2015 IEEE 54th Annual Conference on Decision and Control (CDC)
December 15-18, 2015. Osaka, Japan

A Parallel Dual Fast Gradient Method for MPC Applications*

Laura Ferranti[1], Tamás Keviczky[1]

Abstract— **We propose a parallel adaptive constraint-tightening approach to solve a linear model predictive control problem for discrete-time systems, based on inexact numerical optimization algorithms and operator splitting methods. The underlying algorithm first splits the original problem in as many independent subproblems as the length of the prediction horizon. Then, our algorithm computes a solution for these subproblems in parallel by exploiting auxiliary tightened subproblems in order to certify the control law in terms of suboptimality and recursive feasibility, along with closed-loop stability of the controlled system. Compared to prior approaches based on constraint tightening, our algorithm computes the tightening parameter for each subproblem to handle the propagation of errors introduced by the parallelization of the original problem. Our simulations show the computational benefits of the parallelization with positive impacts on performance and numerical conditioning when compared with a recent nonparallel adaptive tightening scheme.**

I. Introduction

Model Predictive Control (MPC) is a consolidated control technique that can efficiently handle constraints on the process to be controlled. Nevertheless, its application is not yet widespread in many domains where real-time computational constraints and requirements of certified solutions are of major concern, such as aerospace or automotive applications. There is a growing interest in both industry and academia for exploring parallel solutions to MPC problems ([1], [2], [3]), especially in light of the emerging many-core architectures, aiming to improve the computational efficiency of solving the underlying optimization problem.

Contribution. In this paper, we explore the use of parallelization techniques to efficiently solve a typical MPC problem for a linear discrete-time system, with a substantial computational speedup compared to nonparallel implementations. Our proposed algorithm combines the use of Alternating Direction Method of Multipliers (ADMMs [4], [5]) to handle the coupling constraints that arise from the dynamics of the system and inexact solvers (i.e., solvers that guarantee feasibility and optimality only *asymptotically* with the number of iterations), such as the Nesterov's Dual Fast Gradient (DFG) method [10]. In particular, the first step of the proposed algorithm is to split the original MPC problem over the length N of the prediction horizon into $N+1$ independent subproblems (*time-splitting* [3]) solved by $N+1$ parallel *workers* periodically exchanging information at predetermined synchronization points. Then, the second step is to solve these subproblems in parallel using an inexact solver and guarantee, at the same time, that the solution of the original MPC problem is *recursively feasible* and the system is *closed-loop stable*. The combination of parallelization and inexact solvers can result in infeasibility and closed-loop instability. We rely on an algorithm based on constraint tightening to overcome these issues. Loosely speaking, constraint-tightening algorithms solve an alternative problem in which the constraints have been tightened by a certain amount to compensate for the accuracy loss (and possible related infeasibility) introduced by the solver. We rely on an *adaptive* tightening strategy to select an appropriate amount of tightening for our algorithm. In particular, at the beginning of each sampling interval, our algorithm chooses the amount of tightening required for each subproblem in order to compensate for the error introduced by the time-splitting combined with the inexact solver.

Related work. The *time-splitting* technique has been proposed in [3]. In contrast to [3], we combine ADMM with inexact solvers and focus on the requirements for recursive feasibility and closed-loop stability of the original problem.

Other constraint-tightening schemes have been proposed in the literature (outside the parallel framework). For example, the authors in [7] propose an algorithm in which the amount of tightening is chosen offline to guarantee suboptimality and feasibility of the solution for all the initial states of the MPC problem. Solutions based on adaptive constraint tightening have been recently proposed in [8], where the tightening parameter is chosen adaptively. Compared to [8], our tightening update rule allows for a nonuniform amount of tightening (the tightening varies along the prediction horizon). Furthermore, thanks to the modular structure of our approach, the optimizer solves simpler problems of fixed dimension, which is independent from N. As a consequence, an increase of N does not affect the conditioning of the problem and the convergence of the solver. Hence, our approach leads to a performance improvement even when forcing full serialization of the parallel operations (i.e., *serialized* mode [9]).

Outline. In the following, Section II presents the initial problem formulation. Section III introduces the auxiliary subproblems and our proposed solver. Section IV describes our strategy to select the tightening of each subproblem to handle the parallelization error. Section V proposes an online update strategy of the tightening parameters that guarantees recursive feasibility, suboptimality, and closed-loop stability. Section VI presents numerical results using an academic example. Finally, Section VII concludes the paper.

*This research is supported by the European Union's Seventh Framework Programme FP7/2007-2013 under grant agreement n° AAT-2012-RTD-2314544 entitled "Reconfiguration of Control in Flight for Integral Global Upset Recovery (RECONFIGURE)".

[1]L. Ferranti and T. Keviczky are with the Delft Center for Systems and Control, Delft University of Technology, Delft, 2628 CD, The Netherlands, {l.ferranti,t.keviczky}@tudelft.nl

Notation. For $u \in \mathbb{R}^n$, $\|u\| = \sqrt{\langle u, u \rangle}$ is the Euclidean norm and $[u]_+$ is the projection onto non-negative orthant $\mathbb{R}^n_+$. Given a matrix A, $[A]_i$ denotes the i-th row of A and $[A]_{i,j}$ the entry (i, j) of A. Furthermore, $\mathbf{1}_n$ is the vector of ones in $\mathbb{R}^n$ and I_n the identity matrix in $\mathbb{R}^{n\times n}$. In addition, $\text{eig}_{\max}(A)$ and $\text{eig}_{\min}(A)$ denote the largest and the smallest (modulus) eigenvalues of the matrix A, respectively. $P \in \mathbb{S}^n_{>0}$ denotes that $P \in \mathbb{R}^{n\times n}$ is positive definite.

II. Problem Formulation

Consider the discrete-time linear system described below:

$$x(t+1) = Ax(t) + Bu(t) \qquad \forall t \geq 0, \tag{1}$$

where $x(t) \in \mathcal{X} \subseteq \mathbb{R}^n$ denotes the state of the system and $u(t) \in \mathcal{U} \subseteq \mathbb{R}^m$ denotes the control input. The sets $\mathcal{X}$ and $\mathcal{U}$ are simple proper convex sets (i.e., convex sets that contain the origin in their interior). Our goal is to steer $x(t)$ to the origin and satisfy the plant constraints. We use MPC to achieve these objectives. In this respect, consider the following finite-time optimal control problem:

$$\begin{aligned}
\mathcal{V}^*(x_{\text{init}}) = \min_{x,u}\ & \frac{1}{2}\sum_{t=0}^{N-1}(x_t^T Q x_t + u_t^T R u_t) + x_N^T P_N x_N && \text{(2a)}\\
\text{s.t.: } & x_{t+1} = Ax_t + Bu_t, \qquad t=0,\dots,N-1 && \text{(2b)}\\
& Cx_t + Du_t + g \leq 0, \qquad t=0,\dots,N-1 && \text{(2c)}\\
& x_0 = x_{\text{init}} && \text{(2d)}\\
& x_N \in \mathcal{X}_N. && \text{(2e)}
\end{aligned}$$

where x_t and u_t are more compact notations for $x(t)$ and $u(t)$, respectively. For $t = 0,\dots,N-1$ (N denotes the prediction horizon), the states and the control inputs are constrained in the polyhedral set described by (2c), where $C \in \mathbb{R}^{p_t\times n}$, $D \in \mathbb{R}^{p_t\times m}$, $g \in \mathbb{R}^{p_t}$. Note that (2c) can include constraints on the state only, i.e., $x_t \in \mathcal{X}$, and/or constraints on the control inputs only, i.e., $u_t \in \mathcal{U}$. In (2a), $Q \in \mathbb{S}^n_{\geq 0}$ and $R \in \mathbb{S}^m_{>0}$. Our problem formulation considers also a terminal cost $x_N^T P_N x_N$ associated with a polyhedral terminal set $\mathcal{X}_N := \{x \in \mathbb{R}^n | F_N x \leq f_N, F_N \in \mathbb{R}^{p_N\times n}, f_n \in \mathbb{R}^{p_N}\}$.

Through the remaining of the paper, we assume:

Assumption 1. *The pair (A, B) is stabilizable.*

Assumption 2. *Suppose Assumption 1 holds. Given the gain $K_f \in \mathbb{R}^{m\times n}$ obtained by the infinite-horizon linear quadratic regulator (IH-LQR)—characterized by the matrices A, B, Q, and R—the following holds:*

$$\forall x \in \mathcal{X}_N \ \Rightarrow\ \begin{cases} x \in \mathcal{X},\ K_f u \in \mathcal{U},\ \text{and} \\ (A + BK_f)x \in \mu\mathcal{X}_N,\ 0 \leq \mu < 1. \end{cases}$$

In addition, the terminal penalty $P_N \in \mathbb{S}^n_{>0}$ in the stage cost (2a) *is defined by the solution of the algebraic Riccati equation associated with the IH-LQR.*

In general, the MPC controller solves the optimization problem (2) every time new measurements are available from the plant and returns an optimal sequence of states and control inputs that minimizes the cost (2a). Let the optimal sequence be defined as follows:

$$\{\mathbf{x}, \mathbf{u}\} := \{x_0, \dots, x_N^*, u_0^*, \dots, u_{N-1}^*\}. \tag{3}$$

Only the first element of $\mathbf{u}$ is implemented in closed-loop, i.e., the control law obtained using the MPC controller is given by:

$$\kappa_{\text{MPC}}(x_{\text{init}}) = u_0^*, \tag{4}$$

and the closed-loop system is described by

$$x(t+1) = Ax(t) + B\kappa_{\text{MPC}}(x_{\text{init}}). \tag{5}$$

A. Parallelization

We aim to solve Problem (2) in parallel. Hence, we exploit a similar approach as the one proposed in [3]. Specifically, as in [3], Problem (2) is decomposed along the length of the prediction horizon N into $N+1$ independent subproblems to be solved by $N+1$ *parallel workers* Π_t ($t=0,\dots,N$). Each Π_t is allowed to communicate with its neighbours Π_{t-1} and Π_{t+1} at predefined synchronization points. The decomposition is possible thanks to the introduction of N auxiliary variables z_t ($t = 1,\dots,N$) used to break the dynamic coupling that arises from (2b). These z_t can be seen as the global variables of the algorithm. In particular, each z_t stores the local predicted state x_{t+1} of each subproblem and exchanges this stored information to guarantee consensus between neighboring subproblems, i.e., to ensure that the predicted state of the (t)-th subproblem, namely $x_{t+1}^{(t)}$, is equal to the current state of the $(t+1)$-st subproblem, namely $x_{t+1}^{(t+1)}$. Specifically, by introducing the *consensus constraints* $z_{t+1} = x_{t+1}^{(t)} = x_{t+1}^{(t+1)}$, defining $y_t := [x_t^{(t)T}\ u_t^{(t)T}]^T$, $H_1 := [I_n\ 0]$, $H_2 := [A\ B]$, $\rho > 0$ Problem (2) becomes:

$$\begin{aligned}
\min_{y,z}\ & \sum_{t=0}^{N}\mathbf{V}_t(y_t, z_t) && \text{(6a)}\\
\text{s.t.: } & G_t y_t + g_t \leq 0, \qquad t=0,\dots,N-1 && \text{(6b)}\\
& H_1 y_0 = x_{\text{init}}, && \text{(6c)}\\
& H_1 y_N \in \mathcal{X}_N, && \text{(6d)}\\
& z_{t+1} = H_2 y_t, \qquad t=0,\dots,N-1, && \text{(6e)}\\
& z_{t+1} = H_1 y_{t+1}, \ t=0,\dots,N-1, && \text{(6f)}
\end{aligned}$$

where, defining $\xi_t := [y_t^T z_t^T z_{t+1}^T]^T$:

- $\mathbf{V}_0(\xi_0) := \frac{1}{2}\xi_0^T\mathcal{Q}_0\xi_0 = \frac{1}{2}(y_0^T H_0 y_0 + \rho\|H_2 y_0 - z_1\|^2)$, where $H_0 := \text{diag}\{Q, R\}$.
- $\mathbf{V}_t(\xi_t) := \frac{1}{2}\xi_t^T\mathcal{Q}_t\xi_t = \frac{1}{2}(y_t^T H_0 y_t + \rho\|H_2 y_t - z_{t+1}\|^2 + \rho\|H_1 y_t - z_t\|^2)$, $t = 1,\dots,N-1$,.
- $\mathbf{V}_N(\xi_N) := \frac{1}{2}\xi_N^T\mathcal{Q}_N\xi_N = \frac{1}{2}(y_N^T H_N y_N + \rho\|H_1 y_N - z_N\|^2)$, where $H_N = \text{diag}\{P_N, 0_{m\times m}\}$.

Furthermore, G_t and g_t vary for each subproblem as follows:

- $G_t := [C\ D]$ and $g_t := g$, $t = 0,\dots,N-1$.
- $G_N := [F_N\ 0_{p_N\times m}]$ and $g_N := -f_N$.

Remark 1. Note that we introduced a quadratic penalty in the cost of the form $\rho/2(\|H_1 y_t - z_t\|^2 + \|H_2 y_t - z_{t+1}\|^2)$, according to the ADMM strategy [4]. This penalty has no impact on the cost of the original problem (2), if the consensus constraints are satisfied.

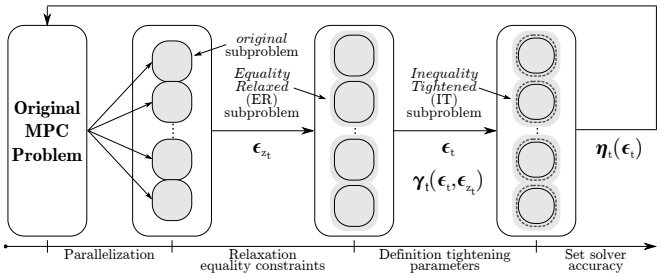

Fig. 1: Notation and terminology.

In the following, we introduce the subproblems that derive from Problem (6). Let v_{t+1} $(t=0,\dots,N-1)$ and w_t $(t=1,\dots,N)$ be the Lagrange multipliers associated with the equality constraints (6e) and (6f), respectively. Then, let the augmented Lagrangian with respect to the multipliers v_{t+1} and w_t be defined as follows:

$$\mathcal{L}_{v_{t+1},w_t}:=\mathbf{V}_t(\xi_t)+\rho[v_{t+1}^T(H_2y_t-z_{t+1})+w_t^T(H_1y_t-z_t)].$$

Hence, we obtain the following $N+1$ independent subproblems, called *original* subproblems, associated with the $N+1$ workers Π_t $(t=0,\dots,N)$:

$$\min_{y_t,z_t}\ \mathcal{L}_{v_{t+1},w_t}(y_t,z_t,z_{t+1})\ \text{s.t.:}\ G_ty_t+g_t\le 0. \qquad (7)$$

B. Overview of our proposed approach and terminology

Figure 1 summarizes the main steps that lead to a suboptimal solution of the aforementioned problem and introduces the keywords used in the remaining of the paper. Consider the MPC problem (2) and refer to this problem as the *original* MPC problem. The first step (*parallelization*, detailed in Section II-A) is to rewrite the original problem in $N+1$ independent subproblems (the *original* subproblems). The aim is to use a Dual Fast Gradient (DFG) method to solve these subproblems in order to certify—in terms of suboptimality and recursive primal feasibility, along with closed-loop stability—the MPC solution. The use of the inexact solver will eventually cause a violation of the consensus constraints (6e)-(6f) introduced to define the subproblems (7). Hence, the second step (*relaxation equality constraints* detailed in Section III-A) is introduced to relax the consensus constraints by a quantity[1] ϵ_{z_t}, preventing the occurrence of consensus-constraint violations. We refer to these subproblems as the *equality relaxed* (ER) subproblems. The set of inequality constraints of the ER subproblems includes the inequality constraints of the original subproblems (7) and the inequality constraints due to the relaxation of the consensus constraints (6e)-(6f). The third step (*definition tightening parameters* detailed in Sections III-C, IV, and V) is required to address the following remaining issues. First, the solution of each ER subproblem computed by the dual fast gradient method might violate the inequality constraints, due to the termination of the solver after a finite number of iterations. Hence, the constraints of the ER subproblems must be tightened by a quantity ϵ_t proportional to the desired level of suboptimality η_t chosen by the algorithm. Second, due to the relaxation of the consensus constraints, the *consolidated* prediction, i.e., the predicted evolution of the state computed (a posteriori) using the control sequence obtained by the independent subproblems, might deviate from the predicted *local* solution computed by the independent subproblems and, eventually, violate the inequality constraints of the original problem. Hence, an additional tightening (dependent on ϵ_{z_t}) must be introduced on the subset of inequality constraints of the ER subproblems that corresponds to the original inequality constraints. The proposed algorithm addresses the aforementioned issues by exploiting the *inequality tightened* (IT) subproblems. The IT subproblems differ from the ER subproblems in the definition of the feasibility region, which is tightened by a quantity $\gamma_t(\epsilon_t,\epsilon_{z_t})$ that depends on both ϵ_t and ϵ_{z_t}. The last step (*set solver accuracy*) selects a suboptimality level η_t for each subproblem, that guarantees a primal feasible and suboptimal solution for the original MPC problem within a fixed number of iterations.

[1]Note that the subscript t indicates that ϵ_{z_t} varies along the prediction horizon. This also holds for the later-defined ϵ_t, γ_t, and η_t.

III. Subproblem reformulation

In the following, we introduce the ER subproblems and the proposed algorithm to solve them using $N+1$ parallel workers. Furthermore, we introduce an initial formulation of the IT subproblems and derive conditions on the choice of the relaxation and tightening parameters to guarantee primal feasible and suboptimal solutions for of each subproblem.

A. Equality constraint relaxation

Our goal is to obtain a solution for Problem (6) by solving the independent subproblems (7) in parallel using inexact solvers, such as the Nesterov's DFG [10]. In order to use our proposed solver (introduced in Section III-B), which relies on first-order methods, we introduce a reformulation of Problem (6) to take into account that the constraints (6e) and (6f) cannot be satisfied at the equality due to the iterative nature of the proposed solver and its asymptotic convergence properties. In particular, introducing the relaxation parameters $\epsilon_{z_t},\epsilon_{z_{t+1}}>0$, for each subproblem $(t=0,\dots,N-1)$, the former equality constraints (6e)-(6f) are replaced by the following inequality constraints:

$$|H_1y_t-z_t|\le\epsilon_{z_t}\mathbf{1}_n \quad\Leftrightarrow\quad |x_t^{(t)}-z_t|\le\epsilon_{z_t}\mathbf{1}_n, \qquad (8a)$$

$$|H_2y_t-z_{t+1}|\le\epsilon_{z_{t+1}}\mathbf{1}_n \Leftrightarrow |x_{t+1}^{(t)}-z_{t+1}|\le\epsilon_{z_{t+1}}\mathbf{1}_n\,. \qquad (8b)$$

Thus, for each subproblem, we can realistically consider a feasible region defined by the following constraints:

$$\begin{bmatrix} G_t & 0 & 0\end{bmatrix}\xi_t+g_t \le 0, \qquad (9a)$$

$$\begin{bmatrix} H_1 & -I_n & 0\end{bmatrix}\xi_t-\epsilon_{z_t}\mathbf{1}_n \le 0, \qquad (9b)$$

$$\begin{bmatrix} -H_1 & I_n & 0\end{bmatrix}\xi_t-\epsilon_{z_t}\mathbf{1}_n \le 0, \qquad (9c)$$

$$\begin{bmatrix} H_2 & 0 & -I_n\end{bmatrix}\xi_t-\epsilon_{z_{t+1}}\mathbf{1}_n\le 0, \qquad (9d)$$

$$\begin{bmatrix} -H_2 & 0 & I_n\end{bmatrix}\xi_t-\epsilon_{z_{t+1}}\mathbf{1}_n\le 0, \qquad (9e)$$

or, in a more compact notation:

$$G_{\xi_t}\xi_t+g_{\xi_t}\le 0, \qquad (10)$$

where $G_{\xi_t}\in\mathbb{R}^{p_{\xi_t}\times(n+m)+2n}$ and $p_{\xi_t}:=p_t+4n$.

In the remaining of the paper, we consider the following *equality relaxed* (ER) subproblems:

$$\mathbf{V}_t^* = \min_{\xi_t} \mathbf{V}_t(\xi_t) \text{ s.t.: } G_{\xi_t}\xi_t + g_{\xi_t} \leq 0,\ t=0,\dots,N. \quad (11)$$

Hence, let $\mu_t := [\lambda_t^T\ w_t^{+T}\ w_t^{-T}\ v_{t+1}^{+T}\ v_{t+1}^{-T}]^T \in \mathbb{R}_+^{p_{\xi_t}}$ be the Lagrange multiplier associated with the new set of inequality constraints defined by (10), where λ_t, w_t^+, w_t^-, v_{t+1}^+, and v_{t+1}^- are the multipliers associated with the constraints (9a), (9b), (9c), (9d), and (9e), respectively. Then, to handle the complicated constraints (10), define, for each subproblem, the dual function $d_t(\xi_t, \mu_t)$ as follows:

$$d_t(\xi_t, \mu_t) = \min_{\xi_t} \mathcal{L}_{\mu_t}(\xi_t), \quad (12)$$

where $\mathcal{L}_{\mu_t}(\xi_t) := \mathbf{V}_t(\xi_t) + \mu_t^T \text{diag}\{I_{p_t}, \rho I_{4n}\}(G_{\xi_t}\xi_t + g_{\xi_t})$. We refer to (12) as the *inner subproblem*. Hence, we aim to solve the following dual subproblems (*outer subproblem*) in parallel to obtain a solution for Problem (6):

$$d_t^* = \max_{\mu_t} d_t(\xi_t, \mu_t),\ t=0,\dots,N. \quad (13)$$

Remark 2. The size of the ER subproblems remains unaffected if N increases. Intuitively, this *modularity* is an additional feature that can be exploited to preserve some favorable numerical properties of the problem (e.g., conditioning, Lipschitz constant, etc.) even when the algorithm is running in serialized mode.

B. Parallel dual fast gradient method

This section introduces Algorithm 1 that we use to solve the MPC problem (6) exploiting $N+1$ parallel workers Π_t. Note that, at this stage, we cannot yet ensure that the computed solution is feasible and suboptimal for Problem (2).

Algorithm 1 relies on Nesterov's DFG in which the inner problem is solved in an ADMM fashion, as explained below. Specifically, at each iteration of the algorithm[2], Π_t computes a minimizer ξ_t^k for $\mathcal{L}_{\mu_t}(\xi_t)$ (steps 1-4), i.e., the algorithm returns a solution for each inner subproblem (12). In particular, our algorithm, in compliance with the ADMM strategy, first minimizes $\mathcal{L}_{\mu_t}$ with respect to y_t in parallel for each subproblem (step 1). Then, using the information received by Π_{t+1}, i.e., the updated value of y_{t+1}^{k+1} (synchronization step 2), our algorithm computes—in parallel for each subproblem—the value of the global variable z_t according to the following rule (step 3):

$$z_t^{k+1} = \frac{1}{2}(H_1 y_t^{k+1} + H_2 y_{t-1}^{k+1} + v_t^+ + w_t^+ - v_t^- - w_t^-). \quad (14)$$

Note that this strategy allows to handle the coupling introduced by the 2-norm in the cost function of (11). Then, (synchronization step 4) Π_t receives (sends) the updated value of z_{t+1}^{k+1} (z_t^{k+1}) from Π_{t+1} (to Π_{t-1}), respectively. Finally, the worker Π_t computes the new values of the multipliers μ_t^{k+1} (steps 5-7). We compute offline (for each subproblem) the Lipschitz constant L_{μ_t} associated with $\nabla_{\mu_t} d_t(\xi_t, \mu_t)$ to perform the multipliers' update:

[2]Note that $[\xi_0^0]_{1:n} = x_0$ and $\mu_t^0 = 0_{p_{\xi_t}}$ to initialize Algorithm 1.

Algorithm 1 Parallel Dual Fast Gradient Method.

Given $\xi_t^0, \mu_t^0, Q_t, G_t, g_t, L_{\mu_t}$, and $\bar{k}_t$ for each Π_t $(t=0,\dots,N)$
while $k \leq \bar{k}_t$ **do**
1. Π_t computes $y_t^{k+1} = \text{argmin}_{y_t}\ \mathcal{L}_{\mu_t}(\xi_t^k, \mu_t^k)$.
2. Π_t receives y_{t+1}^{k+1} from Π_{t+1}, send y_t^{k+1} to Π_{t-1}.
3. Π_t updates z_t^{k+1} according to (14).
4. Π_t receives z_{t+1}^{k+1} from Π_{t+1}, sends z_t^{k+1} to Π_{t-1}.
5. Π_t computes
$$\hat{\mu}_t^{k+1} = \left[\mu_t^k + L_{\mu_t}^{-1}\nabla_{\mu_t}^T d_t(\xi_t^{k+1}, \mu_t)\right]_+.$$
6. Define: $a := \frac{k+1}{k+3} I$, $b_{\mu_t} := L_{\mu_t}^{-1}\frac{2}{(k+3)}$.
7. Π_t computes:
$$\mu_t^{k+1} = a\hat{\mu}_t^{k+1} + b_{\mu_t}\left[\sum_{s=0}^{k}\frac{s+1}{2}\nabla_{\mu_t}^T d_t(\xi_t^s, \mu_t)\right]_+.$$

end while

- $L_{\mu_0} = \text{diag}\left\{\frac{\|G_0\|_2^2}{\text{eig}_{\min}(\mathbf{Q}_0)} I_{p_0}, \frac{\|\rho\text{diag}\{H_2, -I\}\|_2^2}{\text{eig}_{\min}(\mathbf{Q}_0)} I_{2n}\right\}$.
- $L_{\mu_t} = \text{diag}\left\{\frac{\|G_t\|_2^2}{\text{eig}_{\min}(\mathbf{Q}_t)} I_{p_t}, \frac{\|\rho\text{diag}\{H_1, -I\}\|_2^2}{\text{eig}_{\min}(\mathbf{Q}_t)} I_{2n}, \frac{\|\rho\text{diag}\{H_2, -I\}\|_2^2}{\text{eig}_{\min}(\mathbf{Q}_t)} I_{2n}\right\}$, $(t = 1,\dots,N-1)$.
- $L_{\mu_N} = \text{diag}\left\{\frac{\|G_N\|_2^2}{\text{eig}_{\min}(\mathbf{Q}_N)} I_{p_N}, \frac{\|\rho\text{diag}\{H_1, -I\}\|_2^2}{\text{eig}_{\min}(\mathbf{Q}_N)} I_{2n}\right\}$.

Note that our update rule is different from the one proposed in [6], where ADMM is used in combination with Nesterov's fast gradient methods. At each iteration, the algorithm proposed in [6], first computes the exact minimizer y_t and then updates v_t and w_t. Our algorithm does not wait until the DFG returns a minimizer y_t to update the multipliers v_t and w_t, but starts updating their values along with the DFG iterations, encouraging the information exchange between neighboring subproblems. This algorithm is also different from the one proposed in [3]. In particular, the workers exchange the necessary pieces of information before the update of the Lagrange multipliers and none of the dual variables is exchanged between the neighboring workers. Furthermore, the information exchange between neighboring workers is unidirectional, i.e., Π_{t+1} sends the updated information to Π_t, but Π_t does not send any updated information to Π_{t+1}.

Using an argument similar to the one of Theorem 1 in [8], we can compute the primal feasibility violation and the level of suboptimality of the solution of each ER subproblem returned by Algorithm 1.

Theorem 1. *([8]) Let $\mathbf{V}_t(\xi_t)$ be strongly convex, the sequences $(\boldsymbol{\xi}_t^k,\ \hat{\mu}_t^k,\ \mu_t^k)$ be generated by Algorithm 1, and $\hat{\boldsymbol{\xi}}_t^k := \sum_{s=0}^k \frac{2(s+1)}{(k+1)(k+2)}\boldsymbol{\xi}^s$. Then, an estimate on the primal feasibility violation for the ER subproblem* (11) *is given by the following:*

$$\|[\nabla_{\mu_t}^T d_t(\hat{\boldsymbol{\xi}}_t^k)]_+\| \leq \frac{8R_t\max\{L_{\mu_t}\}}{(k+1)^2} =: \eta_t, \quad (15)$$

where $R_t := \|\mu_t^*\|$. Moreover, an estimate on primal suboptimality is given by the following:

$$0 \leq \mathbf{V}_t^* - \mathbf{V}_t(\hat{\boldsymbol{\xi}}_t) \leq R_t\eta_t. \quad (16)$$

Algorithm 1 terminates after a fixed number of iterations

that depends on η_t and R_t [8]:

$$\bar{k}_t := \left\lfloor \sqrt{8R_t\eta_t^{-1}\max\{L_{\mu_t}\}} \right\rfloor. \quad (17)$$

C. Tightening of the original inequality constraints

In order to guarantee the primal feasibility of each subproblem using Algorithm 1, we introduce $N+1$ auxiliary subproblems, namely the *inequality tightened* (IT) subproblems, which differ from the ER subproblems (11) in the definition of the feasible region. In particular, each IT subproblem can be defined as follows:

$$\mathbf{V}^*_{\epsilon_t} = \min_{\xi_t} \mathbf{V}_t(\xi_t) \text{ s.t.: } G_{\xi_t}\xi_t + g_{\xi_t} + \epsilon_t \mathbf{1}_{p_t+4n} \leq 0, \quad (18)$$

where $\epsilon_t \geq 0$ is the tightening parameter, which depends on the suboptimality level η_t that the proposed algorithm can reach within $\bar{k}_t$ iterations (17). According to [8], solving (18) using Algorithm 1 ensures, with a proper choice of ϵ_t, that the solution of (18) is primal feasible and suboptimal for subproblem (11).

To define an ϵ_t similar to the one introduced in [8], we must compute an upper bound for the optimal Lagrange multiplier, namely μ^*_{t,ϵ_t}, associated with the IT subproblem (18). We use an argument similar to the one of Lemma 1 in [12]. In particular, we compute the aforementioned upper bound for μ^*_{t,ϵ_t} according to the following lemma.

Lemma 1. Assume that there exists a Slater vector $\tilde{y}_t \in \mathbb{R}^{n+m}$ such that $G_t\tilde{y}_t + g_t < 0$. Then, there exists $\epsilon_t \geq 0$, $\epsilon_t < \min_{j=1,\dots,p_t}\{-(G_t\tilde{y}_t + g_t)_j\}$, $\epsilon_{z_t}, \epsilon_{z_{t+1}} > \epsilon_t$, such that the upper bound for μ^*_{t,ϵ_t} is given by

$$\|\mu^*_{t,\epsilon_t}\| \leq 2R_{d_t} := 2\frac{\mathbf{V}_t(\tilde{\xi}_t) - d_t(\tilde{\mu}_t)}{\min\limits_{j=1,\dots,p_t+2n}\{[\Gamma_t]_j\}}, \quad (19)$$

where $\Gamma_t := \big[[-(G_t\tilde{y}_t+g_t)^T - \epsilon_t \mathbf{1}_{p_t}^T][2\rho(\epsilon_{z_t}-\epsilon_t)\mathbf{1}_n^T][2\rho(\epsilon_{z_{t+1}}-\epsilon_t)\mathbf{1}_n^T]\big]^T \in \mathbb{R}^{p_t+2n}$, and $d_t(\tilde{\mu}_t)$ is the dual function for the original subproblem (13) evaluated at $\tilde{\mu}_t \in \mathbb{R}^{p_t+4n}$.

Proof: See Appendix I in [15]. ∎

Remark 3. Lemma 1 does not only provide an upper bound for $\|\mu^*_{t,\epsilon_t}\|$, but it also provides guidelines to select the values of ϵ_{z_t} and $\epsilon_{z_{t+1}}$ as a function of $\min_{j=1,\dots,p_t}\{-(G_t\tilde{y}_t + g_t)_j\}$, which only depends on the primal variable $\tilde{y}_t$. An alternative way to determine the relaxation parameters is to include ϵ_{z_t} and $\epsilon_{z_{t+1}}$ in the set of decision variables and penalize them in the cost function as it is usually done to handle soft constraints. This will, however, increase the number of decision variables in the problem formulation and it will have an impact on the original cost.

IV. Tightening improvement to guarantee primal feasible consolidated predictions

The previous section showed how to choose the tightening parameter ϵ_t of each IT subproblem to ensure that the t-th *local* solution, i.e., the solution computed by the t-th IT subproblem (18), is primal feasible for the t-th ER subproblem. This section provides guidelines to improve the choice of the tightening parameter of each IT subproblem (18) in order to guarantee the primal feasibility of the *consolidated* solution, i.e., the predictions obtained, starting from the initial state x_0, using the control sequence

$$\bar{\mathbf{u}}_\epsilon := \{\bar{u}^{(0)}_{0,\epsilon_0}, \dots, \bar{u}^{(N-1)}_{N-1,\epsilon_{N-1}}\}, \quad (20)$$

where the elements of $\bar{\mathbf{u}}_\epsilon$ are computed by the independent IT subproblems (18). In particular, when a new measurement is available from the plant, the subsystems (18) compute in parallel $(x_0, \bar{u}^{(0)}_{0,\epsilon_0}), \dots, (\bar{x}^{(N-1)}_{N-1,\epsilon_{N-1}}, \bar{u}^{(N-1)}_{N-1,\epsilon_{N-1}})$, and $x^{(N)}_{N,\epsilon_N}$, respectively. According to the results of the previous section, the pair $(\bar{x}^{(t)}_{t,\epsilon_t}, \bar{u}^{(t)}_{t,\epsilon_t})$ is primal feasible for the t-th subproblem (11), thanks to the introduction of the IT subproblems. Nevertheless, due to the relaxation introduced on the equality constraints (8a)-(8b), there is a bounded mismatch between $x^{(t)}_{t+1}$ and $x^{(t+1)}_{t+1}$ ($t=0,\dots,N-1$). Hence, starting from the initial state x_0, when the control sequence $\bar{\mathbf{u}}_\epsilon$ is applied to compute the consolidated state prediction

$$\bar{\mathbf{x}}_\epsilon := \{x_0, \bar{x}_{1,\epsilon_1} \dots, \bar{x}_{N,\epsilon_N}\}, \quad (21)$$

the feasibility of $\bar{\mathbf{x}}_\epsilon$ is no longer guaranteed. Note, however, that $\bar{\mathbf{u}}_\epsilon \in \mathcal{U} := \mathcal{U}_1 \times \dots \times \mathcal{U}_N$, i.e., $\bar{\mathbf{u}}_\epsilon$ is feasible. Hence, no additional tightening is needed on the input constraints.

In the following, Section IV-A defines an upper bound on the maximal feasibility violation of $\bar{\mathbf{x}}_\epsilon$. This feasibility violation is a consequence of the local relaxations of the equality constraints. Then, Section IV-B introduces sufficient conditions to ensure the primal feasibility of the *consolidated* prediction and provides guidelines for the choice of the tightening parameters for each IT subproblem.

A. Upper bound on the maximal feasibility violation of $\bar{\mathbf{x}}_\epsilon$

Let $\bar{\mathbf{u}}_\epsilon$ and $\bar{\mathbf{x}}_\epsilon$ be defined by (20) and (21), respectively. Moreover, from (8a) and (8b), the following holds:

$$|\bar{x}^{(t-1)}_{t,\epsilon_{t-1}} - \bar{x}^{(t)}_{t,\epsilon_t}| \leq 2\epsilon_{z_t}. \quad (22)$$

Our goal is to characterize how far the *consolidated* predicted state is from the *local* predicted state.

Lemma 2. Let the t-step-ahead *consolidated* prediction $\bar{x}_{t,\epsilon_t}$ be defined by (21) and assume that (22) holds. Then, there exists $\alpha_t \in \mathbb{R}, \alpha_t \geq 0$, such that the mismatch between $\bar{x}_{t,\epsilon_t}$ and the state of the t-th subproblem $\bar{x}^{(t)}_{t,\epsilon_t}$ is bounded, as follows:

$$|\bar{x}_{t,\epsilon_t} - \bar{x}^{(t)}_{t,\epsilon_t}| \leq \alpha_t. \quad (23)$$

Proof: See Appendix II in [15]. ∎

Remark 4. According to Lemma 2 a possible choice of α_t is the following:

$$\alpha_t := 2\sum_{j=0}^{t-1} \|A^j\|\epsilon_{z_{t-j}}. \quad (24)$$

B. Tightening parameter selection

According to Lemma 2, $\bar{x}_{t,\epsilon_t}$ differs from $\bar{x}^{(t)}_{t,\epsilon_t}$ by a quantity bounded by α_t. Thus, $\bar{x}_{t,\epsilon_t}$ might violate the constraints of the t-th subproblem (7) by as much as α_t, in the worst-case scenario. In particular, we must ensure that $C_t\bar{x}_{t,\epsilon_t} + D_t\bar{u}_{t,\epsilon_t} + g_t \leq 0$. Using the computed upper

bound (23), the following holds:

$$C_t\bar{x}^{(t)}_{t,\epsilon_t} + D_t\bar{u}^{(t)}_{t,\epsilon_t} + g_t + |C_t|\alpha_t\,\mathbf{1}_n + \epsilon_t\,\mathbf{1}_{p_t} \leq 0$$
$$\Updownarrow \text{(23)}$$
$$C_t\bar{x}_{t,\epsilon_t} + D_t\bar{u}^{(t)}_{t,\epsilon_t} + g_t + |C_t|\alpha_t\,\mathbf{1}_n + \epsilon_t\,\mathbf{1}_{p_t} \leq 0,$$

where $|C_t|$ indicates the absolute value of C_t. Recall that these mismatches are caused by the use of inexact solvers and that α_t depends on ϵ_{z_t}. In the following, we provide guidelines to improve the choice of ϵ_t for each subproblem. Furthermore, we provide a modified upper bound for the optimal Lagrange multiplier associated with the tightened subproblems (18), which considers the additional tightening introduced by α_t.

Lemma 3. Consider the following IT subproblems:

$$\mathbf{V}^*_{\gamma_t} = \min_{\xi_t}\ \mathbf{V}_t(\xi_t) \text{ s.t.: } G_{\xi_t}\xi_t + g_{\xi_t} + \gamma_t \leq 0, \tag{25}$$

for $t=0,\ldots,N$, where $\gamma_t := [(|C_t|\alpha_t\,\mathbf{1}_n + \epsilon_t\,\mathbf{1}_{p_t})^T\ \epsilon_t\,\mathbf{1}^T_{4n}]^T$. Consider the assumptions of Lemma 1 and the existence of α_t for all $t=1,\ldots,N$ according to Lemma 2. Then, for each subproblem, there exist $\epsilon_t \geq 0$, $\epsilon_{z_t}, \epsilon_{z_{t+1}} > \epsilon_t$ such that the upper bound for the optimal Lagrange multiplier associated with the IT subproblems (25) is described by

$$\|\mu^*_{t,\gamma_t}\| \leq 2\mathcal{R}_t := 2\frac{\mathbf{V}_t(\tilde{\xi}_t) - d_t(\tilde{\mu}_t)}{\min_{j=1,\ldots,p_t+2n}\{[\Gamma_{\alpha_t}]_j\}},\ t=0,\ldots,N,$$

$\Gamma_{\alpha_t} := \big[[-(G_t\tilde{y}_t + g_t)^T - (|C_t|\alpha_t\,\mathbf{1}_n)^T - \epsilon_t\mathbf{1}^T_{p_t}][2\rho(\epsilon_{z_t} - \epsilon_t)\mathbf{1}^T_n][2\rho(\epsilon_{z_{t+1}} - \epsilon_t)\mathbf{1}^T_n]\big]^T \in \mathbb{R}^{p_t+2n}$.

Proof: See Appendix III in [15]. ■

Remark 5. The choice of ϵ_t ($t=0,\ldots,N$) is not unique and depends on the choice of ϵ_{z_t} ($t=1,\ldots,N$). For example, given α_t in (24), a possible choice of ϵ_{z_t} ($t=1,\ldots,N$) is:

$$\epsilon_{z_t} \leq \min\left\{\frac{\epsilon_{z_N}}{\|A^{N-t}\|},\ldots,\frac{\epsilon_{z_{t+1}}}{\|A\|},\frac{\min_{j=1,\ldots,p_t}\{-(G_t\tilde{y}_t + g_t)_j\}}{1 + 2t\max_{j=1,\ldots,p_t}\big\{\sum_{i=1}^n |[C_t]_{j,i}|\big\}}\right\}. \tag{26}$$

Consequently, the tightening parameters are given by:

$$\epsilon_t \leq \frac{1}{2}\min\left\{\epsilon_{z_t}, \epsilon_{z_{t+1}}, \min_{j=1,\ldots,p_t}\{-(G_t\tilde{y}_t + g_t)_j\}\right\}, \tag{27}$$

for $t=0,\ldots,N$. This choice implies that first we select the relaxation parameters and then we *adapt* the tightening parameters on the original inequality constraints based on the choice of ϵ_{z_t} for all $t=1,\ldots,N$. An alternative is to fix ϵ_t for the inequality constraints and consequently compute ϵ_{z_t}. In general, the choice of the parameters strongly depends on the system-state matrix A in (1).

Remark 6. In the context of this work, Algorithm 2, described in the next section, adapts the above derived parameters at each problem instance. If we consider a fixed tightening scheme, such as the one proposed by [7], ϵ_t and ϵ_{z_t} can be computed offline (for all the initial states in the region of attraction).

In the following, we show that by using $\{\bar{\mathbf{x}}_\gamma, \bar{\mathbf{u}}_\gamma\}$—$\bar{\mathbf{u}}_\gamma$ is the control sequence obtained by solving the IT subproblems (25) and $\bar{\mathbf{x}}_\gamma$ is the corresponding consolidated prediction—the inequality constraints of the original MPC problem (2) are satisfied. Consequently, the predicted final state is in the terminal set of the original problem. If the desired level of suboptimality of Algorithm 1 is chosen as:

$$\eta_t := \epsilon_t/2, \tag{28}$$

then, according to Theorem 1, there exists $\bar{\xi}_{t,\gamma_t} := [\bar{y}^T_{t,\gamma_t}\ \bar{z}^T_{t,\gamma_t}\ \bar{z}^T_{t+1,\gamma_t}]^T$ such that $\|[\nabla^T_{\mu_t} d_{\gamma_t}(\bar{\xi}_{t,\gamma_t})]_+\| \leq \eta_t < \epsilon_t$. Using similar arguments as in [8], the following holds for $t=0,\ldots,N$:

$$\left[G_{\xi_t}\bar{\xi}_{t,\gamma_t} + g_{\xi_t} + \begin{bmatrix}|C_t|\alpha_t\,\mathbf{1}_n + \epsilon_t\,\mathbf{1}_{p_t}\\ \epsilon_t\,\mathbf{1}_{4n}\end{bmatrix}\right]_+ < \epsilon_t\,\mathbf{1}_{p_t+4n}.$$

Hence, for all $j=1,\ldots,p_t$, the following holds

$$\left[[C_t\bar{x}^{(t)}_{t,\gamma_t} + D_t\bar{u}^{(t)}_{t,\gamma_t} + g_t + |C_t|\alpha_t\,\mathbf{1}_n + \epsilon_t\,\mathbf{1}_{p_t}]_j\right]_+ \leq \epsilon_t.$$

Consequently, exploiting the upper bound (23), for all $j = 1,\ldots,p_t$, we have:

$$[C_t\bar{x}_{t,\gamma_t} + D_t\bar{u}_{t,\gamma_t} + g_t + |C_t|\alpha_t\,\mathbf{1}_n + \epsilon_t\,\mathbf{1}_{p_t}]_j \leq \epsilon_t$$

which leads to $C_t\bar{x}_{t,\gamma_t} + D_t\bar{u}_{t,\gamma_t} + g_t < 0 \quad \forall t=0,\ldots,N$, where $\bar{x}_{t,\gamma_t}$ is the t-step-ahead *consolidated* prediction computed using the solution to the IT subproblem (25) with tighening parameter γ_t.

In summary, this section showed that there exists a choice of the relaxation and tightening parameters that guarantee a feasible consolidated prediction with respect to the original problem (2).

V. Suboptimality, recursive feasibility, and closed-loop stability guarantees

In the following, we derive bounds for $\mathbf{V}_\gamma := \sum_{t=0}^N \mathbf{V}_t(\bar{\mathbf{x}}_\gamma, \bar{\mathbf{u}}_\gamma)$, i.e., the cost obtained using $\{\bar{\mathbf{x}}_\gamma, \bar{\mathbf{u}}_\gamma\}$, with respect to the optimal cost $\mathcal{V}^*$ of the original problem.

Theorem 2. *Assuming that there exist ϵ_t ($t=0,\ldots,N$) and ϵ_{z_t} ($t=1,\ldots,N$) selected according to Lemma 3, then the following holds:*

$$\mathcal{V}^* \leq \mathbf{V}_\gamma \leq \mathcal{V}^* + 2\sum_{t=0}^N \mathcal{R}_t\sqrt{p_t}\bar{\gamma}_t, \tag{29}$$

where $\bar{\gamma}_t := \epsilon_t + \max_{j=1,\ldots,p_t}\big\{\sum_{i=1}^n |[C_t]_{j,i}|\big\}\alpha_t$.

Proof: See Appendix IV in [15]. ■

Theorem 2 established the level of suboptimality of the consolidated prediction with respect to the original problem. In particular, the sequence $\{\bar{\mathbf{x}}_\gamma, \bar{\mathbf{u}}_\gamma\}$ is suboptimal for the original problem and satisfies the original inequality constraints (including those associated with $\mathcal{X}_N$).

Recall that for the update of $\mathcal{R}_t$, our algorithm requires a strictly feasible vector $\tilde{y}_t$ for (6b). Hence, every time new measurements are available from the plant, our algorithm must provide a strictly feasible solution (not necessarily

optimal) for the first p_t inequality constraints of each ER subproblem. The following lemma provides guidelines to compute $\tilde{y}_t$.

Lemma 4. Let $\bar{y}_\gamma$ be defined as $\bar{y}_\gamma := [\bar{y}_{0,\gamma_0}^T \dots \bar{y}_{N,\gamma_N}^T] = [(x_0^T \ \bar{u}_{0,\gamma_0}^T) \dots (\bar{x}_{N-1,\gamma_{N-1}}^T \ \bar{u}_{N-1,\gamma_{N-1}}^T) \ (\bar{x}_{N,\gamma_N}^T))]^T$. Then, a feasible $\tilde{y}^+$ at the next problem instance, is given by:

$$\tilde{y}^+ = \left[\bar{y}_{\gamma_{[2:N+1]}} \ [(A+BK_f)\bar{x}_{N,\gamma_N}]^T\right]^T \tag{30}$$

Proof: See Appendix V in [15]. ∎

We want to show that the cost decreases at each problem instance. Using a similar argument as in [13], under Assumption 2 on $\mathcal{X}_N$ and ensuring that $\bar{x}_{N,\gamma_N} \in \mathcal{X}_N$ (thanks to a proper choice of the tightening parameters, as the previous section showed), we can show that:

$$\sum_{t=0}^{N} \mathbf{V}_t(\tilde{y}_t^+) \leq \sum_{t=0}^{N} \mathbf{V}_t(\bar{y}_{t,\gamma_t}) - \mathbf{V}_0(y_{0,\gamma_0}) \ \forall y_{0,\gamma_0} \in \mathcal{Y}_{\text{attr}}, \tag{31}$$

where $\mathcal{Y}_{\text{attr}}$ is the region of attraction. Hence, from (29) and (31), the following holds:

$$\sum_{t=0}^{N} \mathbf{V}_t(\bar{y}_{t,\gamma_t}^+) \leq \sum_{t=0}^{N} \left[\mathbf{V}_t(\bar{y}_{t,\gamma_t}) + f(\bar{\gamma}_t^+, \mathcal{R}_t^+)\right] - \mathbf{V}_0(y_{0,\gamma_0}) \tag{32}$$

where $f(\epsilon_t^+, \mathcal{R}_t^+, \alpha_t^+) := (2\mathcal{R}_t^+\sqrt{p_t})\bar{\gamma}_t^+$, using $\bar{\gamma}_t^+, \mathcal{R}_t^+$ to represent the updated values of these parameters according to $\tilde{y}_\gamma^+$. The inequality above shows that the total cost decreases at each problem instance if $\mathcal{X}_N$ is defined according to Assumption 2 and if the N-step-ahead *consolidated* prediction lies in the terminal set. Asymptotic stability of our controller follows if $\mathbf{V}_0(y_{0,\gamma_0}) \geq \sum_{t=0}^{N} f(\bar{\gamma}_t^+, \mathcal{R}_t^+)$. Hence, we can modify the update of ϵ_t and ϵ_{z_t} to ensure that (32) is satisfied.

Remark 7. A possible choice of ϵ_{z_t} $(t = 1, \dots, N)$ to fulfill (32) is the following:

$$\epsilon_{z_t}^+ \leq \min\left\{\bar{\epsilon}_{z_t}, \epsilon_{z_t} \text{ in (26)}\right\}, \tag{33}$$

$$\bar{\epsilon}_{z_t} = \mathbf{V}_0\left[4N\mathcal{R}_t^+\sqrt{p_t}\left(1 + 2t \max_{j=1,\dots,p_t}\left\{\sum_{i=1}^{n} |[C_t]_{j,i}|\right\}\right)\right]^{-1}.$$

Consequently, ϵ_t can be selected according to (27) to preserve the definition of the upper bound on the optimal Lagrange multipliers given in Lemma 3.

Algorithm 2 summarizes the main steps needed to obtain a stabilizing control law when the original MPC problem is solved in parallel using inexact solvers. In particular, note that, if the measured state is in $\mathcal{X}_N$, from Assumption 2, the state and the control constraints are automatically satisfied without solving the MPC problem in parallel.

Remark 8. Steps 17-23 are the only nonparallel ones of the algorithm (Algorithm 1 is instead fully parallelizable). The main reason is in the adaptive nature of the algorithm (see also Remark 6). Algorithm 2 adapts ϵ_t and ϵ_{z_t} every time new measurements are available from the plant. A fully parallel Algorithm 2 is possible using a fixed tightening strategy, in which ϵ_t and ϵ_{z_t} can be computed offline.

Algorithm 2 MPC with adaptive parallel tightening scheme.

1: Given $A, B, \mathcal{X}, \mathcal{U}, \mathcal{X}_N, N$
2: Compute offline: K_f, P_f, F_N, f_N.
3: Measure: initial state x_{init} at time $t = 0$.
4: **for** $t = 0$ **to** N **do**
5: Compute offline: $G_{\xi_t}, g_{\xi_t}, \mathcal{Q}_t, \mathcal{W}_t, c_t$.
6: Compute: initial strictly feasible vector $\tilde{y}_t$.
7: Compute: initial tightening according to Lemma 3.
8: **end for**
9: **for** $t = 0$ **to** ∞ **do**
10: Measure: initial state x_{init}.
11: **if** $x_{\text{init}} \in \mathcal{X}_N$ **then**
12: Compute: $u = K_f x_{\text{init}}$.
13: **else**
14: Compute in parallel (Alg. 1): $\bar{\xi}_{0,\gamma_t}, \dots, \bar{\xi}_{N,\gamma_N}$ exploiting (25).
15: Compute: $u = \bar{u}_{\gamma_0}$.
16: Update: $\tilde{y} \leftarrow \tilde{y}^+$ according to (30).
17: **for** $t = 0$ **to** $N - 1$ **do**
18: Update: $\epsilon_{z_{N-t}} \leftarrow \epsilon_{z_{N-t}}^+$ according to Lemma 3.
19: **end for**
20: **for** $t = 1$ **to** N **do**
21: Update: $\epsilon_t \leftarrow \epsilon_t^+$ according to Lemma 3.
22: Update: $\gamma_t \leftarrow \gamma_t^+$ according to Lemma 3.
23: **end for**
24: **end if**
25: Implement u.
26: **end for**

VI. Evaluation

We evaluated Algorithm 2 using the LTI system described in [14]. The system (sampled at $T_s = 0.5$ s) is described by:

$$x(t+1) = Ax(t) + Bu(t), \ h(t) = Cx(t) + Du(t),$$

where $x(t) \in \mathcal{X} := \{x(t) \in \mathbb{R}^2 \,|\, |x_i(t)| \leq 4 (i = 1, 2), \forall t \geq 0\}$, $u(t) \in \mathcal{U} := \{u(t) \in \mathbb{R}^2 \,|\, |u_i(t)| \leq 1 \ (i = 1, 2), \forall t \geq 0\}$, $h(t) \in \mathcal{H} := \{h(t) \in \mathbb{R}^2 \,|\, |h_i(t)| \leq 1 \ (i = 1, 2), \forall t \geq 0\}$, and the quadruple (A, B, C, D) is given by:

$$A = \begin{bmatrix} 1.09 & 0.22 \\ 0.49 & 0.02 \end{bmatrix}, \ B = \begin{bmatrix} 1.22 & 0.88 \\ -0.78 & -0.34 \end{bmatrix}$$
$$C = \begin{bmatrix} 1.34 & -0.16 \\ -3.19 & -0.56 \end{bmatrix}, \ D = \begin{bmatrix} 1.60 & 1.01 \\ -0.68 & 0.77 \end{bmatrix}$$

The weighting matrices Q, R, and P_N in the cost (2a) and the IH-LQR gain K_f are selected according to [14]. We implemented our design in MATLAB (to tune the controller and test the initial design) and in C (to run a performance analysis). In particular, in MATLAB, we used the Parallel Computing Toolbox™ to assign the computation of Algorithm 1 to 8 parallel workers, given a prediction horizon $N = 7$. Furthermore, we relied on the MPT3 toolbox [11] to compute $\mathcal{X}_N$ and the optimal solution of Problem (2). Finally, we compared our design to [8].

We considered the following scenario. The initial state of the system is $x_0 = [-0.101 \ -3.7]^T$. The total number of complicated constraints (2c) for the original problem is 90. We used (26) and (27) to initialize ϵ_{z_t} and ϵ_t, respectively. To update them, we relied on (33) and (27). The selected x_0 caused u and h to saturate (12 active constraints). In this scenario, the state enters $\mathcal{X}_N$ in 3 steps.

Figure 2 shows the mismatch between the *local* prediction and the *consolidated* prediction for one problem instance. As Figure 2 depicts, the mismatch (for both states) is below the predicted upper bound α_t for all the $N + 1$ subproblems.

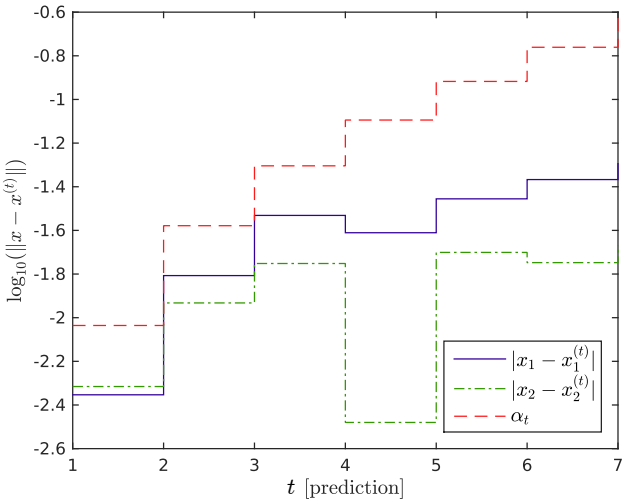

Fig. 2: Mismatch between *local* and *consolidated* predictions.

Table I compares the proposed technique to [8]. The table reports the upper bound $\bar{k}$ on the number of iterations needed to achieve a suboptimal solution for Problem (2) and the level of suboptimality η. The table lists only the first four subproblems, which are the most significant due to the presence of active constraints in these subproblems. The method in [8] and our new proposed parallel algorithm produce a comparable behavior, thanks to an appropriate selection of the tightening parameters. The parallel algorithm, however, is able to achieve similar results to those in [8] using a smaller number of iterations. The larger values of $\bar{k}$ for [8] are probably caused by the value of the Lipschitz constant and by the problem conditioning, which affect the convergence requiring a higher accuracy for the solver. In particular, in our proposed framework, the DFG is applied to simpler problems characterized, in the worst case scenario, by a Lipschitz constant $\max_{t=0,\dots,N}\{L_{\mu_t}\} = 196$ and by a condition number $\max_{t=0,\dots,N}\{\kappa_t\} = 104$. In [8], the DFG solves a larger problem characterized by a Lipschitz constant $L_s = 21994$ and by a condition number $\kappa_s = 7020$. Hence, the modularity of our approach has positive implications on important properties for the convergence of the solver.

Table I lists the time required by the optimizer to return a suboptimal solution for Problem (2). To measure the performance, we implemented both algorithms in C on a Linux-based OS. We noticed that given the small size of the problem, running Algorithm 1 in parallel did not result in significant speedups compared to our algorithm running in serialized mode [9]. Nevertheless, in both cases, we registered a speedup (230x) compared to [8]. The modularity of the proposed algorithm is beneficial even for problems of small size, such as the one considered in this section for the comparison with [8]. We expect the benefits to be even more pronounced when considering problems of larger size.

VII. Conclusions

We proposed an algorithm tailored to MPC that guarantees recursive feasibility and closed-loop stability, when the solution of the MPC problem is computed using inexact solvers in a parallel framework. In particular, our algorithm combines ADMM and DFG methods and relies on an adaptive constraint-tightening strategy to certify the MPC law.

Our numerical analysis shows performance improvements compared to state-of-the-art nonparallel techniques [8]. Furthermore, our study shows that, for small-size problems, even if the solver is implemented in a serialized mode, there is substantial performance improvement with respect to the state of the art. We expect further benefits from the parallelization when the size of the problem increases. A scalability analysis of the proposed algorithm on many-core architectures is part of our ongoing work.

TABLE I
PERFORMANCE ANALYSIS OF ALGORITHM 1 AND [8]. RESULTS SHOW THE MEDIAN OF 11 EXPERIMENTS.

Sample time	*Iterations (for subproblem)*					
	$\bar{k}$ ([8])	$\bar{k}_0$	$\bar{k}_1$	$\bar{k}_2$	$\bar{k}_3$	...
0	$11 \cdot 10^4$ (2.85 ms)	58931 (1.85 ms)	25	10	8	
1	$12{\cdot}10^6$ (302.84 ms)	18218 (0.57 ms)	3480	0	0	
2	$11{\cdot}10^6$ (267.56 ms)	2265 (0.07 ms)	0	0	0	
	Suboptimality Level (for subproblem, $\times 10^{-3}$)					
	η ([8])	η_0	η_1	η_2	η_3	...
0	3.48	1.15	1.15	1.90	2.25	
1	0.31	0.51	1.03	1.21	1.44	
2	0.15	0.62	1.25	1.47	1.74	

References

[1] S. D. Cairano, M. Brand, and S. A. Bortoff, "Projection-free parallel quadratic programming for linear model predictive control". *International Journal of Control*, vol. 86, no. 8, pp. 1367–1385, 2013.

[2] M. Kögel and R. Findeisen, "Parallel solution of model predictive control using the alternating direction multiplier method", Proc. of the *IFAC Conference on NMPC*, vol. 4, no. 1, pp. 369–374, 2012.

[3] G. Stathopoulos, T. Keviczky, and Y. Wang, "A hierarchical time-splitting approach for solving finite-time optimal control problems", Proc. of the *ECC*, pp. 3089–3094, 2013.

[4] S. Boyd et al. "Distributed optimization and statistical learning via the alternating direction method of multipliers". In *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[5] D. P. Bertsekas, "Constrained optimization and Lagrange multiplier methods". *Athena Scientific*, 1996.

[6] M. Kögel and R. Findeisen, "Fast predictive control of linear, time-invariant systems using an algorithm based on the fast gradient method and augmented Lagrange multipliers", Proc. of the *IEEE CCA*, pp. 780–785, 2011.

[7] M. Rubagotti, P. Patrinos and A. Bemporad, "Stabilizing Linear Model Predictive Control Under Inexact Numerical Optimization", In *IEEE Trans. on Automatic Control*, vol. 59, no. 6, pp. 1660–1666, 2014.

[8] I. Necoara, L. Ferranti, and T. Keviczky, "An adaptive tightening approach to linear model predictive control based on approximation algorithms for optimization", In *Optimal Control Applications and Methods*, DOI: 10.1002/oca.2121, 2015.

[9] M. Voss and R. Eigenmann. "Reducing parallel overheads through dynamic serialization", Proc. of the 13th IPDPS, pp. 88–92, 1999.

[10] Y. Nesterov, "Smooth minimization of non-smooth functions", *Mathematical Programming*, vol. 103, pp. 127–152, 2004.

[11] M. Herceg, M. Kvasnica, C.N. Jones, and M. Morari. "Multi-Parametric Toolbox 3.0", Proc. of the *ECC*, pp. 502–510, 2013.

[12] A. Nedić and A. Ozdaglar. "Approximate primal solutions and rate analysis for dual subgradient methods", SIAM Journal on Optimization, vol. 19, no. 4, pp. 1757–1780, 2009.

[13] J.B. Rawlings and D.Q. Mayne. "Model Predictive Control: Theory and Design", Nob Hill Publishing, 2009.

[14] M. Rubagotti, P. Patrinos, and A. Bemporad. "Stabilizing Embedded MPC with Computational Complexity Guarantees", Proc. of the *ECC*, pp. 3065–3070, 2013.

[15] L. Ferranti and T. Keviczky, "A Parallel Dual Fast Gradient Method for MPC Applications", arXiv:1503.06330 [math.OC], 2015.