only have four binary inputs and no RW. This model is used to evaluate the different binary constraint formulations and the timing constraints. The *coupled model* fully models REACSA's inputs and is used for the final controller, which is designed based on the findings of the analysis of the *decoupled model*. Section 3 introduces the MPC and explains the constraints in detail. The analysis of the different binary constraint formulations is done in Section 4, followed by simulation results of the final controller in Section 5. Section 6 provides a summary, discussion, and conclusion.

2 System Description

REACSA has one RW and eight pairwise arranged thrusters to apply force along the tangential lines (Figure 1b). A detailed system description can be found in [30]. The values of the system constants are given in Table 1.

Table 1 REACSA's physical properties

m	r	I_s	I_{RW}	$\omega_{RW,\max}$	F_n	$\tau_\max$	$t_\text{on,min}$	$t_\text{on, max}$	$t_\text{off,min}$
202.81 kg	0.35 m	12.22 kg m^2	0.047 kg m^2	±250 RPM	10.36 N	1.44 N m	100 ms	300 ms	200 ms

2.1 System Models

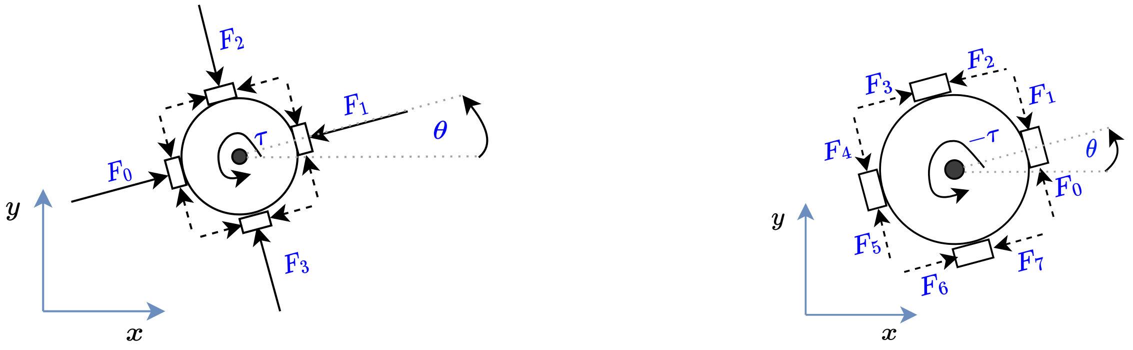
The *decoupled model* is sketched in Figure 1a. Here, the two respective thrusters that accelerate in the same direction are lumped together into one force F_i. Therefore, this model only has four binary inputs. A thruster firing and hence a binary input does not introduce a torque. Torque is only applied by the continuous torque source τ. Linear and angular inputs are hence *decoupled* in this model. Additionally, the limits of the RW are not included in the model. The system dynamics are:

$$\dot{\mathbf{x}} = \underbrace{\begin{bmatrix} \mathbf{0}^{3\times3} & \mathbf{I}^{3\times3} \\ \multicolumn{2}{c}{\mathbf{0}^{3\times6}} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} x \\ y \\ \theta \\ \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} \multicolumn{5}{c}{\mathbf{0}^{3\times5}} \\ 0 & \mathrm{c}\theta\frac{2F_\mathrm{n}}{m} & \mathrm{c}\theta\frac{-2F_\mathrm{n}}{m} & \mathrm{s}\theta\frac{2F_\mathrm{n}}{m} & \mathrm{s}\theta\frac{-2F_\mathrm{n}}{m} \\ 0 & \mathrm{s}\theta\frac{2F_\mathrm{n}}{m} & \mathrm{s}\theta\frac{-2F_\mathrm{n}}{m} & \mathrm{c}\theta\frac{-2F_\mathrm{n}}{m} & \mathrm{c}\theta\frac{2F_\mathrm{n}}{m} \\ \frac{1}{I_S} & \multicolumn{4}{c}{\mathbf{0}^{1\times4}} \end{bmatrix}}_{\mathbf{B}(\theta)} \underbrace{\begin{bmatrix} \tau \\ \mathbf{u}_\mathrm{bin} \end{bmatrix}}_{\mathbf{u}}, \quad \tau \in \mathbb{R}, \mathbf{u}_\mathrm{bin} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \in \{0,1\}^4 \qquad (1)$$

The *coupled model* is sketched in Figure 1b. All thrusters are modeled individually. Therefore, when a thruster is fired, a torque is applied to the system and the angular dynamics are coupled to the binary inputs. In addition, the torque input is modeled as the actuation of the RW: A torque applied to the RW induces a negative torque on the system, which results in an angular acceleration. Thus, a controller using this model can exploit all system's capabilities and takes the RW speed into account. The dynamics are:

$$\dot{\mathbf{x}} = \underbrace{\begin{bmatrix} \mathbf{0}^{3\times3} & \mathbf{I}^{3\times3} & \mathbf{0}^{3\times1} \\ \multicolumn{3}{c}{\mathbf{0}^{4\times7}} \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} x \\ y \\ \theta \\ \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \omega_\mathrm{RW} \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} \multicolumn{9}{c}{\mathbf{0}^{3\times9}} \\ 0 & -\mathrm{s}\theta\frac{F_\mathrm{n}}{m} & \mathrm{s}\theta\frac{F_\mathrm{n}}{m} & -\mathrm{c}\theta\frac{F_\mathrm{n}}{m} & \mathrm{c}\theta\frac{F_\mathrm{n}}{m} & \mathrm{s}\theta\frac{F_\mathrm{n}}{m} & -\mathrm{s}\theta\frac{F_\mathrm{n}}{m} & \mathrm{c}\theta\frac{F_\mathrm{n}}{m} & -\mathrm{c}\theta\frac{F_\mathrm{n}}{m} \\ 0 & \mathrm{c}\theta\frac{F_\mathrm{n}}{m} & -\mathrm{c}\theta\frac{F_\mathrm{n}}{m} & -\mathrm{s}\theta\frac{F_\mathrm{n}}{m} & \mathrm{s}\theta\frac{F_\mathrm{n}}{m} & -\mathrm{c}\theta\frac{F_\mathrm{n}}{m} & \mathrm{c}\theta\frac{F_\mathrm{n}}{m} & \mathrm{s}\theta\frac{F_\mathrm{n}}{m} & -\mathrm{s}\theta\frac{F_\mathrm{n}}{m} \\ \frac{-1}{I_\mathrm{S}} & \frac{F_\mathrm{n}r}{I_\mathrm{S}} & -\frac{F_\mathrm{n}r}{I_\mathrm{S}} & \frac{F_\mathrm{n}r}{I_\mathrm{S}} & -\frac{F_\mathrm{n}r}{I_\mathrm{S}} & \frac{F_\mathrm{n}r}{I_\mathrm{S}} & -\frac{F_\mathrm{n}r}{I_\mathrm{S}} & \frac{F_\mathrm{n}r}{I_\mathrm{S}} & -\frac{F_\mathrm{n}r}{I_\mathrm{S}} \\ \frac{1}{I_\mathrm{RW}} & \multicolumn{8}{c}{\mathbf{0}^{1\times8}} \end{bmatrix}}_{\mathbf{B}(\theta)} \underbrace{\begin{bmatrix} \tau \\ \mathbf{u}_\mathrm{bin} \end{bmatrix}}_{\mathbf{u}} \qquad (2)$$

$$\tau \in \mathbb{R}, \mathbf{u}_\mathrm{bin} = \begin{bmatrix} u_0 & u_1 & u_2 & u_3 & u_4 & u_5 & u_6 & u_7 \end{bmatrix}^T \in \{0,1\}^8$$



(a) Decoupled model of REACSA. The two respective thrusters pointing towards the same direction are modeled as one input F. Torque is applied by τ.

(b) Coupled model of REACSA. Each thruster is modeled as a Force F. A torque τ accelerates the RW and generates an opposing torque in the system.

Fig. 1 The *decoupled* and *coupled* system model of REACSA.

2.2 System Limit Cycle

In both models, linear acceleration can only be achieved through the use of thrusters. If orientation is neglected, both axes behave like an independent double integrator. In the following, without loss of generality, the limit cycle behavior is analyzed for one axis independently. The system is hence considered as a point mass moving on a single axis.

When the thruster is activated for the minimum (maximum) time, the system experiences the lowest (highest) possible thrust impulse. Resulting in a minimum (maximum) change in velocity[1]:

$$\Delta v_{\mathrm{min}} = \frac{2F_{\mathrm{n}} t_{\mathrm{on,min}}}{m} \quad (3a) \qquad \Delta v_{\mathrm{max}} = \frac{2F_{\mathrm{n}} t_{\mathrm{on,\,max}}}{m} \quad (3b)$$

A thruster firing is referred to as a *firing cycle*. Assuming that the thruster has just fired before, waiting for the minimum off time, followed by the minimum firing duration is a *minimum firing cycle*. Within this cycle, the system moves by a certain distance. Assuming the system is at x_0 with the velocity v_k and the goal is to stop the system, there are three cases: case 1: $|v_k| > \Delta v_{\mathrm{max}}$, case 2: $\Delta v_{\mathrm{max}} \geq |v_k| \geq \Delta v_{\mathrm{min}}$ and case 3: $|v_k| < \Delta v_{\mathrm{min}}$. In the first case, the system can not be stopped within one firing cycle and full thrust is applied to reduce the system velocity. In the second case, the system can be fully stopped. For the third case, a minimum thrust firing overcompensates. The system will start moving in the other direction, with again a velocity smaller than Δv_{min}[2]. Hence, the system can not be fully stopped by using the thrusters. Note that case 1 might end up in case 2 or 3 after a few firings. Case 2, in reality, might also degenerate into case 1 and then case 3 due to disturbances and modeling errors. In summary, the system might be fully stopped or held in a limit cycle with a velocity less than Δv_{min} around the target position. Assuming an optimal controller, an upper bound for the limit cycle can be calculated[3]:

$$|\Delta x_{\mathrm{lc}}| \leq \left(\frac{5F_{\mathrm{n}} {t_{\mathrm{min,\,on}}}^2}{4m} + \frac{2F_{\mathrm{n}} t_{\mathrm{off,\,min}} t_{\mathrm{on,min}}}{m} \right) \tag{4}$$

3 MPC Formulation

From the three different binary constraint formulations, this work assembles three different MPCs. Thereby either a linear or quadratic cost function is used, allowing the OCPs with Mixed Integer (MI) constraints and LCCs to be formulated as simpler Linear Program (LP)s. The OCPs are listed in Table 2, and follow the following general structure:

$$\begin{aligned}
J^*(\mathbf{x}_t) = \min_{\mathbf{U}_t, \mathbf{X}_t} \mathcal{L}_f(\mathbf{x}_{t+N|t}) + \sum_{k=0}^{N-1} \mathcal{L}(\mathbf{x}_{t+k|t}, \mathbf{u}_{t+k|t}) & & (5a)\\
\text{s.t. } \mathbf{x}_{t+k+1|t} = \mathbf{x}_{t+k|t} + \Delta t \mathbf{A} \mathbf{x}_{t+k+1|t} + \Delta t \mathbf{B} \mathbf{u}_{t+k|t}, & \quad k \in [[0, N)) & (5b)\\
-\tau_{\mathrm{max}} \leq \mathbf{u}_{0,t+k|t} \leq \tau_{\mathrm{max}}, & \quad k \in [[0, N)) & (5c)\\
\mathbf{x}_{\mathrm{lb}} \leq \mathbf{x}_{t+k|t} \leq \mathbf{x}_{\mathrm{ub}}, & \quad k \in [[0, N)) & (5d)\\
\mathbf{x}_{\mathrm{f,lb}} \leq \mathbf{x}_{t+N|t} \leq \mathbf{x}_{\mathrm{f,ub}} & & (5e)\\
\mathbf{u}_{\mathrm{bin},t+k|t} \in \mathbb{U}_{\mathrm{bin}}, & \quad k \in [[0, N)) & (5f)\\
\mathbf{u}_{\mathrm{bin},t+k|t} \in \mathbb{U}_{\mathrm{time}}, & \quad k \in [[0, N)) & (5g)\\
\mathbf{x}_{t|t} = \mathbf{x}_t & & (5h)
\end{aligned}$$

[1] The system's equations of motion under thruster timing constraints are given in Appendix A.1

[2] The proof for entering the limit cycle is stated in Appendix A.2

[3] The derivation of the limit cycle upper bounds are done is Appendix A.3

The cost function (5a) minimizes the error towards the target state $\hat{\mathbf{x}}$ and the control effort the two cost functions are shown in Section 3.1. The system dynamics are linearized using the first-order Taylor expansion around the current state $\mathbf{x_t}$. This is equivalent to evaluating the respective state-dependent input Matrix at the current orientation, i.e. $\mathbf{B} = \mathbf{B}(\theta_t)$. Using the backward Euler approach with a sampling rate of Δt, they are added as a set of linear equality constraints (5b). The continuous applied torque is limited by a maximum torque constraint (5c) and the system state is limited by a lower- and upper bounding box constraint (5d) to stay below a safety linear and angular velocity, and not to exceed the flat floor. Moreover, in the *coupled model*, this also enforces the minimum and maximum speed of the RW. To ensure recursive feasibility, the final predicted state is constrained by more restrictive terminal bounds (5e). The derivation of these bounds is given in Section 3.4. The different binary constraint formulations, in equation (5f) expressed as a general constraint, are explained in Section 3.2. The linear input timing constraints (5g) are introduced in Section 3.3. To solve the control problem, the first state variable is constrained to the current (measured) system state (5h).

Table 2 The three different MPC formulations that are compared in this work.

	Binary constraints	**Cost function**	**Solver**
MILP	Integer constraint	L1 Norm	SCIP Solver [31]
(non-convex) QP	Penalty term	L2 Norm	SNOPT [32]
MPCC	LCC	L1 Norm, L2 Norm	SNOPT [32], LcqPOW [17]

3.1 Cost Functions

The linear *L1 norm* is expressed as a sum of linear auxiliary constraints:

$$\mathcal{L}(\mathbf{x}_{t+k|t} - \hat{\mathbf{x}}, \mathbf{u}_{t+k|t}) = \mathbb{1}^T \mathbf{e}_{x,k} + \mathbb{1}^T \mathbf{e}_{u,k} \qquad (6a) \qquad \mathcal{L}_f(\mathbf{x}_{t+N|t} - \hat{\mathbf{x}}) = \mathbb{1}^{\mathbb{T}} \mathbf{e}_{x,N} \qquad (6b)$$

Where $\mathbf{e}_{x,t}, \mathbf{e}_{u,t}$ are the auxiliary vectors for state and input cost, with the same dimension as state and input vector. They are related to the state and input by additional linear constraints:

$$-\mathbf{e}_{x,k} \leq \mathbf{Q}(\mathbf{x}_{t+k|t} - \hat{\mathbf{x}}) \leq +\mathbf{e}_{x,k} \qquad (7a) \qquad -\mathbf{e}_{u,k} \leq \mathbf{W}\mathbf{u}_{t+k|t} \leq +\mathbf{e}_{u,k} \qquad (7b)$$

The quadratic *L2 norm* is expressed as quadratic costs:

$$\mathcal{L}(\mathbf{x}_{t+k|t} - \hat{\mathbf{x}}, \mathbf{u}_{t+k|t}) = (\mathbf{x}_{t+k|t} - \hat{\mathbf{x}})^T \mathbf{Q}(\mathbf{x}_{t+k|t} - \hat{\mathbf{x}}) + \mathbf{u}_{t+k|t}{}^T \mathbf{W}\mathbf{u}_{t+k|t} \qquad (8a)$$

$$\mathcal{L}_f(\mathbf{x}_{t+N|t} - \hat{\mathbf{x}}) = (\mathbf{x}_{t+N|t} - \hat{\mathbf{x}})^T \mathbf{Q}(\mathbf{x}_{t+N|t} - \hat{\mathbf{x}}) \qquad (8b)$$

3.2 Binary Input Constraints

In the MIP the respective variables are constrained to *integer decision variables* with value 0 or 1:

$$u_{i,t+k|t} \in \{0, 1\}, \quad \forall u_i \in \mathbf{u_{bin}}, k \in [[0, N)) \qquad (9)$$

For the (non-convex) QP a quadratic *penalty term* (10a) is added to the general cost function (5a):

$$J^*(x_t) = \min_{U_t, X_t} \mathcal{L}_f(\mathbf{x}_{t+N|t}) + \sum_{k=0}^{N-1} \mathcal{L}(\mathbf{x}_{t+k|t}, \mathbf{u}_{t+k|t}) + \sum_{\forall u_i \in \mathbf{u_{bin}}} \beta \ \ (u_{i,t+k|t} - u^2_{i,t+k|t}), \ \beta > 0 \qquad (10a)$$

The weighting factor β relates the penalty term to the rest of the cost function. A bounding box constraint (10b), ensures that (10a) is only optimal if the respective variable is 0 or 1:

$$0 \leq u_{i,t+k|t} \leq 1, \quad \forall u_i \in \mathbf{u_{bin}}, k \in [[0, N)) \tag{10b}$$

In the MPCC the *LCC* is added to the respective variables:

$$0 \leq (1 - u_{i,t+k|t}) \perp u_{i,t+k|t} \geq 0, \quad \forall u_i \in \mathbf{u_{bin}}, k \in [[0, N)) \tag{11}$$

The "$\perp$" means that either the left or the right term must be zero. While the bounds ensure that both are non-negative [33]. An MPCC is difficult to solve, and there are many works and specialized solvers for different specializations of MPCC [17]. Note the reformulation into a general Non-Linear Program (NLP), that can be solved with any NLP solver [16]:

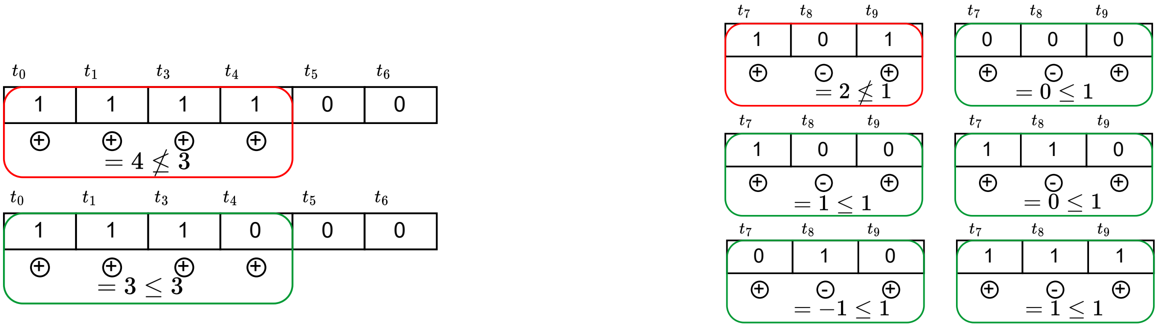
$$(1 - u_{i,t+k|t}) \ u_{i,t+k|t} = 0 \tag{12a}$$

$$0 \leq u_{i,t+k|t} \leq 1 \tag{12b}$$

3.3 Input Timing Constraints

The timing constraints are multiples of the chosen discretization rate $\Delta t = 0.1$s and can be expressed as linear constraints that prevent certain combinatorial binary input patterns. A negative k thereby refers to the input history, before the current time step t.

The *minimum on time* matches the discretization rate, i.e. $t_{\text{on, min}} = 0.1\text{s} = \Delta t$. Thus, assuming a zero-order hold, the minimum on time is enforced naturally.



(a) Maximum on time constraint

(b) Minimum off time constraint

Fig. 2 Examples of input sequences that violate (red) or don't violate (green) the activation time constraints

The *maximum on time* is three times the discretization rate, i.e. $t_{\text{on, max}} = 3 \cdot \Delta t = 0.3$s. A linear equality constraint, sketched in Figure 2a, prevents the respective binary input variables from having the value 1 for four consecutive time steps, by constraining the sum of all four-step-sub-sequences to a maximum value of three:

$$\sum_{j=k}^{k+3} u_{i,t+j|t} \leq 3, \quad \forall k \in [[-3, N-3)), \forall i \in \mathbf{u}_{\text{bin}} \tag{13}$$

The *minimum off time* is twice the discretization rate, i.e. $t_{\text{off, min}} = 2 \cdot \Delta t = 0.2$s. Hence, an input value of 1 must be followed by another 1 or two consecutive 0. The following linear constraint, sketched in Figure 2b, is only violated on the sequence $(1, 0, 1)$ and hence enforces the minimum off time:

$$+u_{i,t+k-1|t} - u_{i,t+k|t} + u_{i,t+k+1|t} \leq 1, \quad \forall k \in [[-2, N-1)), \forall i \in \mathbf{u}_{\text{bin}} \tag{14}$$

3.4 Terminal State Constraint

To ensure feasibility, the system must always be brakeable within the prediction horizon. If e.g. an obstacle appears within the field of view of the MPC, the system can still be stopped. One potential solution is to set the final angular and linear velocity constraint to 0. However, this approach reduces the feasible set, resulting in low system velocity. Furthermore, as demonstrated in Section 2, the system cannot be steered towards zero velocity. Therefore, in this work, a different final constraint is chosen to ensure feasibility.

The *linear velocity terminal constraint* is set to the velocity limits of the minimum limit cycle, as from equation (3a). Hence, it can be ensured that after the final prediction step, the system can be kept within the limit cycle bounds (4). To avoid obstacles (in this work only the edges of the flat floor), the final position constraint is reduced by the limit cycle position bound (equation (4)).

Without loss of generality, assuming the RW's physical speed limit is symmetric $\pm\omega_{\mathrm{RW}}^{\mathrm{max}}$, a *terminal condition for the angular velocity* can be constructed. Enforcing a final RW velocity which is smaller than the physical limit $|\pm\omega_{\mathrm{RW}}^{\mathrm{limit}}| < \omega_{\mathrm{RW}}^{\mathrm{max}}$ leads to a maximum change of velocity before saturating after the final prediction step. This directly leads to a maximum change of the system's angular velocity, which in theory could still be applied after the final prediction step:

$$\Delta\dot{\theta}_{\mathrm{max}} = \left(|\omega_{\mathrm{RW}}^{\mathrm{max}}| - |\omega_{\mathrm{RW}}^{\mathrm{limit}}|\right)\frac{I_{\mathrm{RW}}}{I_{\mathrm{S}}} \tag{15}$$

By constraining the RW's terminal velocity to $\pm\Delta\omega_{\mathrm{RW}}^{\mathrm{limit}}$ and the final system velocity to $\pm\Delta\dot{\theta}_{\mathrm{max}}$, it is guaranteed that the system can always be stabilized to zero angular velocity using only the RW. This ensures that the thrusters can be used for linear feasibility.

4 Binary Constraint Formulation Feasibility Analysis

To compare the three different MPC formulations their feasible regions and the required solver time are compared for different prediction horizons N. Therefore, the OCPs are initialized with $\mathbf{x}_0 = [x_0, 0\,\mathrm{m}, 45°, \dot{x}_0, 0\,\frac{\mathrm{m}}{\mathrm{s}}, 0\,\frac{\mathrm{rad}}{\mathrm{s}}]^T$, with $x_0 \in [-1\,\mathrm{m}, 1\,\mathrm{m}], \dot{x}_0 \in \left[-0.4\,\frac{\mathrm{m}}{\mathrm{s}}, 0.4\,\frac{\mathrm{m}}{\mathrm{s}}\right]$. The target is to reach the origin, while position error and input are minimized, via cost the functions. This analysis is done on the *decoupled model* of REACSA. *SNOPT solver* [32] is used to solve the non-convex QP and MPCC. In addition, the solver *LCQPow* [17] represents a specialized Linear Complementarity Quadratic Program (LCQP) solver. For the MIP problem *SCIPSolver* [34] with python bindings [35] is used. All calculations are done on standard hardware with a 12th generation I7 processor and 32 GB of RAM. For the MPC a control frequency of 10 Hz is desired, which limits the maximum solver time to $\hat{t}_s = 0.1$ s.

The feasible region of the MILP is shown for different prediction horizons in Figure 3. It finds optimal solutions for the whole range of initial x values. Trivially, a higher prediction horizon allows higher initial velocities $\dot{x}$, as it allows for more firing cycles to brake the system into the final limit cycle constraint. Even for the small prediction horizon of $N = 20$, no optimal solution can be found within the desired solver time for initial states close to the origin. With higher prediction horizons, this region becomes bigger. However, the triangles indicate that always at least one feasible, suboptimal solution is found within $\hat{t}_s$.

Figure 4 shows the phase portrait for the system under MILP optimal and suboptimal control law. The latter refers to solutions found within $\hat{t}_s$. While for the shorter prediction horizons, the suboptimal controller only shows marginal difference and still controls the system towards the origin, for $N = 50$ the phase portrait shows regions for which the system drifts away from the origin. Nevertheless, due to the feasibility enforced by the final constraint, the system never gets into an unsafe state e.g., crosses the flat floor edges or breaks other system constraints.

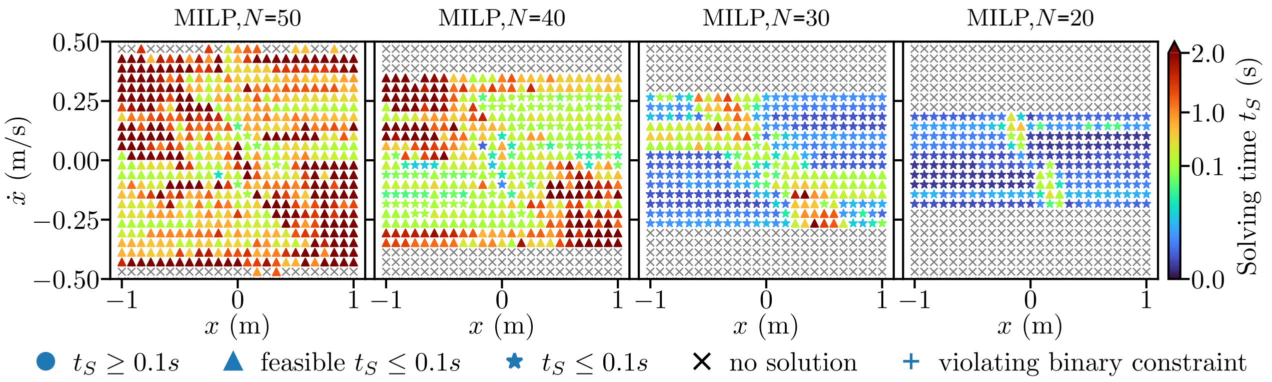


Fig. 3 Feasible region on the x-axis for MILP formulation with different prediction horizons N. The OCP initialized to with the same initial state but different values of x and $\dot{x}$. Colors indicate the solving time for the optimal solution for the respective initial condition. Stars mark optimal solutions in less than 0.1 s, circles in more than 0.1 s. Triangles indicate that there is at least a feasible, suboptimal solution in 0.1 s.

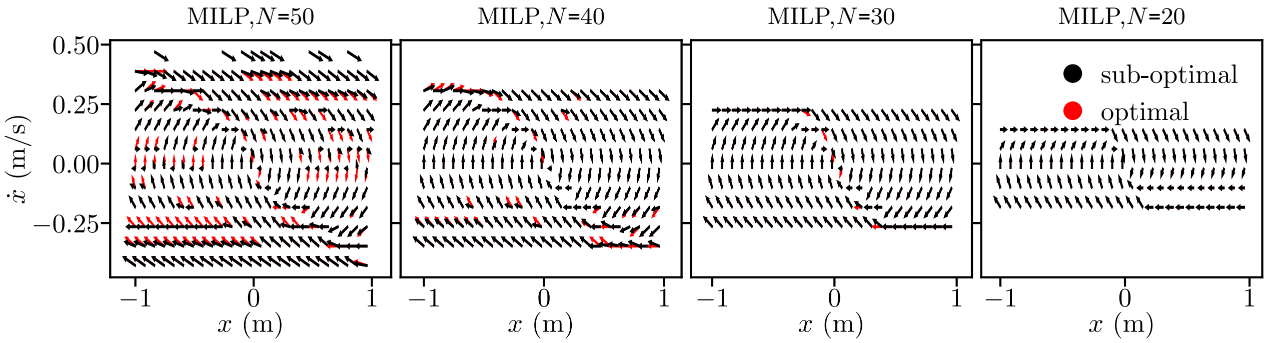


Fig. 4 Phase portrait of the x-axis for MPC with MILP formulation and different prediction horizons N. The red arrows refer to the optimal control law in which the solver might take more than 0.1 s. The black arrows refer to the suboptimal control law where the best solution found before 0.1 s is used as control input.

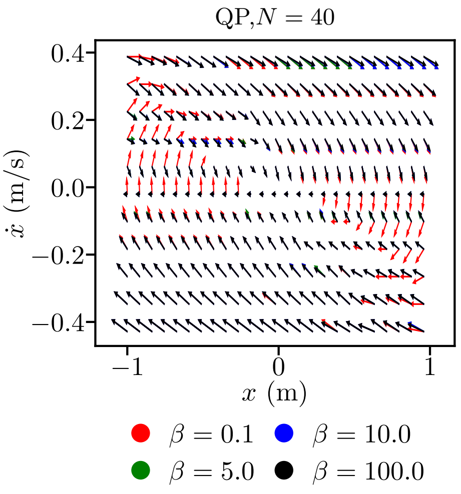


Fig. 5 Phase portrait of the x-axis for MPC with QP formulation and different penalty term weightings β. The different colors refer to the different values of β.

Figure 6 shows the feasible region of the QP for different values of the penalty term weight β. Even for a bigger region than the MILP the optimal solution was found in less than $\hat{t}_s$. Often, even below 0.01 s. However, most of the time the binary enforcing penalty term is not fully minimized, which means that the control inputs take continuous values. The phase portrait of the QP control law (Figure 5) shows, that a higher weighting of the penalty term does not improve this. Instead, a high penalty weight leads the penalty term to dominate the other cost terms. The system is no longer steered toward the target state.

Analyses of the MPCC show that for a few initial conditions, a solution can be found mostly below $0.05s$. While without timing constraints the feasible regions look promising, with timing constraints both tested solvers fail to find solutions except for a very few initial states around the origin. It is worth noting that there are other solvers available for comparison in future work. However, since these solvers often use MIP solving techniques, the results are likely comparable. One other option is to relax the LCC. However, choosing the relax parameter, in general, is not practical and suffers from the same problem as the QP has because it allows non-binary values in the solution.

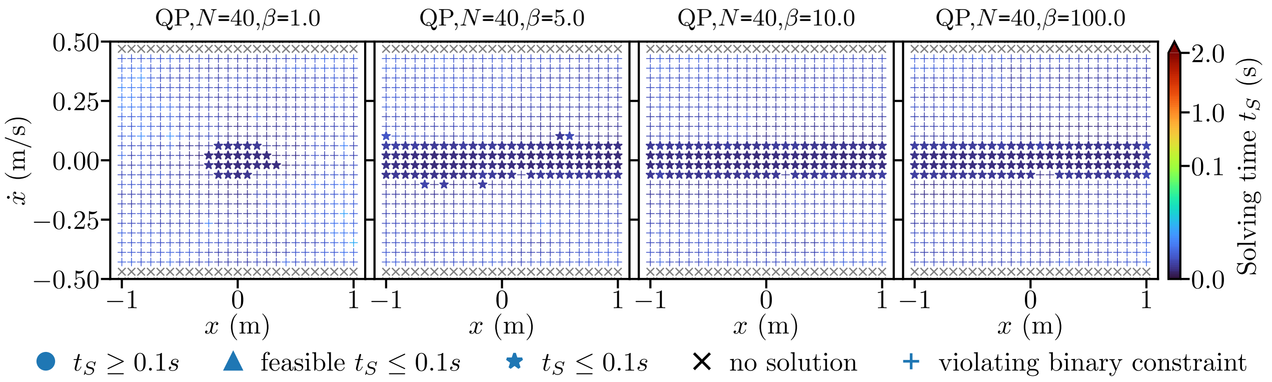


Fig. 6 Feasible region of QP for $N = 40$ with different weighting factors β of the penalty term. The OCP initialized to with the same initial state but different values of x and $\dot{x}$. Colors indicate the solving time for the optimal solution for the respective initial condition. Stars mark optimal solutions in less than 0.1 s. Crosses indicate that an optimal solution is found, but the penalty term is not fully minimized.

Since the goal is optimal control while respecting binary, timing and system constraints, only the MILP formulation turns out to be appropriate. The MPCC formulation provides only partial or no solution, while the QP formulation often yields non-binary inputs. They could be rounded up or down, however, the timing constraints and theoretical assumptions assume binary inputs. In reality, also, these constraints won't be met. Although the MILP formulation does not always achieve optimal solutions in the target time, the feasible solution that is always found in time is optimal enough, at least for small enough prediction horizons, to stabilize the system at the target state. Also, it never breaks any constraints.

5 Simulation Results

By the results of the previous section, the MILP formulation is opted to implement the MPC. If an optimal solution is not found within $\hat{t}_s = 0.1$ s, suboptimal solutions are used, resulting in a suboptimal MIMPC control law. All the results presented in the following are obtained using the *coupled model*. Simulation results for the *decoupled model* can be found in [36]. The final *coupled* controller is implemented as a fast C++ implementation[4] using the *SCIPSolver*'s [34] C-bindings together with *EigenLib* [37]. The simulation is done using *drake-toolbox*'s [38] multibody simulation, which simulates REACSA with thrusters and RW. Since the *coupled model* has twice as many thrusters as the *coupled*, a prediction horizon of $N = 20$ is chosen, which is half of the horizon which still delivered optimal enough results in the previous analysis. The controller runs with a control frequency of 10 Hz together with the simulation on the same hardware as in the previous section.

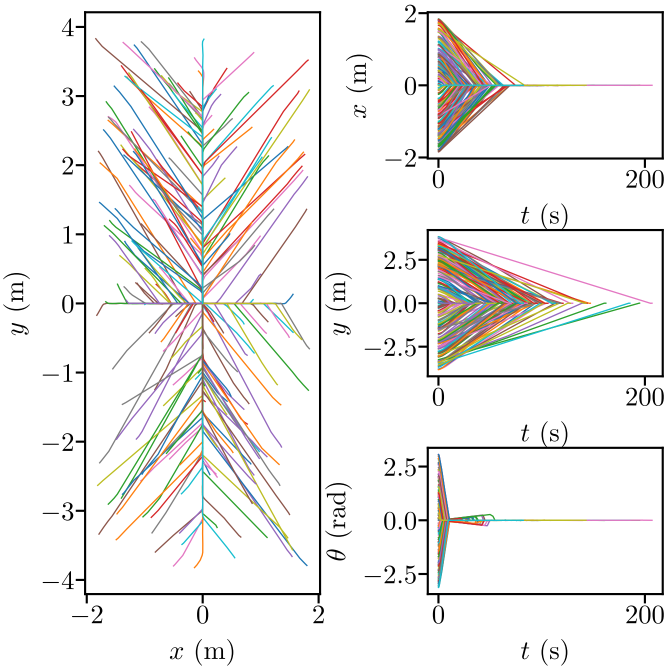
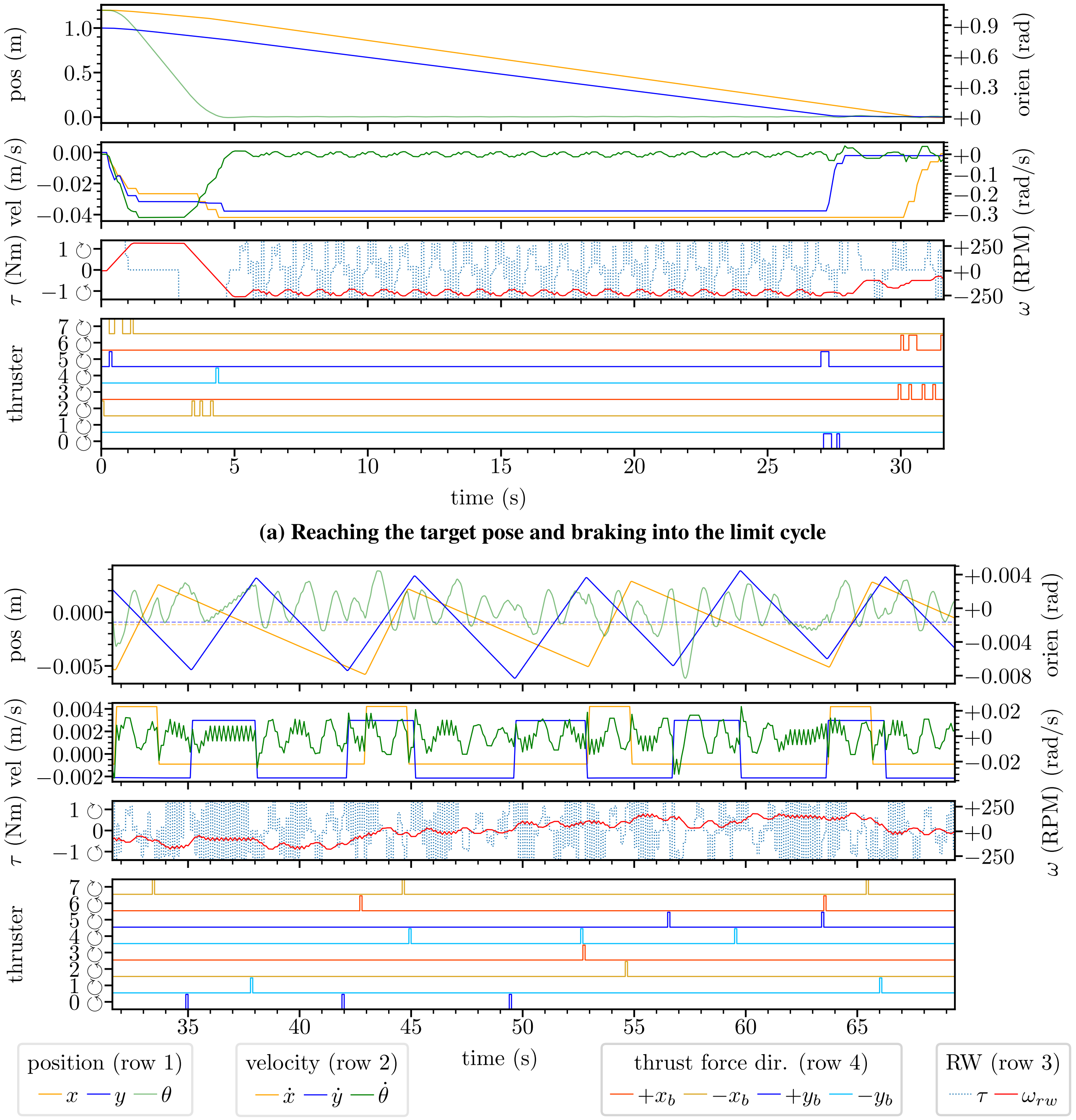


Fig. 7 Simulated system trajectories for 200 tests with random initial poses under the suboptimal MIMPC control law with the *coupled model*.

To evaluate the controller's ability to steer and stabilize the system towards a target state, the system is simulated in 200 experiments, each with a randomly initialized position and orientation. Target is the

[4] The resulting MIMPC library, which can control any (linearized) system with binary and continuous inputs, is open-sourced under `https://github.com/dfki-ric-underactuated-lab/mimpc`. It also contains the simulation.

origin with 0° orientation. The resulting trajectories are shown in Figure 7. Every experiment ends after the system stays for at least 5 s at the origin. For every initial state, the MIMPC is able to stabilize the system at the origin. The average root-mean-square (RMS) position error for the final second of every experiment is 4 mm and the average RMS orientation error 0.1°. The solver found an optimal solution on average after 65 ms. In 30 % of the control cycles, a suboptimal solution was taken as the input.



(a) Reaching the target pose and braking into the limit cycle

(b) Limit cycle around the target pose

Fig. 8 Trajectory of the full system controlled by the MIMPC with *coupled model* of REACSA. In each plot the respective two top rows show position, orientation and velocities over time. The third rows indicate RW's speed and torque. The bottom rows show thruster activations, with the line color denoting the body axis along which the respective impulse is applied. Arrow symbols indicate the direction of torque applied.

One exemplary trajectory, with the system's initial state $\mathbf{x} = [1.2\,\mathrm{m}, 1.0\,\mathrm{m}, 60°, \mathbf{0}^{1\times4}]^T$, is shown in more detail in Figure 8. At the initial phase, the application of multiple firings of thruster 7 applies a negative torque and due to the initial orientation a negative acceleration to the system on both the x and the y axes. This is supported by a slight offset minimum firing of the orthogonal thrusters 2 and 5, which point in the same direction as thruster 7 and thus accelerate in the same direction. The applied torque is canceled out by their orthogonality. In addition, the RW is accelerated to maximum velocity, resulting in negative torque. The controller adheres to the activation time limits, which is evident in the small firing pauses of thrusters 7 during acceleration. After 3 s, the angular speed of the RW is decelerated and so is the system. Thrusters 2 and 4 support the angular deceleration process. After 5 s the system reaches 0° orientation where it keeps oscillating, due to modeling errors, simulation errors and thruster activations. However, the controller solely utilizes the RW for compensation and fine pointing. Note that the speed limits of the RW are never exceeded.

The system reaches the origin first on the y-axis after 27 s. On the x-axis, it reaches the origin after 30 s. The controller uses thruster pairs [5, 0], and [6, 3], pointing along the same axis respectively, to brake the system into the limit cycle on both axes. The zoomed-in part of the figure shows that the system stays in a limit cycle with an average oscillation peak-to-peak amplitude (p-p) of 0.008 m and 0.1° around a RMS error distance of 4 mm and 0.01° to the origin. To do so, the controller activates the thrusters for on average 0.046 s per second. The thruster activation plot shows, that the MIMPC fires the orthogonal thrusters as pairs to keep the system within the limit cycle. In this way, the controller can generate the minimum possible force on both axes without generating a torque. On the other side, at ≈38 s, ≈49 s and ≈60 s, thruster firings do not occur as a pair. Nevertheless, they don't introduce a noticeable large orientation error. Instead, the RW speed shows persistent speed changes with which the controller seems to intercept the torques generated by individual thruster firings.

6 Summary and Conclusion

The analysis of the three different formulations to enforce binary inputs, done in this work, shows that only the MILP formulation works in practice. The non-convex QP finds optimal results, but most of the time the binary constraints are violated. This also has an impact on the timing constraints and theoretical considerations, which assume binary inputs. A higher weighting of the penalty function can not counteract this, while increasing the risk of cost term domination with an unstable system as a result. The MPCC enforces binary values by constraint and in theory provides solvability with any (non-linear) solver. However, the LCC together with the timing constraints are not solvable by a standard solver. Also, a special solver tested in this work failed to find enough feasible solutions. The MILP formulation may not always yield the optimal solution in a reasonable time. Nevertheless, the analysis shows that it can always identify at least one feasible solution, which for short enough prediction horizons is optimal enough to stabilize the system at the target state. Also, by always respecting the constraints, feasibility, and an always safe system state, even with suboptimal control law is guaranteed. Nevertheless, it is a suboptimal solution, which may consume more thrust than necessary.

The simulation of the resulting *coupled* MIMPC, shows that the controller is not only able to drive the system to the target, maintaining the system in a limit cycle at the target, but also exploits the system's capabilities to reduce the thruster usage. The thruster timing constraints, developed in this work, guarantee their safe usage. The RW is driven to its limits but never exceeds them, and the controller does not lose control of the orientation. Hence, it can be concluded that optimal control of a 3-dof free-floating platform with on/off thrusters under consideration of the thruster timings can be achieved via MIMPC. Optimal solutions can be obtained by selecting an appropriate prediction horizon. A safe system state is always ensured even for suboptimal solutions.

After this work successfully developed a novel MIMPC for free-floating platforms, which can natively handle on/off thrusters under timing constraints and control the system in real-time, the next steps are tests on the real hardware to determine if it can handle model errors and disturbances by the not perfectly even flat floor. We also plan a comparison in terms of efficiency and accuracy to the current TVLQR based controller. For future work, this work provides the theoretical and practical base, to add, e.g., trajectory tracking, or obstacle avoidance.

Appendix

A System Limit Cycle

In the following, theoretical considerations of the system's limit cycle around a target position are done. The system's orientation is assumed to be fixed at 0°. Only one axis of movement is considered. Also, in the following, a *thruster firing* refers to a firing of the thruster pair that fires along the same axis as in the *decoupled model*. Thus, it exerts a force of double the magnitude $2F_\mathrm{n}$ but does not produce any torque.

To simplify the analyses, thruster firings are considered as *firing cycles*. It is assumed that before a new cycle starts, the thruster just fired. Hence, a firing cycle k starts with the minimum off time followed by the minimum on time. Between the minimum and maximum on time, the cycle can be stopped, followed by an optional next $(k+1)$ cycle. The notation is slightly different from the rest of the paper: $x_k(t), v_k(t)$ refer to the position and velocity during the k^th firing cycle. Position and velocity just before the cycle are denoted by x_{k-}, v_{k-}, position and velocity just after the cycle by x_{k+}, v_{k+}.

A.1 Constrained Equations of Motion

By integrating Newtons law it follows the velocity $v_k(t)$ during the k^th brake firing cycle:

$$v_k(t) = v_{k-} + \begin{cases} 0, & \text{if } 0 \le t \le t_\mathrm{off,min} \\ -\frac{2F_\mathrm{n}}{m}\left(t - t_\mathrm{off,min}\right), & \text{if } t_\mathrm{off,min} \le t \le t_\mathrm{off,min} + t_\mathrm{on,max} \end{cases} \tag{16}$$

Note that because it is a brake firing, the thruster(s) opposite to the movement direction is fired, reducing the velocity.

By integration, it follows the position during the brake firing cycle:

$$x_k(t) = x_{k-} + \begin{cases} t v_{k-}, & \text{if } 0 \le t \le t_\mathrm{off,min} \\ t v_{k-} - \frac{2F_\mathrm{n}}{2m}\left(t - t_\mathrm{off,min}\right)^2, & \text{if } t_\mathrm{off,min} \le t \le t_\mathrm{off,min} + t_\mathrm{on,max} \end{cases} \tag{17}$$

It follows for a minimum and maximum firing (time) the maximum and minimum change in position and velocity during one firing cycle:

$$\Delta v_\mathrm{min} = \left|v_k(t_\mathrm{on,min}) - v_{k-}\right|, \quad \Delta x_\mathrm{min}(v_{k-}) = \left|x_k(t_\mathrm{on,min}, v_{k-}) - x_{k-}\right| \tag{18}$$

$$\Delta v_\mathrm{max} = \left|v_k(t_\mathrm{on,max}) - v_{k-}\right|, \quad \Delta x_\mathrm{max}(v_{k-}) = \left|x_k(t_\mathrm{on,max}, v_{k-}) - x_{k-}\right| \tag{19}$$

A.2 Limit Cycle Entering

Assuming without loss of generality the systems is at $x_{k-} = 0$ with the velocity $0 \le v_{k-} \le \Delta v_\mathrm{min}$ (Subsection 2.2, limit cycle Case 3). A brake thruster firing with minimum activation time leads to:

$$\Rightarrow -\Delta v_\mathrm{min} < v_{(k+1)-} < 0 \tag{20a}$$

Result (20a) shows that the velocity is overcompensated by a minimum thrust, and therefore the system starts moving with a negative velocity (into the opposite direction) but still less than $|\Delta v_{min}|$. Hence, alternate firings of the opposing thrusters keep it within a limit cycle with velocity bound $\pm\Delta v_{min}$.

A.3 Limit Cycle Stability

A simple control logic is derived which keeps the system in the limit cycle without drifting away from the origin, assuming that an optimal controller knows at least this (possibly suboptimal) control law. Without the loss of generality, assuming the system's current position $x_0 = 0$ and the system is within the limit cycle bound velocity. Also, due to symmetry, only positive velocity is considered, hence $0 < v_{k-} < \Delta v_{\text{min}}$. Now two velocity cases are considered:

1) Case 1: $\frac{1}{2}\Delta v_{\text{min}} \leq v_{k-} < \Delta v_{\text{min}}, \quad x_{k-} = 0$
   Firing directly the opposite thruster will lead to:

$$\begin{aligned} x_k(t_{\text{off,min}} + t_{\text{on,min}}, x_{k-} = 0, v_{k-} = \tfrac{1}{2}\Delta v_{\text{min}}) \leq x_{(k+1)-} &< x_k(t_{\text{off,min}} + t_{\text{on,min}}, x_{k-} = 0, v_{k-} = \Delta v_{\text{min}}) \\ \Rightarrow \qquad \frac{F_{\text{n}} t_{\text{off,min}} t_{\text{on,min}}}{m} \leq x_{(k+1)-} &< \frac{F_{\text{n}} {t_{\text{on,min}}}^2}{m} + \frac{2F_{\text{n}} t_{\text{off,min}} t_{\text{on,min}}}{m} \\ v_k(t_{\text{off,min}} + t_{\text{on,min}}, v_{k-} = \tfrac{1}{2}\Delta v_{\text{min}}) \leq v_{(k+1)-} &< v_k(t_{\text{off,min}} + t_{\text{on,min}}, v_{k-} = \Delta v_{\text{min}}) \\ \Rightarrow \qquad -\frac{1}{2}\Delta v_{\text{min}} \leq v_{(k+1)-} &< 0 \end{aligned}$$

   Considering the current direction of travel as forward, and therefore switching signs, the system has a negative position and is traveling with a positive velocity toward the origin. An optimal controller waits until the system reaches the origin $x_{(k+1)-} = 0$ and moves on with Case 2.

2) Case 2: $0 < v_{k-} < \frac{1}{2}\Delta v_{\text{min}}, \quad x_{k-} = 0$
   The velocity is positive (e.g., $v_{k-} > 0$). Hence, an optimal controller waits till the system is at $\hat{x}_{k-} = \frac{F_{\text{n}}(t_{\text{on,min}})^2}{m}$ and then fires the opposite thruster:

$$\begin{aligned} x_k(t_{\text{off,min}} + t_{\text{on,min}}, x_{k-} = \hat{x}_{k-}, v_{k-} = 0) < x_{(k+1)-} &< x_k(t_{\text{off,min}} + t_{\text{on,min}}, x_{k-} = \hat{x}_{k-}, v_{k-} = \tfrac{1}{2}\Delta v_{\text{min}}) \\ \Rightarrow \qquad 0 < x_{(k+1)-} &< \frac{F_{\text{n}} {t_{\text{on,min}}}^2}{m} + \frac{F_{\text{n}} t_{\text{off,min}} t_{\text{on,min}}}{m} \\ v_k(t_{\text{off,min}} + t_{\text{on,min}}, v_{k-} = 0) < v_{(k+1)-} &< v_k(t_{\text{off,min}} + t_{\text{on,min}}, v_{k-} = \Delta \tfrac{1}{2} v_{\text{min}}) \\ \Rightarrow \qquad -\Delta v_{\text{min}} < v_{(k+1)-} &< -\frac{1}{2}\Delta v_{\text{min}} \end{aligned}$$

   Considering the current direction of travel as forward, and therefore switching signs, the negative position offset is reduced by waiting for $x_{(k+1)-} = 0$. Then continue with Case 1.

In both cases, the maximum deviation from the origin (i.e. $x = 0$) is the limit cycle position bound: The maximum position offset during a minimum firing cycle is the maximum of equation (17).
In Case 1, the firing starts at position $x_k = 0$, and the velocity is bounded by $\frac{1}{2}\Delta v_{\text{min}} \leq v_k < \Delta v_{\text{min}}$:

$$\max_{t \leq t_{\text{off,min}} + t_{\text{on,min}}} [x_k(t, x_{k-} = 0, v_{k-} = \Delta v_{\text{min}})] = \frac{F_{\text{n}} {t_{\text{on,min}}}^2}{m} + \frac{2F_{\text{n}} t_{\text{off,min}} t_{\text{on,min}}}{m} \tag{23}$$

In Case 2, firing starts at position $\hat{x}_{k-} = \frac{F_{\text{n}}(t_{\text{on,min}})^2}{m}$ and the velocity is bounded by $0 < v_k < \frac{1}{2}\Delta v_{\text{min}}$:

$$\max_{t \leq t_{\text{off,min}} + t_{\text{on,min}}} \left[x_k(t, x_{k-} = \hat{x}_{k-}, v_{k-} = \tfrac{1}{2}\Delta v_{\text{min}})\right] = \frac{5F_{\text{n}} {t_{\text{on,min}}}^2}{4m} + \frac{F_{\text{n}}\left(t_{\text{off,min}} t_{\text{on,min}}\right)}{m} \tag{24}$$

Depending on the choice of the timing constraints $t_{\text{off,min}}, t_{\text{on,min}}$, equation (23) or equation (24) gives an upper bound on the maximum position offset. A general conservative bound is given in equation (4). In conclusion, this analysis provides an upper bound of the limit cycle's position and velocity.

Acknowledgments

The first and second authors would like to acknowledge the support of M-RoCk (Grant No.: FKZ 01IW21002) and AAPLE (Grant Number: 50WK2275). The first author would like to acknowledge the support of COOPERANTS (Grant No.: 68GX21003H) and thank the Universität zu Lübeck, especially the Institute for Electrical Engineering in Medicine, for their support not only during this project. Also, he would like to thank the ESA and especially the ORGL team for making this collaboration possible. The second author would also like to acknowledge the support from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 813644. The second author is additionally supported with project funds from the federal state of Bremen for setting up the Underactuated Robotics Lab (201-342-04-01/2023-4-1) Furthermore, many thanks to the *drake* development team for supporting with their toolbox and active support. The same applies to all authors of the *SCIPSolver* and the python bindings, whose non-commercial solver enables this work to be open-sourced.

References

[1] Shubham Vyas, Lasse Maywald, Shivesh Kumar, Marko Jankovic, Andreas Mueller, and Frank Kirchner. Post-capture detumble trajectory stabilization for robotic active debris removal. *Advances in Space Research*, Sept. 2022. ISSN: 0273-1177. DOI: 10.1016/j.asr.2022.09.033.

[2] Shubham Vyas, Marko Jankovic, and Frank Kirchner. Momentum based classification for robotic active debris removal. *Journal of Space Safety Engineering*, 9(4):649–655, Dec. 2022. ISSN: 2468-8967. DOI: 10.1016/j.jsse.2022.09.008.

[3] Francis James, Shubham Vyas, Puneeth Bandikatla, P. Mithun, and S. V. Shah. Design and development of an earth based experimental setup for testing algorithms on space robots. In *Proceedings of the 2015 Conference on Advances In Robotics*, volume 15, pages 1–6, Goa India, July 2015. ACM. ISBN: 978-1-4503-3356-6. DOI: 10.1145/2783449.2783487.

[4] Martin Zwick, Irene Huertas, Levin Gerdes, and Guillermo Ortega. ORGL – ESA's Test Facility for Approach and Contact operations in Orbital and Planetary Environments. In *Proceedings of the International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, volume 6, Madrid, Spain, June 2018.

[5] Willem Suter, Gunter Just, and Marti Vilella. REACSA: Actuated Floating Platform for Orbital Robotic Concept Testing and Control Software Development. In *Proceedings of 17th Symposium on Advanced Space Technologies in Robotics and Automation 2023*, volume 17, Scheltema, Leiden, The Netherlands, Oct. 2023.

[6] Shubham Vyas, Bilal Wehbe, and Shivesh Kumar. Quaternion based LQR for Free-Floating Robots Without Gravity. In *Proceedings of the 2022 CEAS EuroGNC conference*, May 2022.

[7] Anton Bredenbeck, Shubham Vyas, Martin Zwick, Dorit Borrmann, Miguel Olivares-Mendez, and Andreas Nüchter. Trajectory Optimization and Following for a Three Degrees of Freedom Overactuated Floating Platform. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4084–4091, July 2022. arXiv:2207.10693 [cs]. DOI: 10.1109/IROS47612.2022.9981294.

[8] S. Di Cairano, H. Park, and I. Kolmanovsky. Model Predictive Control approach for guidance of spacecraft rendezvous and proximity maneuvering. *International Journal of Robust and Nonlinear Control*, 22(12):1398–1427, 2012. ISSN: 1099-1239. DOI: 10.1002/rnc.2827.

[9] Richard Zappulla, Josep Virgili-Llop, Costantinos Zagaris, Hyeongjun Park, and Marcello Romano. Dynamic Air-Bearing Hardware-in-the-Loop Testbed to Experimentally Evaluate Autonomous Spacecraft Proximity Maneuvers. *Journal of Spacecraft and Rockets*, 54(4):825–839, July 2017. ISSN: 0022-4650. DOI: 10.2514/1.A33769.

[10] Avijit Banerjee, Sumeet G. Satpute, Christoforos Kanellakis, Ilias Tevetzidis, Jakub Haluska, Per Bodin, and George Nikolakopoulos. On the Design, Modeling and Experimental Verification of a Floating Satellite Platform. *IEEE Robotics and Automation Letters*, 7(2):1364–1371, Apr. 2022. ISSN: 2377-3766. DOI: 10.1109/LRA.2021.3140134.

[11] Asumi Nishimura and Katsuyoshi Tsujita. An experimental maneuver control for rendezvous and autonomous docking of a small spacecraft. *SICE Journal of Control, Measurement, and System Integration*, 0(0):1–11, 2023. ISSN: 1882-4889. DOI: 10.1080/18824889.2023.2187163.

[12] Rafael Vazquez, Francisco Gavilan, and Eduardo F. Camacho. Model Predictive Control for Spacecraft Rendezvous in Elliptical Orbits with On/Off Thrusters. 48(9):251–256, Jan. 2015. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2015.08.092.

[13] Gilberto Arantes, Luiz S. Martins-Filho, and Adrielle C. Santana. Optimal On-Off Attitude Control for the Brazilian Multimission Platform Satellite. *Mathematical Problems in Engineering*, 2009:e750945, Nov. 2009. ISSN: 1024-123X. DOI: 10.1155/2009/750945.

[14] Moad Abudia, Michael Harlan, Ryan Self, and Rushikesh Kamalapurkar. Switched Optimal Control and Dwell Time Constraints: A Preliminary Study. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 3261–3266, Jeju, Korea (South), Dec. 2020. IEEE. ISBN: 978-1-72817-447-1. DOI: 10.1109/CDC42340.2020.9304087.

[15] Laura Di Giacomo, Giacomo Patrizi, and Emanuele Argento. Linear Complementarity as a General Solution Method to Combinatorial Problems. *INFORMS Journal on Computing*, 19(1):73–79, Feb. 2007. ISSN: 1091-9856. DOI: 10.1287/ijoc.1050.0146.

[16] Roger Fletcher and Sven Leyffer. Solving mathematical programs with complementarity constraints as nonlinear programs. *Optimization Methods and Software*, 19(1):15–40, Feb. 2004. ISSN: 1055-6788. DOI: 10.1080/10556780410001654241.

[17] Jonas Hall, Armin Nurkanović, Florian Messerer, and Moritz Diehl. A Sequential Convex Programming Approach to Solving Quadratic Programs and Optimal Control Problems With Linear Complementarity Constraints. *IEEE Control Systems Letters*, 6:536–541, 2022. ISSN: 2475-1456. DOI: 10.1109/LCSYS.2021.3083467.

[18] B. Curtis Eaves. The Linear Complementarity Problem. *Management Science*, 17(9):612–634, May 1971. DOI: 10.1287/mnsc.17.9.612.

[19] Michael Posa, Cecilia Cantu, and Russ Tedrake. A direct method for trajectory optimization of rigid bodies through contact. *The International Journal of Robotics Research*, 33(1):69–81, Jan. 2014. ISSN: 0278-3649. DOI: 10.1177/0278364913506757.

[20] Jan Leeuwen. *Algorithms and Complexity*, volume 1 of *Handbook of Theoretical Computer Science*. Elsevier, Sept. 1990. ISBN: 978-0-444-88071-0.

[21] David B. Doman, Brian J. Gamble, and Anhtuan D. Ngo. Quantized Control Allocation of Reaction Control Jets and Aerodynamic Control Surfaces. *Journal of Guidance, Control, and Dynamics*, 32(1), Jan. 2009. DOI: 10.2514/1.37312.

[22] Márcio Santos Vieira, Roberto Kawakami Harrop Galvão, and Karl Heinz Kienitz. Attitude stabilization with actuators subject to switching-time constraints using explicit MPC. In *2011 Aerospace Conference*, pages 1–8, Mar. 2011. DOI: 10.1109/AERO.2011.5747482.

[23] Mirko Leomanni, Andrea Garulli, Antonio Giannitrapani, and Fabrizio Scortecci. All-Electric Spacecraft Precision Pointing Using Model Predictive Control. *Journal of Guidance, Control, and Dynamics*, 38(1):161–168, 2015. ISSN: 0731-5090. DOI: 10.2514/1.G000347.

[24] Pantelis Sopasakis, Daniele Bernardini, Hans Strauch, Samir Bennani, and Alberto Bemporad. A Hybrid Model Predictive Control Approach to Attitude Control with Minimum-Impulse-Bit Thrusters. pages 2079–2084, July 2015. DOI: 10.1109/ECC.2015.7330846.

[25] Imane Khayour, Sylvain Durand, Loïc Cuvillon, and Jacques Gangloff. Active Damping of Parallel Robots Driven by Elastic Cables using On-Off Actuators through Model Predictive Control Allocation. *IFAC-PapersOnLine*, 53(2):9169–9174, Jan. 2020. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2020.12.2167.

[26] A. Richards and J. How. Mixed-integer programming for control. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 2676–2683 vol. 4, June 2005. ISSN: 2378-5861. DOI: 10.1109/ACC.2005.1470372.

[27] Lijun Zong, Jianjun Luo, Mingming Wang, and Jianping Yuan. Obstacle avoidance handling and mixed integer predictive control for space robots. *Advances in Space Research*, 61(8):1997–2009, Apr. 2018. ISSN: 02731177. DOI: 10.1016/j.asr.2018.01.025.

[28] Matheus Henrique Marcolino, Roberto Kawakami Harrop Galvão, and Karl Heinz Kienitz. Predictive Control of Linear Systems with Switched Actuators Subject to Dwell-Time Constraints. *Journal of Control, Automation and Electrical Systems*, 32(1):1–17, Feb. 2021. DOI: 10.1007/s40313-020-00667-9.

[29] Jerry Mendel. Performance cost functions for a reaction-jet-controlled system during an on-off limit cycle. *IEEE Transactions on Automatic Control*, 13(4):362–368, Aug. 1968. ISSN: 1558-2523. Conference Name: IEEE Transactions on Automatic Control. DOI: 10.1109/TAC.1968.1098940.

[30] Anton Bredenbeck. SatSim-REACSA - A Floating Platform for Simulation of Micro-Gravity With Three Degrees of Freedom, Finding and Following Optimal Trajectories With an Overactuated System. Master's thesis, Julius-Maximulians-Universität Würzburg, Würzburg, Feb. 2022.

[31] Tobias Achterberg. SCIP: solving constraint integer programs. *Mathematical Programming Computation*, 1(1):1–41, July 2009. ISSN: 1867-2957. DOI: 10.1007/s12532-008-0001-1.

[32] Philip E. Gill, Walter Murray, and Michael A. Saunders. SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization. *SIAM Review*, 47(1):99–131, 2005. DOI: 10.1137/S0036144504446096.

[33] Richard W. Cottle, Jong-Shi Pang, and Richard E. Stone. *The Linear Complementarity Problem*. Classics in Applied Mathematics. SIAM, 2 edition, 2009. ISBN: 978-0-89871-686-3. DOI: 10.1137/1.9780898719000.

[34] Ksenia Bestuzheva, Mathieu Besançon, Wei-Kun Chen, Antonia Chmiela, Tim Donkiewicz, Jasper van Doornmalen, Leon Eifler, Oliver Gaul, Gerald Gamrath, Ambros Gleixner, Leona Gottwald, Christoph Graczyk, Katrin Halbig, Alexander Hoen, Christopher Hojny, Rolf van der Hulst, Thorsten Koch, Marco Lübbecke, Stephen J. Maher, Frederic Matter, Erik Mühmer, Benjamin Müller, Marc E. Pfetsch, Daniel Rehfeldt, Steffan Schlein, Franziska Schlösser, Felipe Serrano, Yuji Shinano, Boro Sofranac, Mark Turner, Stefan Vigerske, Fabian Wegscheider, Philipp Wellner, Dieter Weninger, and Jakob Witzig. The SCIP Optimization Suite 8.0. ZIB-Report 21-41, Zuse Institute Berlin, Dec. 2021. http://nbn-resolving.de/urn:nbn:de:0297-zib-85309.

[35] Stephen J. Maher, Matthias Miltenberger, João Pedro Pedroso, Daniel Rehfeldt, Robert Schwarz, and Felipe Serrano. PySCIPOpt: Mathematical Programming in Python with the SCIP Optimization Suite. volume 9725, pages 301–307. Springer, 2016. DOI: 10.1007/978-3-319-42432-3_37.

[36] Franek Stark, Shubham Vyas, Georg Schildbach, and Frank Kirchner. Linear Model Predictive Control for a planar free-floating platform: A comparison of binary input constraint formulations. In *Proccedings of 17th Symposium on Advanced Space Technologies in Robotics and Automation*, volume 17, Oct. 2023. DOI: 10.48550/arXiv.2312.10788.

[37] Gael Guennebaud, Benoit Jacob, et al. Eigen v3, 2010. http://eigen.tuxfamily.org.

[38] Russ Tedrake and Drake-Development-Team. Drake: Model-based design and verification for robotics, 2019. https://drake.mit.edu.