Stabilization of inexact MPC schemes

Markus Kögel and Rolf Findeisen

Abstract—In model predictive control often the underlying optimization problem is not solved exactly to meet hard real-time bounds or to save computations. This jeopardizes properties of the closed loop system, such as stability, performance or recursive feasibility. We present a framework for linear system with polytopic constraints and quadratic performance criteria, which guarantees recursive feasibility and stability subject to inexact solutions. We combine the approach with simple optimization methods to obtain real-time feasibility and stability.

I. INTRODUCTION

Model predictive control (MPC) is often used to stabilize and "optimally" control constrained linear systems, c.f. [1]–[6]. By now it is well known how stability and recursive feasibility in MPC can be guaranteed, assuming that the underlying optimization problem is solved exactly, c.f. [4]–[6]. However often only inexact solutions of the required optimization problems are available, e.g. for fast systems and for control using low cost hardware.

One challenge in MPC is the solution of the optimization problem, which needs to be done at each time instance. By now many tailored solution approaches exist. For small systems one can use explicit MPC [7], [8], which allows to solve the optimization problem mostly offline and requires online basically only the evaluation of a look up table.

The optimization problem can also be solved online. Several works consider tailored optimization algorithms to allow an efficient solution. Tailored interior-point methods are investigated in e.g. [9]–[16] and specialized active set methods in e.g. [17]. Gradient based methods or so-called first order methods are considered in [18]–[28].

Besides the advancements of the solution speed of the underlying optimization problem it can often not be guaranteed that the exact solution or at least a suboptimal, but feasible solution can be found within the available time. To allow a stabilizing feedback with guaranteed constraint satisfaction the inexact solution needs to be directly considered.

Stability in the presence of inexact solution satisfying the equality constraints due to the dynamics, but not the state and input constraints is guaranteed in the works [20], [22] using constraint tightening. Furthermore, if a feasible solution is available, then one can guarantee, using tailored MPC setups, straightforwardly stability by employing optimization methods, which guarantee cost decrease, if possible, and maintains feasibility, see [16], [29].

We present a framework to guarantee stability subject to inexact solutions consisting of a feasibility recovery scheme and a robustified optimization problem. Compared to existing results we assume that an approximation of the exact solution is available, which satisfies the state and input constraints, but violates the equality constraints (of the dynamics).

In particular, we propose a procedure to remove the mismatch in the dynamics and combine it with ideas from robust MPC, see [4], [30], to guarantee that after this recovery the constraints are still satisfied. The framework can be combined with tailored optimization methods to obtain stabilizing control laws with hard bounds on the computational complexity. The considered class of optimization methods is general, e.g. it includes also distributed methods, such as [27]. As an example we discuss a combination of a quadratic penalty method with Nesterov’s fast gradient method, c.f. [31], [32] and illustrate it with an example.

The remainder of this paper is structured as follows. Section II states the framework and the problem. Section III contains as main result a framework to guarantee stability and constraint satisfaction in the presence of inexact solutions. In Section IV we combine the framework with simple optimization methods to obtain a real-time control scheme.

The notation is standard. A ⊕ B, A ⊖ B denote the Minkowski sum, Minkowski difference, see [4]. For a matrix M, M > 0 means that M = Mᵀ and M is positive definite.

II. PROBLEM FORMULATION

This section outlines first the considered problem class and reviews MPC based on exact optimization.

A. Considered problem class

We consider constrained, linear, discrete time systems

\[ x(k + 1) = Ax(k) + Bu(k), \]
\[ x(k) ∈ X, \ u(k) ∈ U, \]

where \( x(k) ∈ ℝ^n \) is the state and \( u(k) ∈ ℝ^p \) the input. For simplicity of presentation we assume that the system is controllable. The sets \( X \) and \( U \) are compact, convex polytopes and contain the origin in their interior.

B. MPC based on exact optimization

A common method to stabilize the system (1) and guarantee constraint satisfaction is MPC. In MPC an input sequence \( u_k \) and a state trajectory \( x_k \) defined over a horizon \( N \)

\[ X_k = (x^T_{k|k} \cdots x^T_{k+N|k}), \ u_k = (u^T_{k|k} \cdots u^T_{k+N-1|k})^T, \]

[2]
are determined. For consistency with the dynamics and current state \( x(k) \) the sequences \( u_k \) and \( x_k \) need to satisfy
\[
\begin{align*}
x_{i+1 | k} &= A x_i | k + B u_i | k, \quad i = k, \ldots, k + N - 1, \\
x_k | k &= x(k).
\end{align*}
\] (3a) (3b)

Furthermore, often the state \( x_{k+N | k} \) should be in a given terminal set \( \mathcal{X}_T^f \). The set \( \mathcal{X}_T^f \) contains the origin in its interior and is a convex, compact polytope. Consequently, the constraints \( x_k \in \mathcal{X}, \ u_k \in \mathcal{U} \) need to hold where
\[
\begin{align*}
\mathcal{X} &= \mathcal{X} \times \ldots \times \mathcal{X} \times \mathcal{X}_T^f, \\
\mathcal{U} &= \mathcal{U} \times \ldots \times \mathcal{U}.
\end{align*}
\] (4)

To guarantee stability and tune the control performance \( u_k \) and \( x_k \) minimize a quadratic objective \( J(x_k, u_k) \) given by
\[
J(x_k, u_k) = \sum_{j=k}^{N+k-1} u_j^T R u_j + \sum_{j=k+1}^{N+k-1} x_j^T Q x_j
\] (5)

which is for simplicity assumed to be strongly convex, i.e. \( Q > 0, R > 0, F > 0 \).

Summarizing, ideal MPC, i.e. without errors or uncertainties due to the optimization, solves in each time step the optimization problem \( O(x(k), \mathcal{X}, \mathcal{U}) \) given by
\[
O(x(k), \mathcal{X}, \mathcal{U}) : \min_{x_i \in \mathcal{X}, u_i \in \mathcal{U}} J(x_i, u_i) \text{ s.t. } (3),
\] (6)

Exactly. The applied input is given by the first part of the optimal input: \( u(k) = u_k^* \). Note that the given optimization problem is a convex quadratic program (QP).

By now it is well understood how the above framework can be tuned to guarantee stability and recursive feasibility, for example by a suitable selection of the terminal constraint \( \mathcal{X}_T^f \) and terminal penalty \( F \), see [4]–[6].

Unfortunately, many stability and recursive feasibility results require the exact solution. The assumptions to guarantee stability can be weakened to allow suboptimal, but feasible solutions, see e.g. [4], [6], [16], [29]. However, this does not significantly simplify the problem, because in general finding a suboptimal, feasible solution of (6) can be almost as challenging as finding the optimal solution (6), compare [31]–[33].

C. MPC based on inexact optimization

We consider an approximate solution of the problem \( O(x(k), \mathcal{X}, \mathcal{U}) \) and propose a framework to guarantee stability. To this end we introduce the notion of \( (\epsilon, M) \)-approximate solutions of the problem \( O(x(k), \mathcal{X}, \mathcal{U}) \).

**Definition 1:** \((\epsilon, M)\)-approximate solution

An \( (\epsilon, M) \)-approximate solution \( \hat{x}, \hat{u} \) of \( O(x(k), \mathcal{X}, \mathcal{U}) \) with a suboptimality \( \epsilon \) and mismatch \( \hat{m} \) satisfies
\[
\begin{align*}
&J(\hat{x}, \hat{u}) - \min_{x \in \mathcal{X}, u \in \mathcal{U}} J(x, u) \leq \epsilon, \\
&\hat{x} \in \mathcal{X}, \ \hat{u} \in \mathcal{U}, \\
&\hat{x}_{i+1 | k} = A \hat{x}_i | k + B \hat{u}_i | k + \hat{m}_i | k, \ \hat{x}_k | k = x(k), \\
&\hat{m} = (\hat{m}_k^T \ldots \hat{m}_{N-1 | k}^T)^T, \ \hat{m} \in M,
\end{align*}
\] (7a) (7b) (7c) (7d)

where \( M \) is a convex, compact set containing the origin. Clearly, with \( \epsilon = 0 \) and \( M = \{0\} \) (7) is the exact solution of \( O(x(k), \mathcal{X}, \mathcal{U}) \). This definition is different from [20], [22], because we assume the approximate solution violates the dynamics (1a), but not the constraints (7b).

The sets \( M \) can be rather general, e.g. one can use for \( M \) norm balls: \( M = \{m\ s.t. \|m\| \leq \beta\}, \ \beta > 0, [33] \).

In the above definition the mismatch \( M \) is independent of the actual state \( x(k) \). However the sets \( \mathcal{X} \) and \( \mathcal{U} \) contain a neighborhood of the origin, so the unconstrained solution of (6), a finite horizon LQR controller with the structure
\[
u_k = G x_k, \quad u_k | k = G x_k | k,
\] (8)

is locally admissible, compare [4]. The states for which the LQR is admissible are in the following denoted as:

Definition 2: (Admissible set of LQR control law)

For a controller \( u = G x \) we define by \( \mathcal{X}_0^G \) the set of admissible states, i.e. for every \( x \in \mathcal{X}_0^G \)
\[
x_{i+1} = (A + BG)x_i \in \mathcal{X}, \quad u_i = G x_i \in \mathcal{U},
\]
\[
x_0 = x, \quad x_{k+N} \in \mathcal{X}_T^f.
\]

where \( i = 0, \ldots, N - 1 \).

Now we can state the problem considered in this work.

**Problem 1:** (Stabilizing, inexact MPC)

Choose constraints \( \mathcal{X}, \mathcal{U}, \mathcal{X}_T^f \) and a terminal penalty \( F \) such that for a given \( \mathcal{M} \) and \( \epsilon \), if \( x \notin \mathcal{X}_0^G \), then any \( (\epsilon, \mathcal{M}) \)-approximate solution of \( O(x(k), \mathcal{X}, \mathcal{U}) \) guarantees stability and constraint satisfaction.

III. MAIN RESULTS

We first present the mismatch matching procedure, which allows to remove the mismatch in the equality constraints. Then we outline how the constraints can be tightened to guarantee that after removing the mismatch the state and input constraints are still satisfied. This allows to directly obtain a stabilizing feedback. In a second step we extend the approach such that we can guarantee that the feasible region is time invariant.

A. Mismatch matching procedure

This sections outlines the mismatch matching procedure to guarantee constraint satisfaction and stability for approximate solutions of (6). The key idea to remove the mismatch in (7c) resulting from an inexact solution is to add
\[
\begin{align}
\Delta x_{i+1 | k} &= (A + BL)\Delta x_i | k - \hat{m}_k | k, \quad \Delta x_k | k = 0, \\
\Delta u_i | k &= L \Delta x_i | k, \\
\end{align}
\] (9a) (9b)

to \( \hat{u}_k \), \( \hat{x}_k \), i.e.
\[
\hat{x}_k = \hat{x}_k + \Delta \hat{x}_k, \quad \hat{u}_k = \hat{u}_k + \Delta \hat{u}_k, \quad (10)
\]

which results in
\[
\begin{align}
\nabla_{i+1 | k} &= A \nabla_i | k + B \nabla_i | k, \quad \nabla_{k} | k = x(k), \\
\nabla_k | k &= \hat{u}_k | k + L \Delta x_i | k, \quad (11a) (11b)
\end{align}
\]

Observe that this procedure leads to \( \nabla_k \) and \( \nabla_k \), which satisfy the dynamics exactly, i.e. without any mismatch.
Unfortunately, if \( \tilde{x}_k \in X \) and \( \tilde{u}_k \in U \), then this does not guarantee in general that \( x_k \in X \) and \( u_k \in U \).

Note that the gain \( L \) in (9) is an additional degree of freedom in the dynamics (9). It can be used to shape \( \Delta x_k, \Delta u_k \). In particular, for unstable systems choosing a stabilizing \( L \) can limit the growth of \( \Delta x_{i:k} \). Note that the \( \tilde{m}_k \) is given by (7c). So the combination of (7c) with this procedure is a map from \( \tilde{u}_k, \tilde{x}_k \) to \( \tilde{u}_k, \tilde{x}_k \) denoted by
\[
(\tilde{u}_k, \tilde{x}_k) = \mathcal{M}(\tilde{u}_k, \tilde{x}_k).
\]

Adding \( \Delta x_k \) to \( \tilde{x}_k \) can be interpreted as a disturbance \( \tilde{m}_k \) acting onto the entire state sequence \( \tilde{x}_k \). So we can compute the constraint violation easily using Minkowskii sums, see e.g. [4]. Let us first express \( \Delta x_k \) and \( \Delta u_k \) as
\[
\Delta x_k = C^L \tilde{m}_k, \quad \Delta u_k = D^L \tilde{m}_k,
\]
where \( C^L, D^L \) are given in (35) in the Appendix A. Consequently, the state trajectory \( \tilde{x}_k \) and input sequence \( \tilde{u}_k \) obtained by \( \mathcal{M} \) (12) satisfy
\[
\tilde{x}_k \in \tilde{x}_k + C^L M, \quad \tilde{u}_k \in \tilde{u}_k + D^L M.
\]

This estimate allows later on to guarantee constraint satisfaction and stability.

B. Feasibility recovery and feasibility induced stability

Solving \( O(x(k), \tilde{X}, \tilde{U}) \) with tightened constraints (compare [4], [30]) and using the map \( \mathcal{M} \) (12) one can obtain a suboptimal, but feasible solution to \( O(x(k), \tilde{X}, \tilde{U}) \) outlined in the next proposition as well as an estimate of the resulting cost increase.

**Proposition 1: (Feasibility recovery)**

If \( L, \tilde{X} \) and \( \tilde{U} \) are such that
\[
X \supseteq \tilde{X} \oplus C^L M, \quad X \supseteq \tilde{U} \oplus D^L M,
\]
then the state sequence \( \tilde{x}_k \) and the input sequence \( \tilde{u}_k \) obtained by (12) satisfy the constraints \( \tilde{x}_k \in X, \tilde{u}_k \in U \) and the dynamics (3). Moreover, the cost increase due to (12) is bounded by
\[
J(\tilde{x}_k, \tilde{u}_k) \leq J(x_k, u_k) + \| \Delta x_k \|^2 Q + \| \Delta u_k \|^2 R + \| \Delta x_k \|^2 Q + \| \Delta u_k \|^2 R,
\]

where \( q_{\max} = \max_{x \in X} ^{2Qx} \) and \( r_{\max} = \max_{u \in U} ^{2Ru} \).

For the proof see Appendix B.

This result can be straightforwardly extended to obtain directly a stabilizing feedback utilizing warm starting and tailored optimization algorithms. We first make some assumptions on the terminal set and terminal penalty.

**Assumption 1: (Positive invariant terminal set)**

There is a terminal controller \( u = Kx \) and terminal constraint \( \tilde{x}_f \) such that for every \( \in X^f \)
\[
x \in X, \quad Kx \in U, \quad (A + BK)x \in X^f.
\]
A common choice for \( K \) is the control gain of the unconstrained, infinite horizon LQR controller with the same stage cost as in the performance index (5). Note that if the terminal penalty \( F \) is chosen as the solution of the LQR Riccati equation, then \( K = G \) compare [4]–[6].

The assumption allows to obtain from a feasible and suboptimal solution \( \tilde{x}_k, \tilde{u}_k \) of \( O(x(k), X, U) \) and for the nominal dynamics \( x(k+1) = Ax(k) + Bu(k), u(k) = u_k(k) \) a feasible and suboptimal state sequence \( x_{k+1} \) and input sequence \( u_{k+1} \) of \( O(x(k), X, U) \)
\[
x_{i+1|k+1} = \tilde{x}_{i+1|k}, \quad x_{N+1|k+1} = (A + BK)\tilde{x}_{N+1|k},
\]

compare [4]–[6]. To guarantee stability also the terminal penalty needs to be chosen suitably.

**Assumption 2: (Terminal penalty)**

The terminal penalty matrix \( F \) and terminal control gain \( K \) satisfy
\[
F \geq (A + BK)^T F (A + BK) + Q + K^T R K.
\]

The warm-starting (17) and the chosen terminal penalty guarantee the following cost bound
\[
J(\tilde{x}_{k+1}, u_{k+1}) \leq J(\tilde{x}_k, \tilde{u}_k) + x(k)^T Q x(k),
\]
leading to:

**Corollary 1: (Stability using warm-starting)**

Let Assumption 1 and 2 hold. Let \( \varepsilon > 0, M, \tilde{X} \) and \( \tilde{U} \) be given such that (15) holds. Assume that, if \( k = 0 \), we solve \( O(x(0), \tilde{X}, \tilde{U}) \) with a suboptimality \( \varepsilon \) and mismatch \( M \) and for any \( k > 0 \) we utilize an optimization method, which maintains feasibility and decreases the cost.

Then \( x_k \in X, u_k \in U \) and \( x_k \rightarrow 0 \) for \( k \geq 0 \). Moreover, if for \( x(0) \in A_0^c \), \( u(0) = Gx(0) \), then the origin is stabilized.

The proof can be found in the Appendix C.

The above corollary allows to guarantee stability utilizing a precise enough, inexact solution at \( k = 0 \). Note that only at \( k = 0 \) the problem needs to be solved and afterward it is enough to use (17). The major drawback of the proposed solution is that the feasible region is not time invariant, i.e., if \( O(x(j), \tilde{X}, \tilde{U}) \) is feasible for \( x(j) \) with \( j = 0 \), then this does not imply that this holds for any \( j > 0 \).

**Remark 1: (Maximum acceptable error)**

Note that the major restriction on acceptable errors is that the sets \( \tilde{X}, \tilde{U} \) given by (15) are nonempty. In particular, there are no restrictions on the suboptimality. This is to expect, because only feasibility of the initial step needs to be guaranteed explicitly, recursive feasibility and stability are obtained directly from (17), c.f. [16], [29].

C. Stability and robust feasibility

Now we present an extension of the previous approach, which does not rely on warm-starting allowing to obtain a time-invariant feasible region. To this end we require:

**Assumption 3: (Robust positive invariant terminal set)**

There is a terminal controller \( u = Kx \) and terminal constraint \( \tilde{x}_f \) such that for every \( x \in \tilde{X}_f \)
\[
\tilde{X}_f \subseteq X \oplus C^L M \oplus \ldots \oplus C^N M,
\]
\[
Kx \in U \oplus D^1 M \oplus \ldots \oplus D^N M \oplus KC^N M,
\]
\[
(A + BK)x \in \tilde{X}_f \oplus (A + BK)C^N M.
\]
Note that for small enough errors $M$ this assumption can be always satisfied. A set $\hat{X}^f$ can be computed as follows: first one determines the tighten constraints on the right hand side of (19a), (19b). If $M$ is a polytope/ellipsoid, then this can be done with linear programming/second order cone programming, [33]. Second, one computes bounds for $(A + BK)^T C_k^f M$ with a polytope and finally one computes $\hat{X}^f$ for the tightened constraints in (19a), (19b) using computational geometry, e.g. with [34].

The above assumptions guarantees feasibility and repeated feasibility of the recovering scheme (12).

Proposition 2: (Feasibility and repeated feasibility) Let Assumption 3 hold and $\hat{X}, \hat{U}$ be given by
\[
\hat{X} = X \times \hat{X}_1 \times \ldots \times \hat{X}_f, \\
\hat{U} = U \times \hat{U}_1 \times \ldots \times \hat{U}_{N-1},
\]
(20a)
\[
\hat{X}_i \subseteq X \ominus C_i^f M \ominus \ldots \ominus C_i^f M, \\
\hat{U}_i \subseteq U \ominus D_i^f M \ominus \ldots \ominus D_i^f M.
\]
(20b)
(20c)
(20d)
For any $\hat{x} \in \hat{X}$, $\hat{u} \in \hat{U}$ using $(\hat{\mu}_k, \hat{x}_k) = M(\hat{u}_k, \hat{x}_k)$ guarantees
- $\hat{x}_k, \hat{u}_k$ satisfy (3)
- $\hat{x}_i \in \hat{X}$ for $i = k, \ldots, k + N - 1$
- $\hat{u}_i \in \hat{U}$ for $j = k, \ldots, k + N - 1$
- For $x(k+1) = Ax(k) + Bu(k)$, $u(k) = \hat{u}_k$ there exist $x_{k+1}$ and $u_{k+1}$ satisfying (3), $x_{k+1} \in X$ and $u_{k+1} \in U$

The proof is provided in Appendix D.

Similar as Proposition 1 this guarantees feasibility of the recovering scheme (12), but also recursive feasibility under inexact solution. To guarantee stability we make two additional assumptions.

Assumption 4: (Bounded cost increase of recovery (12)) There exists a constant $\Delta J(M)$ such that for every $x_0 \in X$, $\hat{u}_k \in \hat{U}$ and the procedure $(\hat{x}_k, \hat{u}_k) = M(x_k, \hat{u}_k)$ increases the cost maximally by
\[
J(\hat{x}_k, \hat{u}_k) - J(\hat{x}_k, \hat{u}_k) \leq \Delta J(M).
\]
(21)

Computing the exact value of $\Delta J(M)$ is challenging, because it would require to maximize the left hand side of (21) over $x(k)$, which is a bi-level program. However an upper bound on $\Delta J(M)$ can be easily be computed based on (16) of Proposition 1.

Assumption 5: (Admissible set of terminal control law) A set $X^G_0 = \{ x \in X | x \leq \gamma \}$ exists, where $\gamma > 0$ and for every $x \in X_0$ for $i = 0, \ldots, N - 1$
\[
x_{i+1} = (A + BG)x_i \in \hat{X}_i, \\
u_i = Gx_i \in \hat{U}_i,
\]
\[
x_0 = x(k), \\
x_{k+1} \in \hat{X}_f.
\]
Note that this set exists, if the sets $\{\hat{X}_i\}, \{\hat{U}_i\}$ and $\hat{X}_f$ contain each their interior and it can be efficiently computed.

Corollary 2: (Stability using inexact solution) Let Assumptions 2, 3 and 5 hold. Moreover, let $u(k) = Gx(k)$ if $x(k) \in X^G_0$, otherwise let $u(k) = \hat{u}_k$, where $\hat{u}_k$ and $\hat{u}_k$ is an $(\epsilon, M)$-approximate solution of $O(x(k), \hat{X}, \hat{U})$. If for any $c > 0$
\[
\Delta J(M) + \epsilon \leq (1 - c) \gamma
\]
(22)
then the closed loop system is asymptotic stable.

The proof can be found in the Appendix E.

Remark 2: (Maximum suboptimality and mismatch) Compared to the feasibility based approach (Corollary 1 maximum acceptable error of an solution is further restricted. First the mismatch needs to be small enough to guarantee that Assumption 3 is satisfied, which implies that the sets (20) are nonempty. Moreover, the suboptimality and mismatch need to be small enough to guarantee (22).

IV. QUADRATIC PENALTY, FAST GRADIENT METHOD

This section outlines how an $(\epsilon, M)$ approximate solution, c.f. Definition 1, can be obtained utilizing the quadratic penalty approach (see e.g. [31]) and Nesterov’s fast gradient method [32] for the case of box constraints. In detail, we present a method to choose the penalty factor and the minimum number of iterations to guarantee these bounds. Note that this algorithm is similar to the inner problem in the algorithm presented in [23], which does not provide a convergence analysis, but contains implementation details.

A. Quadratic penalty method

We consider a quadratic penalty method, which softens the equality constraints due to the dynamics by a quadratic penalty, but strictly enforces the inequality constraints (the state and input constraints). The method is based on the inexact solution of the following problem
\[
\min_{x_k \in X, u_k \in U} J^P(x_k, u_k, x(k)),
\]
(23)
where $\rho > 0$ denotes the penalty factor, compare [31] and $J^P(x_k, u_k, x(k)) = J(x_k, u_k) + P(x_k, u_k, x(k))$,
(24)
\[
P(x_k, u_k, x(k)) = \rho \sum_{i=k+1}^{k+N-1} \|Ax_i[k] + Bu[i,k] - x_{i+1}[k]\|^2
\]
\[
+ \rho \|Ax(k) + Bu[k] - x_{k+1}[k]\|^2.
\]
The quadratic penalty method is inexact, i.e. the minimizer of (23) does not guarantee an exact satisfaction of the equality constraints (3) and is in general different from (6). Note that this is not a problem, because we focus on inexact solutions.

The choice of $\rho$ influences the mismatch $M$ the condition of problem (24). The following proposition provides a way to choose $\rho$ and the accuracy of the solution of (23) such that the mismatch can be bounded and a specific suboptimality can be guaranteed.

Proposition 3: (Bounds on mismatch and suboptimality) If for any $x(k)$ such that $O(x(k), \hat{X}, \hat{U})$ is feasible, there exist $\alpha, \epsilon, \rho$ such that\[
J^P(x_k^e, u_k^e, x(k)) - J^P(x_k^e, x_k^e, x(k)) \leq \epsilon,
\]
(25a)
\[
\max_{x(k)} \|Ax(k) + Bu[k] - x_{k+1}[k]\|_\infty \leq \alpha,
\]
(25b)
\[
\max_{x(k)} \|Ax_i[k] + Bu[i,k] - x_{i+1}[k]\|_\infty \leq \alpha,
\]
(25c)
\[
(x_k^e, u_k^e) \in \arg \min_{x_k \in X, u_k \in U} J^P(x_k, u_k, x(k)),
\]
(25d)
\footnote{Here we consider the sets $\hat{X}, \hat{U}$. They can be replaced by $\hat{X}, \hat{U}$. We assume that $x(k) \in X$.}
where \( i = k + 1, \ldots, k + N - 1 \), then \( x^k_i, u^k_i \) is an \((\varepsilon, M)\)-approximate solution of the problem \( O(x(k), \hat{X}, \hat{U}) \) with \( M = M_1 \oplus M_2 \) where \( M_1 = \{ z \text{ s.t. } \|z\| \leq \beta_k \} \) and \( M_2 = \{ z \text{ s.t. } \|z\|_\infty \leq \alpha \} \).

Note that the condition (25a) describes how accurate (23) needs to be solved. The verification of (25b) and (25c) requires to solve bilevel optimization programs, which can be done for example using mixed integer linear programming.

### B. Nesterov’s gradient method

As already mentioned, we assume that the constraints \( \hat{X}, \hat{U} \) are given by boxes, i.e. that

\[
\begin{align*}
\hat{X}_{i|k} & = \{ x \text{ s.t. } x_{i|k}^l \leq x \leq x_{i|k}^u \}, \\
\hat{U}_{i|k} & = \{ u \text{ s.t. } u_{i|k}^l \leq u \leq u_{i|k}^u \}, \\
\hat{X}_f^{k+N|k} & = \{ x \text{ s.t. } x_{k+N|k}^l \leq x \leq x_{k+N|k}^u \}.
\end{align*}
\]

Note that this assumption is rather conservative; even if the state constraints \( X \) and input constraints \( U \) are boxes the terminal set \( \hat{X}_f^{k+N|k} \) is in general not a box. Fortunately, this restriction can be relaxed, e.g. if \( A + BK \) is diagonalizable, then one can use a box constraint on the terminal state using a state transformation. Alternatively, it might be possible to use a cyclic varying horizon to obtain box constraints [35].

To simplify the presentation let us denote

\[
z = (u_k^T, x_{k+1}^T, \ldots, u_{k+N-1}^T, x_{k+N}^T),
\]

which allows to formulate (23) by

\[
J^P(z, x(k)) = \frac{1}{2} z^T H_a z + z^T G_a x(k) + \text{const.},
\]

This allows to direct use Nesterov’s gradient method. For this problem Nesterov’s gradient method is given by the following algorithm

\[
\begin{align*}
z^i & = \mathcal{P} \left( y^{i-1} - \frac{(H_a y^{i-1} - G_a x(k))}{\mu_a} \right), \\
y^i & = z^i + \beta_i (z^i - z^{i-1}), \\
\alpha_{i+1} & \text{ s.t. } (1 - \alpha_{i+1}) \alpha_i + \frac{1}{\mu_a} \alpha_{i+1} = \alpha_i \in (0, 1), \\
\beta_{i+1} & = \frac{\alpha_i (1 - \alpha_i)}{(\alpha_i)^2 + \alpha_{i+1}}.
\end{align*}
\]

where \( i \geq 1 \) denotes the iteration index and which is initialized here with \( z^0 = y^0 = 0 \) and \( \alpha_0 \in (0, 1) \) and \( \mathcal{P} \) denotes an entry-wise saturation

\[
\mathcal{P}(z) = \max(\min(z^{lb}, z), z^{ub}).
\]

and \( \mu_a = \lambda_{\text{Max}}(H_a), \) \( \lambda_a = \lambda_{\text{Min}}(H_a) \) the minimum / maximum eigenvalues. The convergence rate and thus the maximum number of iterations can be computed by the following theorem.

**Theorem 1:** (Convergence of Fast gradient method) The iterates \( z^i \) \((28)\) satisfies

\[
J^P(z^W, x(k)) = \min_{z^{lb} \leq z \leq z^{ub}} \psi(z, x(k)) \leq \psi \omega_i D_x,
\]

where with \( \omega_0 = 1 \)

\[
\begin{align*}
\psi & = \rho \left( 1 + \frac{\alpha_i^2 H_a - \beta_a \alpha_i}{2(1 - \alpha_i)} \right), \\
D_x & = \max_{z \in \mathcal{X}} \|x\|^2_2, \\
\omega_i & = \alpha_i \omega_{i-1}.
\end{align*}
\]

The proof of this work is avoided here due to space limitations, it is based on [32], [36] and the use of \( z = 0 \) as feasible initial guess. Note that we use here the explicit expression of \( \lambda_i \), which yields tighter results in contrast to the asymptotic approximation in [32, Thm. 2.2.3.] and [36].

Note that more methods exist to improve the convergence speed and improve the bounds. However due to space limitations a detailed discussion of this topic is avoided here.

### V. Simple example

We consider a double integrator with box constraints:

\[
x(k + 1) = (0, 0.2) x(k) + (0.02, 0) u(k),
\]

\[
X = \{ x \text{ s.t. } \|x\|_\infty \leq 1 \}, \quad \omega_1 \text{ s.t. } \|u\|_1 \leq 1, \quad \omega_2
\]

with \( Q = I, R = I \) and the terminal control law, terminal set \( \hat{X}^f \) and terminal penalty \( F \) chosen such that Assumption 1 and 2 hold.

Let us first illustrate the effect of feedback in the mismatch matching procedure (12) on the size of the sets \( \hat{X}_i \) and \( \hat{U}_i \) (Proposition 2). In Figures 1, we illustrate the size of the set \( \hat{X}_i \) \((20)\) for \( L = 0 \) (blue, black) and with \( L \) as infinite horizon LQR gain (red, green) for \( M = M_1 \oplus M_2 \) with \( M_1 = \{ z \text{ s.t. } \|z\|_2 \leq 10^{-4} \} \) and \( M_2 = \{ z \text{ s.t. } \|z\|_\infty \leq \zeta \} \) with \( \zeta = 10^{-3} \) (green, black) and with \( \zeta = 5 \cdot 10^{-4} \) (blue, red) using \( N = 30 \). We observe that for the large mismatch and for \( L = 0 \) the size of the sets \( \hat{X}_i \) decreases faster. In particular, in the black case the sets are empty for \( i > 27 \). Figure 2 shows the sets \( \hat{U}_i \). For the green cases \((L \neq 0, \) larger \( M_2)\) the set shrinks significantly.

Now let us investigate if the condition \((22)\) in Corollary 4 holds. For this example we have a simple, but conservative estimate of \( \Delta J(M) \) is

\[
\begin{align*}
\Delta J(M) \leq & \|F\|_2 \|2 + \zeta + (\epsilon + \zeta) \\
& \{(2 + \epsilon + \zeta \sqrt{N(p + n)}) \} \}
\end{align*}
\]

With \( \epsilon \) as above and \( \zeta = 10^{-3} \) we get that \( \Delta J \leq 0.081 / \Delta J \leq 0.057 \). Since we have \( \gamma = 0.3 > \Delta J(M) + \epsilon \) the condition \((22)\) holds for all cases.

Inspired by the above figures we want to design the optimization algorithm with \( L \neq 0 \) as given above such that the above assumptions on \( M \) hold and that stability is guaranteed based on Proposition 3. To satisfy the assumptions on \( M_1 \) we need to choose here \( \epsilon = 1e - 4 \). We can guarantee that \( M_2 = \{ z \text{ s.t. } \|z\|_\infty \leq 10^{-3} \} \) hold by \( \rho = 10^4 \) and \( M_2 = \{ z \text{ s.t. } \|z\|_\infty \leq 5 \cdot 10^{-4} \} \) holds for \( \rho = 3 \cdot 10^4 \).

The number of required iterations of the quadratic penalty, fast gradient method optimization method are illustrated in Table 1.
We compare different setups and different values for $\alpha_0$:

$$\alpha_0 = \sqrt{\frac{b_i}{h_i}},$$

which results in $\alpha_i = \alpha_0$, $\forall i > 0$ and a large $\alpha_0$:

$$\alpha_0 \in (0, 1) \text{ s.t. } 10h_0 = \frac{\alpha_0(\alpha_0 T_h - b_i)}{1 - \alpha_0} \quad (34)$$

resulting in a monotonic decreasing $\alpha_i$ with $\alpha_i \rightarrow \sqrt{\frac{b_i}{h_i}}$.

For this example the proposed approach seems to deliver promising results. Extensive evaluations and comparison with other methods are necessary, but due to the lack of space not included here.

VI. SUMMARY AND OUTLOOK

This work presented a framework to handle inexact solutions of the underlying optimization problem in MPC, which satisfy state and input constraints, but violate the dynamics. A combination of a feasibility recovery scheme with a robustified MPC setup is proposed to guarantee stability and constraint satisfaction even for inexact solutions. We outlined how this enables real-time bounds by considering an optimization method combining a penalty approach with Nesterov’s gradient method.

In future works we plan an extensive evaluation of the proposed approach as well as a comparison with other approaches. Furthermore, we envision an extension to robust model predictive control and control tasks beyond regulation such as trajectory tracking based on [37].

REFERENCES

Appendix

A. Matrices $C^L$, $D^L$

\[
C^L = \begin{pmatrix}
0 & 0 & \ldots & 0 \\
A + BL & 0 & \ldots & 0 \\
(A + BL)^2 & A + BL & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & 0
\end{pmatrix}
= \begin{pmatrix}
C^L_0 \\
C^L_1 \\
C^L_2 \\
\vdots \\
C^L_k
\end{pmatrix}
\]

\[D^L = \begin{pmatrix}
0 & L(A + BL) & \ldots & 0 \\
0 & L(A + BL)^2 & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & 0
\end{pmatrix}
= \begin{pmatrix}
D^L_0 \\
D^L_1 \\
D^L_2 \\
\vdots \\
D^L_k
\end{pmatrix}
\]

B. Proof of Proposition 1

The first statement of the proposition follows straightforwardly from the derived estimate (14). To estimate the cost decrease (16) note that

\[J(X, u) = (\dot{x} + \Delta x)^T Q(\dot{x} + \Delta x) + (u + \Delta u)^T R(u + \Delta u) \leq J(\dot{x}, u) + \|2Q\dot{x}\| \|\Delta x\| + 2R\|\Delta u\| \]

so $J_\ast(x(k))$ also decreases for the inexact solution, i.e., it can be used as Lyapunov function, compare [4], [6].

C. Proof of Corollary 1

For $k = 0$ Proposition 1 guarantees that $x_0 \in X$ and $u_0 \in U$. Note that for $k \geq 1$, the feasibility follows from (17). Since the cost decreases (18), we can guarantee constraint satisfaction and convergence, compare [5], [16], [29]. If for $k = 0$, the locally optimal feedback is used, then $J(x_0, u_0) \leq c\|x(0)\|^2$ for some $c \in \mathbb{R}$ and due to a finite $c$ and the fact that $\mathcal{X}_0^\ast$ contains a neighborhood of the origin. This guarantees together with (18) stability, see [4], [6].

D. Proof of Proposition 2

The first three claims follow from Proposition 1 and the fact that $X \supseteq X$, $\hat{U} \supseteq \hat{U}$, compare (15), (20).

For the last item first note that for $i = 0, \ldots, N$

\[\bar{x}_{i+k}|k+1 = x_{i+k}|k + \Delta x_{i+k}|k, \quad \bar{u}_{i+k}|k \in \bar{U}_{i+k}|k \]

which guarantees that $\bar{x}_{i+k}|k+1 \in X_{i+1}$. Similarly, we can guarantee $\bar{x}_{i+k}|k+1 \in U_{i+1} - 1$ for $i = 0, \ldots, N - 1$. For $\bar{x}_{N+k+1}|k+1$, we get

\[\bar{x}_{N+k+1}|k+1 = (A + BK)\bar{x}_{N+k}|k + (A + BK)\Delta x_{N+k}|k, \quad \bar{u}_{N+k+1}|k \in U_{N+k+1} - 1 \]

so due to Assumption 3 $\bar{x}_{N+k+1}|k+1 \in \bar{X}_f$. Thus it remains to show that $\bar{x}_{N+k+1}|k+1 \in \bar{U}_N - 1$. We have

\[\bar{x}_{N+k+1}|k+1 = K\bar{x}_{N+k}|k + \Delta x_{N+k}|k, \quad \bar{u}_{N+k+1}|k \in U_{N+k} - 1 \]

so $\bar{x}_{N+k+1}|k+1$ is in $U \cap D_L^1 M \cup \ldots \cup D_L^k M$.

E. Proof of Corollary 2

Note that if $x(k) \in X_0^R$, then the above statement is a standard result, compare [4]. For $x(k) \notin X_0^R$, let $J^\ast(x(k)) = \min_{X, u} J(x, u)$ s.t. (3). We have

\[J(\bar{x}_k, \bar{u}_k) - J^\ast(x(k)) = \tilde{J}(\bar{x}_k, \bar{u}_k) - J(\tilde{x}_k, \tilde{u}_k) - J^\ast(x(k)) \leq \Delta J(M) + \epsilon \leq (1 - c)\gamma.

This guarantees that the cost increase at the first time instance $k = 0$ is finite:

\[J((x_0, u_0)) - J^\ast(x(0)) \leq (1 - c)\gamma.

Moreover, since $J^\ast(x^\ast(k+1)) - J^\ast(x(k)) \leq -x(k)^T Q x(k)$ for the optimal input $u^\ast(k)$ (and $x^\ast(k + 1) = Ax(k) + Bu^\ast(k)$), we can guarantee that $J^\ast(x^\ast(k+1)) - J^\ast(x(k)) \leq c\gamma$. So $J^\ast(x(k))$ also decreases for the inexact solution, i.e., it can be used as Lyapunov function, compare [4], [6].