LINEAR MODEL PREDICTIVE CONTROL FOR A PLANAR FREE-FLOATING PLATFORM: A COMPARISON OF BINARY INPUT CONSTRAINT FORMULATIONS

Franek Stark[1,2], **Shubham Vyas**[2,4], **Georg Schildbach**[3], **and Frank Kirchner**[2,4]

[1]*Robotics and Autonomous Systems, Universität zu Lübeck, Germany, franek.stark@student.uni-luebeck.de*
[2]*Robotics Innovation Centre (RIC), DFKI GmbH, Germany, {franek.stark, shubham.vyas, frank.kirchner}@dfki.de*
[3]*Institute for Electrical Engineering in Medicine, Universität zu Lübeck, Germany, georg.schildbach@uni-luebeck.de*
[4]*AG Robotik, University of Bremen, Germany*

ABSTRACT

This work develops a first Model Predictive Control for European Space Agency's 3-dof free-floating platform. The challenges of the platform are the on/off thrusters, which cannot be actuated continuously and which are subject to certain timing constraints. This work compares penalty-term, Linear Complementarity Constraints, and classical Mixed Integer formulations in order to develop a controller that natively handles binary inputs. Furthermore, linear constraints are proposed which enforce the timing constraints. Only the Mixed Integer formulation turns out to work sufficiently. Hence, this work develops a new Mixed Integer Model Predictive Control on the decoupled model of the platform. Feasibility analysis and simulation results show that for a short enough prediction horizon, this controller can (sub)optimally stabilize and control the system under consideration of the constraints in real-time.

1. INTRODUCTION

Satellite control to do rendezvous maneuvers, e.g., in the field of space debris removal, recently gained attention [20]. To test such controllers and subsystems, the European Space Agency (ESA)'s Orbital Robotics and GNC Lab (ORGL) features a 9×5 m flat floor [21]. On top, the 220 kg heavy air-bearing platform REACSA simulates a satellite [4]. REACSA is equipped with a cold gas propulsion system and a reaction wheel. The latter exerts a precise desired torque to the system. The eight thrusters are mounted in a way to be able to control linear and angular acceleration. Like a real satellite or system that uses rocket motors or reaction wheels, this experimental platform is subject to certain input constraints. In REACSA's case, these constraints are rather strict: Besides the reaction wheel's minimum and maximum angular velocity, the thrusters can only be on/off actuated. Thus, a thruster can either deliver its full thrust or no thrust. In addition, the thrusters are subject to certain time restrictions: Once a thruster has been switched on, it must remain switched on for a minimum time $t_{\text{on,min}}$ to exert repeatable and reliable thrust. Also, it must not exceed a maximum activation time $t_{\text{on,max}}$ due to the requirements imposed by the pneumatic system. After an activation phase, a thruster must have an off period $t_{\text{off,min}}$, while the buffer is filled up.

The current controller 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 [4]. By enabling optimal control while respecting any constraint, developing an Model Predictive Control (MPC) is hence the next step.

Existing works on MPC for satellites with binary thrusters can be divided into different classes: First, continuous MPC formulations treat the control input as continuous and convert it into binary values, e.g., by using a Delta-Sigma modulator [19]. Or, like the MPC presented by Arantes et al. [2], output continuous PWM parameters, which assumes one firing cycle per prediction step and cannot flexibly allocate binary thruster values.

Since optimal control by exploiting perfect thruster allocation under consideration of the timing constraints is desired, this work focuses on the second class: formulations that have direct binary outputs.

Enforcing an input to be a binary variable makes the controller's underlying optimization problem significantly complex, since a non-convexity is introduced. The current state-of-the-art literature tackles this 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. The binary variables are assumed to be continuous, and the problem is a QP. To prioritize

binary values, an additional quadratic penalty term is used [1].

3. Good results have been achieved with so called Mathematical Program with Complementarity Constraints (MPCC) in the area of contact-implicit trajectory optimization [15]. Here, a quadratic non-convex constraint, called LCC, enforces binary values.

While for the MIP an appropriate solver is necessary, the QP can be solved by any (non-convex) QP solver. A MPCC can be solved with any non-linear solver. However, problems with binary variables are NP-hard, and solvability in real-time depends strongly on the problem [11].

For the MILP approach, there are already works in the space and satellite context, but these are mostly concerned with attitude control and do not have such strict timing constraints on their inputs [18, 6]. And even of these, only a very few have implemented a real-time capable controller, like [12, 16]. An MPC for a very similar platform without timing constraints is developed in Khayour et al. [10]. Compared to classical controllers, it performs better, but is not real-time capable. For the LCP and quadratic-cost-based formulation, to the best of the author's knowledge, there are no results yet for a similar application.

The goal of this work is to find a MPC formulation that knows about both the binary and the timing constraints and can solve the optimization problem within a reasonable time. Hence, the three different binary constraint formulations are compared on a simplified model of REACSA. An MPC for this is developed and tested in simulation.

This paper is organized as follows: Section 2 describes the system model used and gives some theoretical considerations about the system's limit cycle. Section 3 introduces the MPC formulations. The results are described in section 4 followed by the summary and conclusion in section 5.

2. SYSTEM DESCRIPTION

In this section, the used model of REACSA is presented and analyzed for its limit cycle behavior.

2.1. Dynamic model

Fig. 1 shows the model of REACSA used in this work. The two respective thrusters that accelerate in the same direction are combined into one force F_i. Therefore, the binary number of binary inputs is reduced to four, and binary inputs don't introduce a torque. Orientation changes are applied by the torque source τ. Hence, this model is referred to as the decoupled model. Note that in this work the reaction wheel with its limits is not modeled.

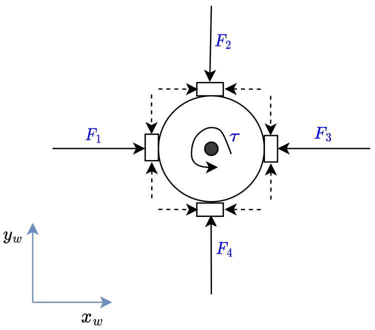

Figure 1. Schematics of the modeled system. Pairs of thruster forces (dashed) that apply thrust in the same direction are lumped together to one single (virtual) thruster which applies force F_i. Rotation is applied by the torque source τ.

The state vector $\mathbf{x}$ contains the linear and angular position and velocity in world coordinates:

$$\mathbf{x} = \begin{bmatrix} x & y & \theta & \dot{x} & \dot{y} & \dot{\theta} \end{bmatrix}^T \tag{1}$$

The system input vector $\mathbf{u}$ contains the applied torque τ and the four binary inputs $u_{1,\dots,4} \in \{0, 1\}$:

$$\mathbf{u} = \begin{bmatrix} \tau & u_1 & u_2 & u_3 & u_4 \end{bmatrix}^T \tag{2}$$

The system dynamics are given by:

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}) = \underbrace{\begin{bmatrix} \mathbf{0}^{3\times3} & \mathbf{I}^{3\times3} \\ \multicolumn{2}{c}{\mathbf{0}^{3\times6}} \end{bmatrix}}_{\mathbf{A}} \mathbf{x}$$

$$+ \underbrace{\begin{bmatrix} \multicolumn{5}{c}{\mathbf{0}^{3\times5}} \\ 0 & c\theta\frac{2F_\text{n}}{m} & c\theta\frac{-2F_\text{n}}{m} & s\theta\frac{2F_\text{n}}{m} & s\theta\frac{-2F_\text{n}}{m} \\ 0 & s\theta\frac{2F_\text{n}}{m} & s\theta\frac{-2F_\text{n}}{m} & c\theta\frac{-2F_\text{n}}{m} & c\theta\frac{2F_\text{n}}{m} \\ \frac{1}{I_{zz}} & \multicolumn{4}{c}{\mathbf{0}^{1\times4}} \end{bmatrix}}_{\mathbf{B}(\theta)} \mathbf{u} \tag{3}$$

Where $s\theta, c\theta$ denote the sinus and cosine of the system's orientation, F_n denotes the force applied by a thruster firing, m and I_{zz} denote the system's mass and inertia. Since two respective thrusters are actuated together, a firing creates $2F_\text{n}$ force. Note that the input matrix $\mathbf{B}(\theta)$ depends on the orientation of the system, so it is a linear time-variant system.

2.2. Systems limit cycle

The minimum activation time for a thruster causes the system to experience a minimum thrust force or acceleration. Consequently, this leads to a minimum change in velocity on the system:

$$\Delta v_{\min} = \frac{2F_{\mathrm{n}} t_{\mathrm{on,min}}}{m} \tag{4}$$

During this minimum firing, the system will travel a certain distance:

$$\Delta x_{\min} = +\frac{2F_{\mathrm{n}} \left(t_{\mathrm{on,min}}\right)^2}{2m} + t_{\mathrm{off,min}} v_0 + t_{\mathrm{on,min}} v_0 \tag{5}$$

Where v_0 denotes the system's velocity before the firing, measured on the thruster axis and the direction of thrust force. Note the assumption that the thruster had just fired before, and hence waits $t_{\mathrm{off,min}}$. If the system has a speed lower than $\Delta v_{\min}$ on the axis in which respective thruster points, a minimum thrust firing overcompensates. The system will start moving in the other direction, with a velocity that is again smaller than $\Delta v_{\min}$. Hence, the system can not be fully stopped [14].

Instead of fully stopping the system, it can only be held in an artificial limit cycle around the target position. Assuming that an optimal controller fires at the right moment to keep the system as close as possible, an upper one-dimensional position bound for the limit cycle can be derived:

$$\pm\left(\frac{2F_{\mathrm{n}} t_{\mathrm{off,\ min}} t_{\mathrm{on,min}}}{m} + \frac{0.625 \cdot 2F_{\mathrm{n}} \left(t_{\mathrm{on,min}}\right)^2}{m}\right) \tag{6}$$

3. MPC FORMULATION

The general MPC optimization problem is stated in the following:

$$\min_{\mathbf{x}_{t+k|t}, \mathbf{x}_{t+N|t}, \mathbf{u}_{t+k|t}} \mathcal{L}_f(\mathbf{x}_{t+N|t}) + \sum_{k=0}^{N-1} \mathcal{L}(\mathbf{x}_{t+k|t}, \mathbf{u}_{t+k|t}) \tag{7a}$$

$$\begin{aligned}
\forall k \in [0, N) \text{ s.t. } \mathbf{x}_{t+k+1|t} &= f(\mathbf{x}_{t+k|t}, \mathbf{u}_{t+k|t}), && \text{(7b)}\\
\mathbf{u}_{t+k|t} &\in \mathbb{U} && \text{(7c)}\\
\mathbf{x}_{t+k|t} &\in \mathbb{X} && \text{(7d)}\\
\mathbf{x}_{t+N|t} &\in \mathbb{X}_f && \text{(7e)}\\
\mathbf{x}_{t|t} &= \mathbf{x}_t && \text{(7f)}
\end{aligned}$$

Note, that $\mathbf{x}_{t+k|t}$ and $\mathbf{u}_{t+k|t}$ refers to the k steps into the future predicted state or input at time step t. The above problem consists of the cost function (7a) to minimize. The system prediction is based on the system dynamics constraint (7b). The inputs $\mathbf{u}_{t+k|t}$ have to be in the input set $\mathbb{U}$ (7c), the intermediate predicted states within the state set $\mathbb{X}$ (7d) and the final predicted state within the final state set $\mathbb{X}_f$ (7e). In the following, the parts of the optimization problem are explained in more detail. The three MPC formulations compared in this work are listed within Table 1 and consist of different parts.

	Binary constraints	**Cost function**
MILP	Integer constraint	L1 Norm
QP	Penalty term	L2 Norm
MPCC	LCC	L2 Norm

Table 1. The three different MPC foundations that are compared in this work together with their optimization problem ingredients.

3.1. System dynamics

The system dynamics (3) are linearized using the first-order Taylor expansion around the current state $\mathbf{x_t}$. This is equivalent to evaluating the state-dependent input Matrix $\mathbf{B}(\theta)$ at the current orientation θ_t. The system is discretized using the backward Euler approach, with a sampling rate of Δt. Hence, the general dynamic constraint (7b) in the MPC formulation expands to a set of linear constraints:

$$\mathbf{x}_{t+k+1|t} = \mathbf{x}_{t+k|t} + \Delta t \mathbf{A} \mathbf{x}_{t+k+1|t} + \Delta t \mathbf{B} \mathbf{u}_{t+k|t}$$

Where $\mathbf{A}$ and $\mathbf{B} = \mathbf{B}(\theta_t)$ refer to the state and input matrix of the system dynamics (3).

3.2. Thruster timing constraints

Minimum on time The chosen system discretization rate $\Delta t = 0.1$s matches the minimum on time $t_{\mathrm{on,\ min}} = 0.1$s. Hence, by assuming a zero-order hold, the minimum on-time is enforced naturally.

Maximum on time The maximum on time is a multiple of the discretization rate: $t_{\mathrm{on,\ max}} = 3 \cdot \Delta t = 0.3$s. Therefore, a constraint must prevent the binary input variables from having the value 1 for four consecutive time steps, respectively. This is done via a sliding window constraint, which for each input limits the sum of all consecutive subsequences of length four to three and is stated as:

$$\sum_{j=k}^{k+3} u_{i,t+j|t} \leq 3, \forall k \in [-3, N-3), \forall i \in [1, 3] \tag{8}$$

The sliding summation window constraint is sketched for a sequence of 6 binary inputs in Fig. 2.

Minimum off time The minimum off time is twice the discretization rate, i.e. $t_{\mathrm{off,\ min}} = 2 \cdot \Delta t = 0.2$s. Hence, an input value of 1 must be followed by another 1 or two consecutive 0. Hence, the input sequence $(1, 0, 1)$ is prevented by another sliding window constraint (Fig. 3), which is given by:

$$u_{i,t+k-1|t} - u_{i,t+k|t} + u_{i,t+k+1|t} \leq 1,\ \forall k \in [-2, N-1), \forall i \in [1, 3] \tag{9}$$

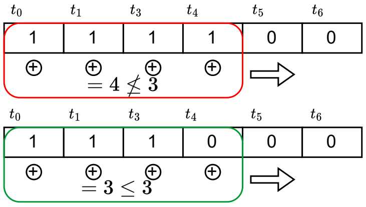

Figure 2. Example for an input sequence that (top) violates the maximum on time constraint and (bottom) does not.

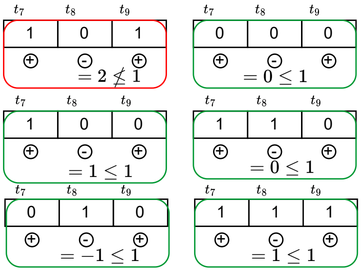

Figure 3. Example for the input sequence (top left) that violates the minimum off-time constraint and sequences that don't.

3.3. Continuous input constraints

The applied torque is limited by a maximum torque:

$$-\tau_{\max} \leq \mathbf{u}_{0,t+k|t} \leq \tau_{\max} \tag{10}$$

3.4. Binary input constraints

The three formulations to enforce binary values for the thruster inputs compared in this work are stated in the following. Together with the bounding box constraint on the continuous input (10), they replace the input constraint (7c) in the MPC formulation.

Linear Complementarity Constraints The condition for the input variables only taking value 0 or 1 can be expressed as a set of LCC's:

$$0 \leq (1 - u_{i,t+k|t}) \perp u_{i,t+k|t} \geq 0, \;\forall i \in [1,3] \tag{11}$$

These constraints emerge from Linear Complementarity Programs and the "$\perp$" means that either the left or the right term must be zero, while both have to be non-negative [5]. A Mathematical Program with Complementarity Constraints is difficult to solve, and there exist many techniques and specialized solvers for MPCC specializations [9]. Note that (11) can be reformulated into a general Non Linear Program (NLP):

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

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

This formulation can be solved with any appropriate NLP solver [7]. With the LCC, the optimization problem becomes a Linear Complementarity Quadratic Program (LCQP).

Penalty term In this formulation, the binary variables are not enforced via a hard constraint. Instead, a quadratic penalizing term [1] together with a bounding box constraint is added to the cost function:

$$J^*(x_t) = \min_{U_t, X_t} \mathcal{L}_f(\mathbf{x}_{t+N|t}) + \sum_{k=0}^{N-1} \mathcal{L}(\mathbf{x}_{t+k|t}, \mathbf{u}_{t+k|t})$$

$$+ \sum_{j=1}^{3} 4\beta\; (u_{i,t+k|t} - u_{i,t+k|t}^2),\; \beta > 0 \tag{13a}$$

$$0 \leq u_{i,t+i|t} \leq 1, \forall i \in [1,3] \tag{13b}$$

The penalty term is only optimal if the respective variable is 0 or 1. Note that the penalizing term is non-convex. Hence, the optimization problem becomes a non-convex QP and can be solved by any appropriate QP-solver.

Integer constraint The binary decision variables are formulated as integer variables which can only take the value 0 or 1:

$$u_{i,t+k|t} \in \{0,1\},\; \forall i \in [1,3] \tag{14}$$

This approach transforms the problem into a Mixed Integer Program. As mentioned above, solving these problems requires a special class of solver.

3.5. State constraints

The system state is limited by a lower- and upper bound to stay below a safety velocity, and not to exceed the edges of the flat floor. Thus (7d) becomes:

$$\mathbf{x}_{\text{lb}} \leq \mathbf{x}_{t+k|t} \leq \mathbf{x}_{\text{ub}} \tag{15}$$

3.6. Final constraint

To ensure recursive feasibility, the terminal constraint (7e) is set to the velocity limits of the minimum limit cycle (4). Hence, it can be ensured that after the final prediction step, the system can be kept within the limit cycle bounds (5).

3.7. Cost function

The cost function (7a) is used in two ways. First, to minimize the deviation from the target pose $\hat{\mathbf{x}} \in \mathbb{R}^6$.

Secondly, to minimize the control effort, in this case mainly thrust. In this work, due to the different binary constraint formulations and solvers used, two different cost terms are studied:

L1 Norm The L1 norm can be expressed as a linear sum of additional linear auxiliary constraints:

$$\mathcal{L}(\mathbf{x}_{t+k|t} - \hat{\mathbf{x}}, \mathbf{u}_{t+k|t}) = \mathbb{1}^T \mathbf{e}_{x,k} + \mathbb{1}^T \mathbf{e}_{u,k} \tag{16a}$$

$$\mathcal{L}_f(\mathbf{x}_{t+N|t} - \hat{\mathbf{x}}) = \mathbb{1}^{\mathbb{T}} \mathbf{e}_{x,N} \tag{16b}$$

Where $\mathbf{e}_{x,t} \in \mathbb{R}^6, \mathbf{e}_{u,t} \in \mathbb{R}^5$ are the auxiliary vectors for state and input cost. They are related to the state and input by additional linear constraints:

$$-\mathbf{e}_{x,k} \leq \mathbf{Q}(\mathbf{x}_{t+k|t} - \hat{\mathbf{x}}) \leq +\mathbf{e}_{x,k} \tag{17a}$$

$$-\mathbf{e}_{u,k} \leq \mathbf{W}(\mathbf{u}_{t+k|t}) \leq +\mathbf{e}_{u,k} \tag{17b}$$

Where $\mathbf{Q} \in \mathbb{R}^{6\times}, \mathbf{W} \in \mathbb{R}^{5\times}$, are the cost matrices that define the weighting for each state and input. The auxiliary vectors are added as another decision variable to the optimization problem. Hence, the optimization problem stays a linear program.

L2 Norm The L2 norm is expressed as a quadratic cost:

$$\mathcal{L}(\mathbf{x}_{t+k|t} - \hat{\mathbf{x}}, \mathbf{u}_{t+k|t}) = (\mathbf{x}_{t+k|t} - \hat{\mathbf{x}})^T \mathbf{Q} (\mathbf{x}_{t+k|t} - \hat{\mathbf{x}}) + (\mathbf{u}_{t+k|t})^T \mathbf{W} (\mathbf{u}_{t+k|t}) \tag{18}$$

$$\mathcal{L}_f(\mathbf{x}_{t+N|t} - \hat{\mathbf{x}}) = (\mathbf{x}_{t+N|t} - \hat{\mathbf{x}})^T \mathbf{Q} (\mathbf{x}_{t+N|t} - \hat{\mathbf{x}}) \tag{19}$$

4. RESULTS

The controllers have been tested on a rigid body simulation using *drake toolbox* [17]. *SNOPT solver* [8] is used to solve the non-convex QP and MPCC. In addition, the solver *LCQPow* [9] represents a specialized LCQP solver. To efficiently solve the MIP problem, *SCIPSolver* [3] with python bindings [13] is used. All experiments were performed on standard hardware with a 12th generation I7 processor and 32 GB of RAM.

4.1. Feasible region

The feasible regions of the different formulations are calculated to compare their ability to find solutions. The initial state is $\mathbf{x}_0 = [x_0, 0\,\mathrm{m}, 45°, \dot{x}_0, 0\,\frac{\mathrm{m}}{\mathrm{s}}, 0\,\frac{\mathrm{rad}}{\mathrm{s}}]^T$, where $x_0, \dot{x}_0$ take different values to analyze the feasibility in state space. The final velocity constraint is set to the limit cycle bounds (4). Position error and thruster usage are minimized via the cost function. Fig. 4 shows the feasible region for the MILP formulation for different prediction horizons. For each initial condition, the feasible solutions are marked according to their solving time t_s. A star indicates $t_s \leq 0.1s$, which was chosen as the desired real-time solving rate. Due to the final velocity constraint, the maximum initial $\dot{x}$, for which the solver finds a solution, is limited. For large values, the solver is not able to enter the velocity limit cycle within the prediction horizon. A higher prediction horizon leads to a bigger feasible region. For a prediction horizon $N = 30$, the solving time is up to 2 s for certain regions, indicated by the red areas in Fig. 4. For the longer prediction horizon, this becomes even more evident. However, the triangles indicate that there was at least one feasible suboptimal solution available at $t_s \leq 0.1s$.

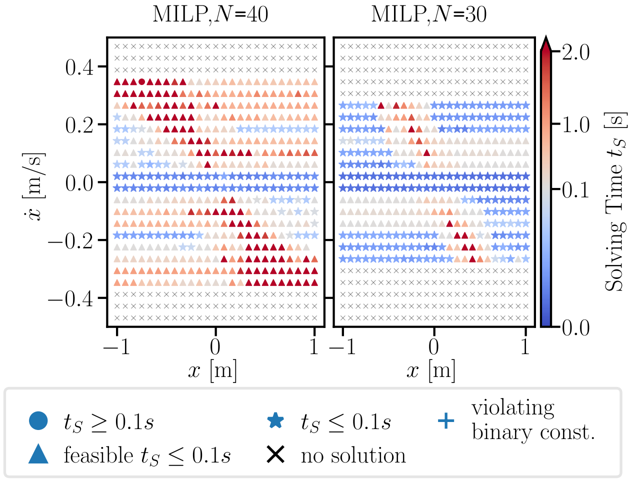

Figure 4. Feasible region for MILP under final limit cycle velocity constraint. Position error and thruster usage are minimized via cost.

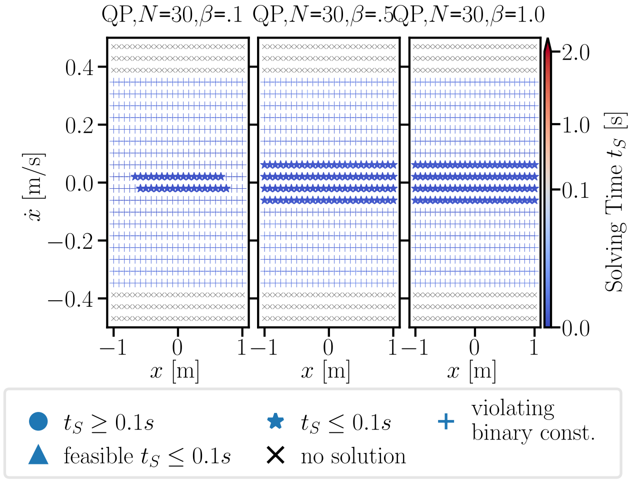

Figure 5. Feasible region for QP under final limit cycle velocity constraint. Position error and thruster usage are minimized via cost. The different plots refer to different weightings of the penalty cost.

Fig. 5 shows the feasible regions for the QP formulation for a prediction horizon of $N = 40$ and different weightings of the penalty term. The feasible regions are bigger and the solving time is significantly lower than for the MILP. However, the plus indicates that for most of the initial conditions, the penalty costs couldn't be minimized. Hence, the input values do not take a binary value. No significant improvement can be observed through higher weighting of the penalty term.

Experiments with the MPCC have shown that for a few initial conditions, a solution can be found mostly below $0.05s$. However, most of the time, all solvers fail to find a solution. One option is to relax the constraint. However, choosing the relax parameter, in general, is not practical because it introduces non-binary values to the solution.

Since the objective of this paper is an MPC that optimally controls the system while respecting binary and timing constraints, only the MILP formulation is appropriate. The LCP formulation provides only partial or no solution, while the QP formulation often yields non-binary inputs, which may require rounding up or down in a preliminary step. The timing constraints and theoretical assumptions however assume binary inputs, which means that in reality, these constraints will not be met. Therefore, the MILP formulation is chosen. If an optimal solution is not found within 0.1 s, suboptimal solutions are used. The impact of suboptimally (i.e., prediction horizon) on suboptimally is analyzed in the following.

4.2. Closed loop simulation

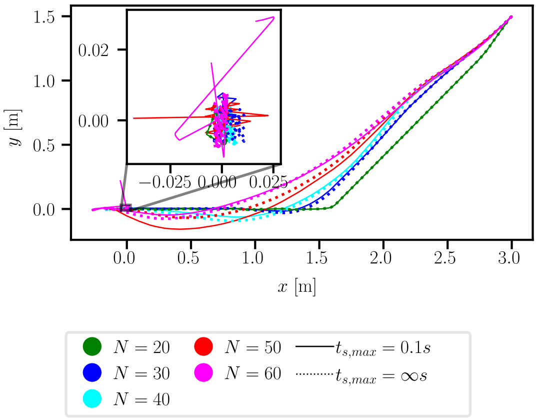

Figure 6. System trajectories under MPC control law for different prediction horizons for optimal solver solutions and suboptimal, fast solutions.

Fig. 6 shows the trajectory of the system for the closed-loop simulation for different prediction horizons. For each horizon, there exist two trajectories. One where the optimal result is taken as the control input and one where the most optimal solution within 0.1 s is taken, to meet the real-time requirement. The initial system state is $x = [3.0m, 1.5m, 90°, 0, 0, 0]^T$, and the target to minimize it towards the origin (i.e., $x = [0, 0, 0, 0, 0, 0]^T$). Fig. 7 shows the time that the solver took to find the optimal solution for each time step for the different prediction horizons. Only horizon $N = 20$ never exceeds the target limit of 0.1 s. For $N = 30$ the optimal solution was found not within 0.1 s for some time steps at the beginning. For the rest of the experiment, the optimal solution could be found within 0.1 s. For $N = 40$ the solver time oscillates around 0.1 s, while $N = 50$ requires especially at the beginning more considerably more time. For $N = 60$, the solver takes on average more than 1 s to find the optimal solution. Note that for all these tests, a feasible solution was always found within 0.1 s. The different prediction horizons also reflect in the system trajectories obtained by the respective controllers (Fig.6). It is worth noting that for values of N up to 40, the suboptimal and optimal trajectories are similar. On the other hand, for $N = 50$, a distinct overshoot becomes evident. For $N = 60$ the system position at the origin can not be maintained, and it drifts away.

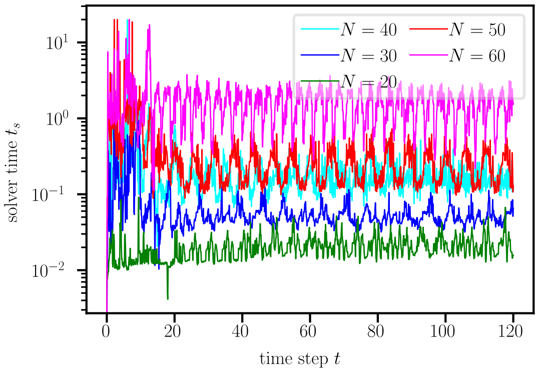

Figure 7. Solving time for MILP solver to find the optimal solution over the whole simulation for different prediction horizons.

A more detailed plot of the system's behavior together with actuation for $N = 40$ under MILP controller is shown in Fig. 8. It shows that the system is steered within 22 s to the origin, where it is oscillating in a limit cycle with ±0.004 m around the origin. The thruster usage at the beginning is high, to accelerate the system and finally break it into the limit cycle. Where then a much lower thruster usage is necessary to maintain the limit cycle.

5. SUMMARY AND CONCLUSION

The comparison of the three different binary constraint formulations shows that only the MILP formulation works in practice. Even if the non-convex

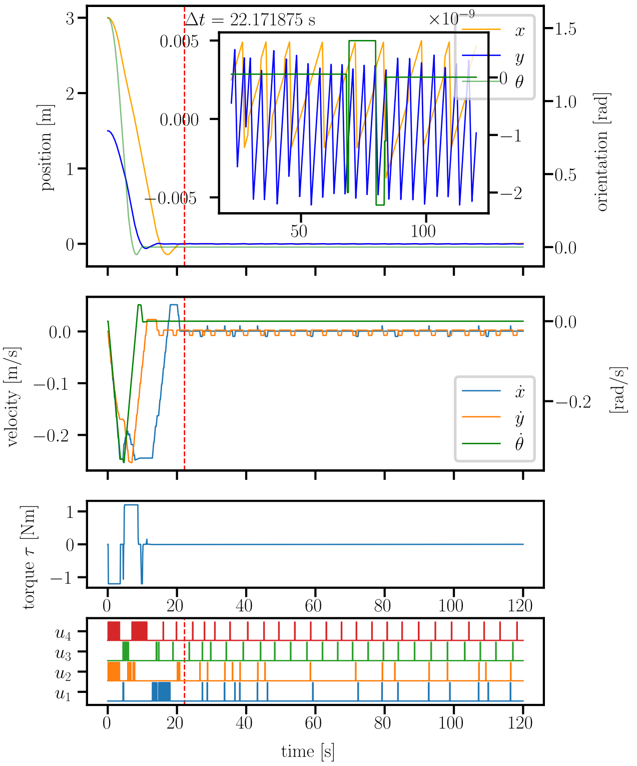

Figure 8. States and input of the system controlled by the MILP MPC towards the origin, where position is maintained.

QP finds optimal results, most of the time the binary constraints are violated. A higher weighting of the penalty function can not counteract this. Either way, this formulation adds another tuning parameter to the problem, which has a critical impact. In addition, the constraints are not guaranteed to be met.
The MPCC formulations were very promising, since the constraints are enforced, whereas the problem can be solved in theory with any non-linear solver. However, it shows that the LCP constraints in this problem are too complex to solve for the standard solver. Also, a special solver tested in this work could not bring any improvement. It is worth noting that there are other solvers available for comparison in future work. However, these solvers often use MIP solving techniques and the results are likely similar.
Although the MILP formulation may not always yield the optimal solution in a reasonable time, it has been demonstrated that it can always identify at least one feasible solution. Therefore, it can be assumed that the controller system always stays feasible. Additionally, it was found that the feasible solution, which is obtained within 0.1 s, has a narrow optimality gap for a sufficiently small prediction horizon. The simulation experiments demonstrate that the MILP MPC formulation can effectively guide the system to the origin and maintain it there.
Furthermore, this work shows a simple linear formulation of the binary input timing constraints. For future work, it would be interesting if these constraints can be generalized and applied to other timings.
The next steps are to extend this to a coupled formulation involving all thrusters and the reaction wheel limits, to fully exploit the capabilities of the system. Secondly, the MPC has to be tested on the real system to determine if it can handle model errors and disturbances by the not perfectly even flat floor.

ACKNOWLEDGMENTS

The first and second authors would like to acknowledge the support of M-RoCk (Grant No.: FKZ 01IW21002) and AAPLE (Grant Number: 50WK2275) and 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 first author would like to thank his university, Universität zu Lübeck, and especially the Institute for Electrical Engineering in Medicine, for their support. Furthermore, many thanks to the *drake*, *SCIPSolver* and *PySCIPOpt* development teams for their tools and active support.

REFERENCES

[1] 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), December 2020. IEEE. ISBN 978-1-72817-447-1. doi: 10.1109/CDC42340.2020.9304087. URL `https://ieeexplore.ieee.org/document/9304087/`.

[2] 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, November 2009. ISSN 1024-123X. doi: 10.1155/2009/750945. URL `https://www.hindawi.com/journals/mpe/2009/750945/`. Publisher: Hindawi.

[3] Ksenia Bestuzheva and others. The SCIP Optimization Suite 8.0. Technical Report, Optimization Online, December 2021. URL `http://www.optimization-online.org/DB_HTML/2021/12/8728.html`.

[4] 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. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4084–4091, July 2022. URL `http://arxiv.org/abs/2207.10693`. arXiv:2207.10693 [cs].

[5] Richard W. Cottle, Jong-Shi Pang, and Richard E. Stone. *The Linear Complementarity Problem.*

SIAM, January 1992. ISBN 978-0-89871-900-0. Google-Books-ID: bGM80_pSzNIC.

[6] David Doman, Brian Gamble, and Anhtuan Ngo. Control Allocation of Reaction Control Jets and Aerodynamic Surfaces for Entry Vehicles. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, August 2007. doi: 10.2514/6.2007-6778. URL https://arc.aiaa.org/doi/10.2514/6.2007-6778.

[7] Roger Fletcher* and ‡ Leyffer, Sven. Solving mathematical programs with complementarity constraints as nonlinear programs. *Optimization Methods and Software*, 19(1):15–40, February 2004. ISSN 1055-6788. doi: 10.1080/10556780410001654241. URL https://doi.org/10.1080/10556780410001654241. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10556780410001654241.

[8] 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. URL https://doi.org/10.1137/S0036144504446096. _eprint: https://doi.org/10.1137/S0036144504446096.

[9] 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. Conference Name: IEEE Control Systems Letters.

[10] 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, January 2020. ISSN 2405-8963. doi: 10.1016/j.ifacol.2020.12.2167. URL https://www.sciencedirect.com/science/article/pii/S2405896320328202.

[11] Jan Leeuwen. *Algorithms and Complexity*, volume 1. Elsevier, September 1990. ISBN 978-0-444-88071-0.

[12] Mirko Leomanni, Andrea Garulli, Antonio Giannitrapani, and Fabrizio Scortecci. An MPC-based attitude control system for all-electric spacecraft with on/off actuators. In *52nd IEEE Conference on Decision and Control*, pages 4853–4858, December 2013. doi: 10.1109/CDC.2013.6760650. ISSN: 0191-2216.

[13] 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. URL https://opus4.kobv.de/opus4-zib/frontdoor/index/index/docId/6045.

[14] 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, August 1968. ISSN 1558-2523. doi: 10.1109/TAC.1968.1098940. Conference Name: IEEE Transactions on Automatic Control.

[15] 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, January 2014. ISSN 0278-3649. doi: 10.1177/0278364913506757. URL https://doi.org/10.1177/0278364913506757. Publisher: SAGE Publications Ltd STM.

[16] 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. In *2015 European Control Conference (ECC)*, pages 2079–2084, July 2015. doi: 10.1109/ECC.2015.7330846.

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

[18] 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, March 2011. doi: 10.1109/AERO.2011.5747482. ISSN: 1095-323X.

[19] Josep Virgili-Llop, Costantinos Zagaris, Hyeongjun Park, Richard Zappulla, and Marcello Romano. Experimental evaluation of model predictive control and inverse dynamics control for spacecraft proximity and docking maneuvers. *CEAS Space Journal*, 10(1):37–49, March 2018. ISSN 1868-2510. doi: 10.1007/s12567-017-0155-7. URL https://doi.org/10.1007/s12567-017-0155-7.

[20] 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*, September 2022. ISSN 0273-1177. doi: 10.1016/j.asr.2022.09.033. URL https://www.sciencedirect.com/science/article/pii/S0273117722008742.

[21] 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.