Cooperative model predictive control: Current status and limitations

Kaushik Subramanian [*] James B. Rawlings[*†]

September 26, 2013

Abstract

In this paper, we summarize the current status of cooperative model predictive controllers for linear systems. We outline the suboptimal MPC theory that is used to guarantee closed loop properties of cooperative MPC and describe two methods to relax the centralized control problem to ensure that cooperative MPC iterates converge to the Pareto optimal solution of the relaxed optimization problem. Finally, we introduce a robust cooperative MPC algorithm based on tube-based MPC

1 Introduction

Over the past two decades, model predictive control (MPC) has emerged as one of the most commonly used control strategies in the process industry. MPC is an optimization based control algorithm in which a model of the process is used to predict the future evolution of the process under different inputs. The best among these input sequences are selected by solving an optimization problem that also includes the constraints of the system. However, only the first of these optimal inputs are injected to the plant, and the whole procedure (along with state estimation and data reconciliation) is repeated at the next sampling time, thereby incorporating feedback. MPC is widely used in many industries like Petrochemicals, fine chemicals, food, automotive and aerospace etc. [26]

Since only the first input is used and the optimization problem is solved at the next sampling time, it is important to provide guarantees that the control objectives are met in the closed-loop. Stability theory provides the theoretical support and the guidelines to design the open-loop optimization problem so that the desirable closed-loop properties (like asymptotic stability to the set-point) are guaranteed. Stability theory for MPC can either be (i) Optimal MPC, in which it is required that the online optimization problem be solved to optimality to ensure closed-loop stability or (ii) Suboptimal MPC, in which suboptimal

[*] Department of Chemical and Biological Engineering, University of Wisconsin-Madison, U.S.A.

[†] Corresponding author: rawlings@engr.wisc.edu

solutions to the online optimization problems that have certain properties are shown to be enough to ensure closed-loop stability [27, Chap 2]. We discuss Optimal and suboptimal MPC in section 2

Chemical plants are composed highly interconnected subsystems that exchange material, energy and information with one another. As model predictive controllers are being developed for more and more processes, the focus is shifting towards plant-wide control. Plant-wide control strategies must take into account the interactions between the many different subsystems that comprise the plant.

Centralized MPC is one way of achieving plant-wide control. In centralized model predictive control, the whole plant is considered as a single system and a monolithic controller is designed. Although offering good performance, the centralized controller can be hard to implement and maintain.

At the other end, in decentralized control, the plant is decomposed into many weakly interacting subsystems and the subsystems are controlled independently [30]. It has been shown that decentralized control can make the plant unstable [6]. It has also been shown that the so-called Noncooperative MPC, in which the subsystems share objective functions but not constraints can also destabilize the plant [39].

The middle ground is distributed control, in which we seek to coordinate multiple controllers [31]. A lot of research focus has been dedicated to distributed control strategies over the past decade. The preferred architecture for distributed MPC is a parallel optimization problem to solve the centralized control problem. In this architecture, each subsystem solves a part of the centralized control optimization problem. Based on the parallel optimization algorithm used, we can classify distributed MPC into three broad categories. Parallel optimization algorithms based on the dual of the centralized optimization problem form a class of distributed MPC algorithms. These algorithms use centralized MPC stability theory for ensuring closed-loop properties. However, they require that the subsystems converge to the optimal solution (with the intermediate iterates generally being infeasible). These algorithms may also require a coordinator. The second family of distributed MPC algorithms are the so-called negotiation based methods. In these algorithms, the subsystems find a feasible, but suboptimal input based on negotiations. The stability results for this family of algorithms is based on suboptimal MPC theory. The main drawback of these methods is that they do not provide any convergence properties. A brief survey of results belonging to both these classes of distributed MPC algorithms is provided in Section 4. In this paper, we review cooperative MPC. In cooperative MPC, we use the Gauss-Jacobi parallel optimization algorithm. We modify the centralized MPC optimization problem so that (i) every iterate is feasible and ensures cost-drop for the centralized optimization problem and (ii) the iterations converge to the centralized optimal solution. Therefore, this family of distributed MPC algorithms (i) requires no coordinator/ coordinating layer, (ii) ensure stability using suboptimal MPC theory and, (iii) ensure convergence to optimal solution on convergence.

In this paper, (i) we summarize the methods for cooperative MPC for linear systems and discuss the advantages and disadvantages of each method and (ii) propose a robust cooperative MPC algorithm based on tube-based robust MPC [27, Chapter 3].

In Section 2, we briefly overview suboptimal MPC theory which is the base upon which cooperative MPC is built. In Section 3 we present the cooperative MPC algorithm and two modifications to the centralized control problem to ensure performance guarantees, namely,(i) Using Kalman decomposition of the state space (3.1.1) and (ii) Using a relaxation of the terminal region (3.1.2). In Section 3.2, we present the tube-based cooperative MPC algorithm for robust cooperative control. Finally, in Section 4, we briefly outline related work in cooperative MPC before concluding in Section 5.

2 MPC Stability theory

As mentioned in the previous section, to study the model predictive controller, we need to focus, not only on the optimization problem solved at each sampling time, but also on the dynamics of the system with the injected move. Lyapunov theory is an convenient tool to study the stability and convergence properties of dynamic systems and MPC design procedures using Lyapunov theory has been widely studied. In Optimal MPC [27, Chapter 2], the on-line optimization problem needs to be solved to optimality to ensure closed-loop properties like stability and asymptotic convergence. In Suboptimal MPC [25, 33], the same closed-loop properties can be ensured without requiring the on-line problems to be solved to optimality. In this section, we briefly review the design procedure for ensuring closed-loop stability in suboptimal MPC.

2.1 Preliminaries

Consider a system comprising of M subsystems described by the following linear dynamics:

$$x_i^+ = A_i x_i + B_{ii} u_i + \sum_{\substack{l \in \{1,2,\ldots,M\} \\ l \neq i}} B_{il} u_l \tag{1}$$

$$x_i \in \mathbb{X}_i \qquad u_i \in \mathbb{U}_i \tag{2}$$

in which x_i, u_i are the states and inputs in subsystem i.

Denoting $x = (x_1, x_2, \ldots, x_M)$ and $u = (u_1, u_2, \ldots, u_M)$, the model can be written as

$$x^+ = Ax + Bu \tag{3}$$

The states of the system are described by x, while x^+ denotes the state at the next sampling instance. The states are constrained to lie in the set $\mathbb{X}$ while the inputs are constrained to lie in the set $\mathbb{U}$.

The MPC objective function is defined as:

$$V_N(x, \mathbf{u}) = \sum_{j=0}^{N-1} \ell(x(j), u(j)) + V_f(x(N)) \tag{4}$$

in which N is the control horizon, $\mathbf{u} = (u(0), u(1), \ldots, u(N-1))$ is the input vector and $x(j)$ is the short-hand notation for the state evolution under the input $\mathbf{u}$. That is,

$$x(j) = \phi(j; x, \mathbf{u}) = A^{j-1}x + \sum_{l=0}^{j-1} A^l Bu(j-1-l)$$

The cost function $V_N(x, \mathbf{u})$ consists of the stage cost $\ell(x(j), u(j))$ and the terminal cost $V_f(x(N))$. The costs are chosen to be positive definite quadratic cost:

$$\ell(x, u) = \sum_i 1/2(x_i' Q_i x_i + u_i' R_i u_i) \qquad V_f(x) = 1/2x'Px \tag{5}$$

in which $R_i > 0, Q_i > 0, P > 0$. Note that the the terminal cost $V_f(x)$ is not written as a sum of the terminal costs of the M subsystems. For simplicity, we refer to the matrix $\text{diag}(Q_1, Q_2, \ldots, Q_M)$ as Q. Similarly $\text{diag}(R_1, R_2, \ldots, R_M)$ is referred to as R.

The on-line optimization problem that is solved is:

$$\begin{aligned}
\mathbb{P}_N(x) : \min_{\mathbf{u}} & V_N(x, \mathbf{u}) & \\
& \text{s.t } x(j+1) = Ax(j) + Bu(j) & \forall j \in 0, 1, \ldots, N-1 \\
& x(j) \in \mathbb{X}, & \forall j \in 0, 1, \ldots, N-1 \\
& u(j) \in \mathbb{U}, & \forall j \in 0, 1, \ldots, N-1 \\
& x(0) = x & \\
& x(N) \in \mathbb{X}_f &
\end{aligned} \tag{6}$$

The set $\mathbb{X}_f$ is called as the terminal region.

We define the set $\mathbb{Z}$ as the feasible region of the problem $\mathbb{P}_N(x)$. That is,

$$\mathbb{Z} = \left\{(x, \mathbf{u}) \mid \phi(j; x, \mathbf{u}) \in \mathbb{X}, \phi(N; x, \mathbf{u}) \in \mathbb{X}_f, \mathbf{u} \in \mathbb{U}^N\right\} \tag{7}$$

The projection of the feasible space onto x is the set of feasible inputs for a given initial state x. This set is denoted by $\mathcal{U}_N(x)$ and is given by:

$$\mathcal{U}_N(x) = \{\mathbf{u} \mid (x, \mathbf{u}) \in \mathbb{Z}\}$$

The following assumptions are made on the system:

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

Assumption 2. *The cost functions $\ell(x, u)$ and $V_f(x)$ are positive definite.*

Assumption 3. *The set $\mathbb{X}_f$ and the costs $\ell(x,u), V_f(x)$ are chosen such that there exists a controller $u = \kappa_f(x)$ that satisfies:*

$$V_f(Ax + B\kappa_f(x)) - V_f(x) \leq -\ell(x, \kappa_f(x)) \qquad \forall x \in \mathbb{X}_f \tag{8}$$

$$Ax + B\kappa_f(x) \in \mathbb{X}_f, \kappa_f(x) \in \mathbb{U} \qquad \forall x \in \mathbb{X}_f \tag{9}$$

Assumption 4. *The set $\mathbb{U}$ is convex, closed and compact and contains the origin in its interior. The set $\mathbb{X}$ is convex, closed and contains the origin in its interior. The set $\mathbb{X}_f$ is convex, closed and compact and contains the origin in its interior.*

Remark 5. Assumption 1 ensures that a feedback linear controller $u = Kx$ and a cost function $V_f(x) = 1/2x'Px$ exist such that $P > 0$ and the closed-loop $x^+ = (A + BK)x$ is stable. For any K such that $(A + BK)$ is stable, we can find the terminal penalty by solving the Lyapunov equation (the existence of a solution to the Lyapunov equation is guaranteed by Assumption 2):

$$(A + BK)'P(A + BK) + (Q + K'RK) = P$$

The control invariant region in which the feedback controller $u = Kx$ is feasible is then the set $\mathbb{X}_f$ defined as:

$$\mathbb{X}_f := \left\{x \mid x^+ = (A + BK)x \in \mathbb{X}_f \subseteq \mathbb{X}, Kx \in \mathbb{U}\right\}$$

See Gilbert and Tan [9] for an algorithm to construct such sets for linear systems.

Remark 6. Assumption 3 also ensures that the following identity is true for any $\beta \geq 1$

$$\beta V_f(Ax + B\kappa_f(x)) + \ell(x, \kappa_f(x)) \leq \beta V_f(x)$$

2.2 Optimal MPC

The optimal solution to the optimization problem (6) is denoted by $\mathbf{u}^0(x)$ and the optimal objective value is given by $V_N^0(x)$. We define the optimal MPC law as $\kappa_0(x) = u^0(0; x)$ in which $u^0(0; x)$ is the first input in the optimal sequence $\mathbf{u}^0(x)$. The close-loop is hence $x^+ = Ax + B\kappa_0(x)$. The centralized MPC asymptotic (exponential) stabiity theorem [27, Thm 2.24(b), Chap. 2] is given by Theorem 7.

Theorem 7 (Optimal MPC stabilty)**.** *Let Assumptions 1–4 hold. Then the origin is exponentially stable with a region of attraction $\mathcal{X}_N$ for the system $x^+ = Ax + B\kappa_0(x)$. If $\mathcal{X}_N$ is unbounded, then the region of attraction is any sublevel set of $V_N^0(\cdot)$.*

The detailed proof for Theorem 7 is provided in Rawlings and Mayne [27, Chap. 2].

2.3 Suboptimal MPC

As the name suggests, in suboptimal MPC, we wish to inject *some* feasible input to the plant. There could be several reasons to inject suboptimal input sequences. For example, the sampling time might be too small to solve the problem to optimality. In order to ensure that despite injecting suboptimal inputs to the plant, we maintain the desirable closed-loop properties, we define the warm start and the successor input set as follows. We denote $\kappa_s(x)$ as the first input in the suboptimal input sequence.

Definition 8 (Warm Start)**.** Let $(x, \mathbf{u})$ be a state-input vector pair such that $\mathbf{u} \in \mathcal{U}_N(x)$ (that is $\mathbf{u}$ is feasible for $\mathbb{P}_N(x)$). Then the warm start for the successor initial state $x^+ = Ax + B\kappa_s(x)$ is defined as:

$$\tilde{\mathbf{u}} = (\mathbf{u}(1;x), \mathbf{u}(2;x), \ldots, \mathbf{u}(N;x), u_+)$$

in which $u_+ = \kappa_f(\phi(N;x,\mathbf{u}))$.

Definition 9 (Successor input set)**.** Consider $(x, \mathbf{u})$ such that $\mathbf{u}$ is feasible for $\mathbb{P}_N(x)$. For the successor state $x^+ = Ax + B\kappa_s(x)$, we define the set $G(x, \mathbf{u})$

$$\begin{aligned} G(x,\mathbf{u}) = \{\mathbf{u}^+ \mid \mathbf{u}^+ \in \mathcal{U}_N(x^+), V(x^+,\mathbf{u}^+) \leq V(x,\tilde{\mathbf{u}}), \\ V(x^+,\mathbf{u}^+) \leq V_f(x^+) \text{ if } x \in r\mathbb{B}\} \end{aligned}$$

in which $\tilde{\mathbf{u}}$ is the warm start given by Definition 8 and $r\mathbb{B}$ is a ball of radius $r > 0$. We choose r sufficiently small such that $r\mathbb{B}$ is a subset of the terminal region.

Additionally, we make the following assumptions on the cost function $V_N(x, \mathbf{u})$.

Assumption 10. *There exist positive constants a, a_1', a_2', a_f and r, such that the cost function $V_N(x, \mathbf{u})$ satisfies:*

$$\begin{aligned} \ell(x,u) &\geq a_1'|(x,u)|^a & (x,u) &\in \mathbb{X} \times \mathbb{U} \\ V_N(x,\mathbf{u}) &\leq a_2'|(x,\mathbf{u})|^a & (x,u) &\in \mathcal{B}_r \\ V_f(x) &\leq a_f|x|^a & x &\in \mathbb{X} \end{aligned}$$

in which $\mathcal{B}_r$ is the ball of radius r.

Note that it is easy to show that Assumption 10 is satisfied for linear systems and quadratic costs.

The suboptimal MPC optimization problem can be written as:

$$
\begin{aligned}
\mathbb{P}_N(x) : \min_{\mathbf{u}} \; & V_N(x, \mathbf{u}) \\
& \text{s.t } x(j+1) = Ax(j) + Bu(j) && \forall j \in 0, 1, \ldots, N-1 \\
& x(j) \in \mathbb{X}, && \forall j \in 0, 1, \ldots, N-1 \\
& u(j) \in \mathbb{U}, && \forall j \in 0, 1, \ldots, N-1 \\
& x(0) = x \\
& x(N) \in \mathbb{X}_f \\
& |\mathbf{u}|_i \leq d_i |x|, && \forall i \in 1, 2, \ldots, M, \\
& && \forall x \in \mathcal{B}_r
\end{aligned}
\tag{10}
$$

Theorem 11. *Let Assumptions 1 –4 and 10 hold. Choose optimization problem $\mathbb{P}_N(x)$ given by* (10) *and the appropriate terminal regions. For any x for which $\mathcal{U}_N(x)$ is not empty, choose $\mathbf{u} \in \mathcal{U}_N(x)$. Then, the origin of the closed-loop system*

$$
\begin{aligned}
x^+ &= Ax + B\kappa_s(x) \\
\mathbf{u}^+ &\in G(x, \mathbf{u})
\end{aligned}
$$

is asymptotically stable on (arbitrarily large) compact subsets of the feasible region $\mathcal{X}_N := \{x \mid \exists u \in \mathbf{U}^N,\ s.t\, \mathcal{U}_N(x) \neq \varnothing\}$.

The proof of Theorem 11 is presented in [25].

Observe that for the nominal system, the warm start is a member of the set $G(x, \mathbf{u})$ because of Assumption 3. Therefore, if we have a feasible $(x, \mathbf{u})$ pair, we can construct an asymptotically stable closed loop without any optimization. In-fact, if the optimization algorithm used to solve the on-line MPC problem generates feasible iterates which decrease the objective value, then we can stop the optimizations at any suboptimal solution and inject that solution to the plant and ensure asymptotic stability. Further, note that unlike the optimal MPC optimization problem, the on-line problem for suboptimal MPC has an extra constraint, namely, $V(x^+, \mathbf{u}^+) \leq V_f(x^+) \forall x \in \mathcal{B}_r$. This constraint can also be written as $|\mathbf{u}| \leq d|x|, d > 0, \forall x \in \mathcal{B}_r$. We can show that this constraint is equivalent to $|\mathbf{u}_i| \leq d_i|x|, x \in r\mathbb{B}, d_i > 0, i \in 1, 2, \ldots, M$ [36] in which $\mathbf{u}_i = (u_i(0), u_i(1), \ldots, u_i(N-1))$ and i denotes the input of subsystem i.

For future reference, we define the following set:

$$
\mathcal{U}_N^s(x; r) := \{u \mid u \in \mathcal{U}_N(x), |\mathbf{u}| \leq d|x| \text{ if } x \in \mathcal{B}_r\}
\tag{11}
$$

In the following section, we discuss the design of the on-line optimization problem for cooperative MPC algorithm that establishes stability using suboptimal MPC theory.

3 Cooperative MPC

In Cooperative MPC, we assume that each subsystem knows (i) the overall system model (3) and (ii) the overall system objective function (4). With the knowledge of the system-wide model and objective function, each subsystem then solves the centralized MPC problem (10). Each subsystem shares (i) its current decision variable $\mathbf{u}_i$ with all the other subsystems, (ii) optimizes the centralized problem over its decision variables having fixed the decision variables of the other sub-systems at the shared value, and (iii) makes the control move without requiring any coordinating layer. The cooperative MPC algorithm is presented in Algorithm 12.

For subsystem i, we denote all the other subsystems as $-i$. The following is the optimization problem for subsystem i . The input of all other subsystems is denoted by $\mathbf{u}_{-i}$. That is,

$$\mathbf{u}_{-i} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{i-1}, \mathbf{u}_{i+1}, \ldots, \mathbf{u}_M)$$

$$\begin{aligned}
\mathbb{P}_N^i(x, \mathbf{v}_{-i}) : \min_{\mathbf{u}_i} & V_N(x, \mathbf{u}) \\
& \text{s.t } x(j+1) = Ax(j) + Bu(j) && \forall j \in 0, 1, \ldots, N-1 \\
& x(j) \in \mathbb{X} && \forall j \in 0, 1, \ldots, N-1 \\
& u(j) \in \mathbb{U} && \forall j \in 0, 1, \ldots, N-1 \\
& u_{-i}(j) = v_{-i}(j), && \forall j \in 0, 1, \ldots, N-1 \\
& x(0) = x \\
& x(N) \in \mathbb{X}_f \\
& |\mathbf{u}_i| \leq d_i |x| && \forall x \in \mathcal{B}_r
\end{aligned} \tag{12}$$

Algorithm 12 (Cooperative MPC)**.**

In the inner loop in Algorithm 12, all the subsystems are solving the same optimization problem, but with the inputs of the other subsystem fixed. Each system, having solved the optimization problem, takes the convex combination of its optimal solution and the starting point. This algorithm is the implementation of the Jacobi parallel optimization routine [3, Section 3.3.5]. For convex optimization problem (convex objective and convex constraints), the Jacobi optimization generates feasible iterates that decrease the objective function value (when initialized from a feasible starting iterate). Hence, from the conclusions of the previous section, we can use suboptimal MPC theory to ensure stability of the cooperative MPC algorithm.

Before presenting the cooperative MPC stability theorem (which in different forms have been stated in Stewart et al. [34], Venkat [39], Stewart et al. [35], Subramanian et al. [36]), we briefly review the properties of Jacobi algorithm for convex optimization problem.

Jacobi algorithm

For the sake of simplicity, we consider only two subsystems in this section. Consider the following optimization problem (13), solved using Algorithm 13.

```
Data: Starting state $x(0)$, initial guess
      $(\tilde{\mathbf{u}}_1(0), \tilde{\mathbf{u}}_2(0), \ldots, \tilde{\mathbf{u}}_M(0)) \in \mathcal{U}_N^s(x(0); r)$, $\bar{p} \geq 1$ and $\omega_i \in (0,1)$ such
      that $\sum_{i=0}^{M} \omega_i = 1$
Result: Closed loop $(x(j), u(j)), j = \{1, 2, \ldots\}$
set $j \leftarrow 0$
while $j \geq 0$ do
    Set $p \leftarrow 0$, $x \leftarrow x(j)$
    Set $\mathbf{u}_i^{(p)} \leftarrow \tilde{\mathbf{u}}_i(j)$ for $i = 1, 2, \ldots, M$
    Broadcast current subsystem inputs $\tilde{\mathbf{u}}_i(j)$ to other subsystems
    while $p < \bar{p}$ do
        Solve $\min_{\mathbf{u}_i} V_N(x, \mathbf{u}) \quad \text{s.t. } \mathbf{u} \in \mathcal{U}_N^s(x; r); \mathbf{u}_{-i} = \mathbf{u}_{-i}^{(p)}$ to obtain $\mathbf{u}_i^0$
        for i in $1, 2, \ldots, M$
        Set $\mathbf{u}_i^{(p+1)} \leftarrow \omega_i \mathbf{u}_i^{(p)} + (1 - \omega_i)\mathbf{u}_i^0$ for i in $1, 2, \ldots, M$
        Set $p \leftarrow p + 1$
    end
    Set $\mathbf{u} \leftarrow (\mathbf{u}_1^{(p)}, \mathbf{u}_2^{(p)}, \ldots, \mathbf{u}_M^{(p)})$ and find $x(j+N) \leftarrow \phi(N; x(j), \mathbf{u})$
    Obtain $u_+ = (u_{1+}, u_{2+}, \ldots, u_{M+}) \leftarrow \kappa_f(x(j+N))$
    Obtain warm start $\tilde{\mathbf{u}}_i(j+1) = (\mathbf{u}_i^{(p)}(1), \mathbf{u}_i^{(p)}(2), \ldots, u_{i+})$ for
    $i = 1, 2, \ldots, M$.
    Set input as $u(j) = (\mathbf{u}_1^{(p)}(0), \mathbf{u}_2^{(p)}(0), \ldots, \mathbf{u}_M^{(p)}(0))$
    Evolve state from $x(j)$ to $x(j+1)$ under input $u(j)$
    Set $j \leftarrow j + 1$
end
```

$$\mathbb{J}:\min_{y_1,y_2} J(y_1,y_2) \qquad \text{s.t. } (y_1,y_2)\in\Omega \tag{13}$$

in which $J(\cdot)$ is a convex function and Ω is a convex, closed and compact set.

Algorithm 13 (Jacobi algorithm [3])**.**

```
Data: Starting guess $(y_1,y_2)\in\Omega$ and $\omega_i\in(0,1)$ such that $\sum_{i=1}^2\omega_i = 1$
Result: Sequence of feasible iterates $\{y^{(p)}\}$ and non-increasing objective
        function values $\{J(y^{(p)})\}$
Set $p \leftarrow 0$
while $p \geq 0$ do
    Set $y_1^{(p)} \leftarrow y_1$, $y_2^{(p)} \leftarrow y_2$
    Solve $\min_{y_1} J(y_1,y_2)$, s.t. $y_2 = y_2^{(p)}, (y_1,y_2)\in\Omega$ to obtain $y_1^0$.
    Solve $\min_{y_2} J(y_1,y_2)$, s.t. $y_1 = y_1^{(p)}, (y_1,y_2)\in\Omega$ to obtain $y_2^0$.
    Set $y_i^{(p+1)} \leftarrow \omega_i y_i^{(p)} + (1-\omega_i)y_i^0$ for i in $1,2$
    Set $p \leftarrow p+1$
end
```

Proposition 14 establishes that all iterates generated by the Jacobi algorithm for a convex optimization problem are feasible and decrease the objective function value.

Proposition 14. Let $J(y)$ be continuously differentiable and strongly convex function on the convex, closed and compact set Ω. Let Algorithm 13 be used to solve convex optimization problem (13) from an initial feasible point $y\in\Omega$. Then, (i) every iterate $y^{(p)}$ generated by the algorithm is feasible and , (ii) $J(y^{(p+1)}) \leq J(y^{(p)}), \forall p > 0$.

The proof is provided in Stewart et al. [34].

Proposition 14 does not establish any property about the convergence of the iterates generated by the Jacobi algorithm. In order to establish that the iterates converge to the optimal solution of the problem (13), we require a much stricter condition on the constraints. Proposition 15 lists the requirements on the optimization problem to ensure that the iterates generated by the Jacobi algorithm converge to the optimal solution.

Proposition 15. Let $J(y)$ be continuously differentiable and strongly convex. Let $\Omega = \Omega_1\times\Omega_2$, in which $y_1\in\Omega_1$ and $y_2\in\Omega_2$, with Ω_i convex, closed and compact. Then, as $p\to\infty$, the iterates $y^{(p)}$ converges to the y^o, in which y^o is the optimal solution to optimization problem (13)

This proof is provided in Stewart et al. [34]. Another proof is also provided in Bertsekas and Tsitsiklis [3, Prop 3.9] for Gauss-Seidel algorithm which is closely related to the Jacobi algorithm. Note that the important requirement in 15 is that the constraints be uncoupled.

We now present the exponential stability of the cooperative MPC algorithm.

Theorem 16. *Let Assumptions 1 – 4 and 10 hold. Choose $r > 0$ such that $\mathcal{B}_r \subset \mathbb{X}_f$. For any x for which $\mathcal{U}_N^s(x; r)$ is not empty, choose $\mathbf{u} \in \mathcal{U}_N^s(x; r)$. Then, the origin of the closed-loop system obtained by Algorithm 12 is exponentially stable. The region of attraction are (arbitrarily large) compact subsets of the feasible region*

$$\mathcal{X}_N := \left\{x \mid \exists u \in \mathbf{U}^N,\ s.t\, \mathcal{U}_N^s(x; r) \neq \varnothing\right\}$$

We present a sketch of the proof below. The detailed proof is availalbe in Stewart et al. [34].

We show that the closed-loop system obtained by Algorithm 12 is an implementation of suboptimal MPC and use Theorem 11 to prove exponential stability. The optimization problem solved by the Jacobi algorithm in cooperative MPC is the centralized suboptimal MPC optimization problem (10) We note that the optimization problem (10) has convex constraints and a strongly convex objective. By choice $Q, R, P > 0$, we know that the objective function is positive definite and hence strongly convex. Using Assumption 4, we can conclude that the constraints are convex, that is $\mathcal{U}_N(x)$ is convex. The set $\mathcal{U}_N^s(x; r)$ is the intersection of two convex sets, namely, the convex set $\mathcal{U}_N(x)$ and the level set $V_f(x, \mathbf{u}) \leq V_f(x) \forall x \in \mathcal{B}_r$. Therefore $\mathcal{U}_N^s(x; r)$ is convex as well. Hence, by Proposition 14, we know that if $(x, \tilde{\mathbf{u}})$ is feasible for (10) then (i) *all* the iterates generated by the inner loop in Algorithm 12 are feasible; implying $\mathbf{u}^{(p)} \in \mathcal{U}_N^s(x; r)$ and, (ii) the cost at iterate p is not greater than the cost achieved by $V_N(x, \tilde{\mathbf{u}})$; that is $V_N(x, \mathbf{u}^{(p)}) \leq V_N(x, \tilde{\mathbf{u}})$.

By choice of $\tilde{\mathbf{u}}(0)$, we know that $(x(0), \tilde{\mathbf{u}}(0))$ is feasible for (10). Therefore, $(x(0), \mathbf{u}^{(p)}(0)) \in \mathbb{Z}_N$ with $V_N(x(0), \mathbf{u}^{(p)}(0)) \leq V_N(x(0), \tilde{\mathbf{u}}(0))$. Since $\tilde{\mathbf{u}}(1)$ is the warm start constructed from $\mathbf{u}^{(p)}(0)$, we know that it is feasible and that $\tilde{\mathbf{u}}(1) \in G(x(0), \mathbf{u}^{(p)}(0))$. Therefore, by induction the closed-loop obtained by Algorithm 12 belongs to the family of closed-loop solutions for which we showed exponential stability in Theorem 11.

3.1 Performance guarantees

In showing exponential stability of cooperative MPC algorithm for linear systems, we used Proposition 14 to ensure that all the iterates generated by the cooperative MPC algorithm are feasible and provide cost-drop. Also, in establishing the stability of cooperative MPC, we did not modify the on-line optimization problem solved by the distributed controller. The distributed controller solves the centralized suboptimal MPC optimization problem. But, since the centralized suboptimal MPC optimization problem has coupled constraints (namely, the state constraints, the input constraint and the terminal constraint. The equality constraints can be eliminated fairly easily); the Jordan parallel optimization algorithm need not necessarily converge to the optimal solution. In order to establish performance guarantees (if we let the cooperative MPC algorithm to converge), we modify the on-line optimization problem to be solved such that it has no coupled constraints. By doing so, we ensure that the Jacobi algorithm converges to the optimal solution of the (re) designed on-line optimization problem. To “uncouple” the constraints, we make further assumptions on the system as listed below:

Assumption 17. *There are no state constraints. State constraints are enforced as soft-penalties by tuning the Q matrix in the stage cost.*

Assumption 18. *The input constraint space is uncoupled. That is, the input constraint set $\mathbb{U}$ is the Cartesian product of the input constraint sets of each subsystem.*

$$\mathbb{U} = \mathbb{U}_1 \times \mathbb{U}_2 \times \ldots \times \mathbb{U}_M$$

Under Assumption 17 and 18, the only coupled constraint is the terminal region constraint which needs to be enforced to guarantee stability and asymptotic convergence. In the next two sections, we briefly review two techniques to “uncouple” the terminal region constraint.

3.1.1 Sub-states

This relaxation was proposed by Stewart et al. [34] to solve the terminal equality constraint problem, that is, $\mathbb{X}_f = \{0\}$. To uncouple the terminal constraint $x(N) = 0$, we consider the “sub-state” x_{il}. The sub state x_{il} is the part of the state x_i in subsystem i that is *only* influenced by input u_l in subsystem l. The sub-state model is a non-minimal realization of the system (1). The dynamics for sub-state x_{il} is given by:

$$\tilde{x}_{il}^{+} = \tilde{A}_{il} x_{il} + \tilde{B}_{il} u_l \tag{14}$$

Defining $\hat{x}_i = \left[\{\tilde{x}_{il}\}, l = 1, 2, \ldots, M\right]$, each subsystem model is given by (1). The matrices $\hat{A}_i, \hat{B}_{il}$ are used to describe the dynamics in subsystem i.

$$\underbrace{\begin{bmatrix} \tilde{x}_{i1} \\ \tilde{x}_{i2} \\ \vdots \\ \tilde{x}_{iM} \end{bmatrix}^{+}}_{\hat{x}_i^{+}} = \underbrace{\begin{bmatrix} \tilde{A}_{i1} & & & \\ & \tilde{A}_{i2} & & \\ & & \ddots & \\ & & & \tilde{A}_{iM} \end{bmatrix}}_{\hat{A}_i} \underbrace{\begin{bmatrix} \tilde{x}_{i1} \\ \tilde{x}_{i2} \\ \vdots \\ \tilde{x}_{iM} \end{bmatrix}}_{\hat{x}_i} + \underbrace{\begin{bmatrix} \tilde{B}_{i1} \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\hat{B}_{i1}} u_1 + \underbrace{\begin{bmatrix} 0 \\ \tilde{B}_{i2} \\ \vdots \\ 0 \end{bmatrix}}_{\hat{B}_{i2}} u_2 + \ldots \underbrace{\begin{bmatrix} 0 \\ 0 \\ \vdots \\ \tilde{B}_{iM} \end{bmatrix}}_{\hat{B}_{iM}} u_M \tag{15}$$

$$x_i = \underbrace{\begin{bmatrix} \tilde{C}_{i1} & \tilde{C}_{i2} & \ldots & \tilde{C}_{iM} \end{bmatrix}}_{\hat{C}_i} \underbrace{\begin{bmatrix} \tilde{x}_{i1} \\ \tilde{x}_{i2} \\ \vdots \\ \tilde{x}_{iM} \end{bmatrix}}_{\hat{x}_i} \tag{16}$$

We assume that the centralized states x_i can be constructed from the substates $\hat{x}_i$ (see Assumption 19).

The subsystem stage cost $\ell_i(x_i, u_i) = 1/2(x_i' Q_i x_i + u_i R_i u_i)$ can now be written as

$$\ell_i(x_i, u_i) = \ell_i(\hat{x}_i, u_i) = 1/2(\hat{x}_i \hat{C}_i' Q_i \hat{C}_i \hat{x}_i + u_i R_i u_i)$$

The centralized stage cost is $\ell(x, u) = \sum_{i=1}^{M} \ell_i(\hat{x}_i, u_i)$. Since, we use a terminal constraint that all states are zero at the end of the horizon, we do not need a terminal penalty.

We define $\underline{x}_l$ as the sub-states that are affected by input l. That is $\underline{x}_l = \left[\{\tilde{x}_{il}\}\, i = \{1, 2, \ldots, M\}\right]$. Correspondingly, we define $\underline{A}_l, \underline{B}_l$ as follows:

$$\underbrace{\begin{bmatrix} \tilde{x}_{1l} \\ \tilde{x}_{2l} \\ \vdots \\ \tilde{x}_{Ml} \end{bmatrix}^{+}}_{\underline{x}_l^{+}} = \underbrace{\begin{bmatrix} \tilde{A}_{1l} & & & \\ & \tilde{A}_{2l} & & \\ & & \ddots & \\ & & & \tilde{A}_{Ml} \end{bmatrix}}_{\underline{A}_l} \underbrace{\begin{bmatrix} \tilde{x}_{1l} \\ \tilde{x}_{2l} \\ \vdots \\ \tilde{x}_{Ml} \end{bmatrix}}_{\underline{x}_l} + \underbrace{\begin{bmatrix} \tilde{B}_{1l} \\ \tilde{B}_{2l} \\ \vdots \\ \tilde{B}_{Ml} \end{bmatrix}}_{\underline{B}_l} u_l \tag{17}$$

The constraint $x_i(N) = 0$ can be equivalently written as $\tilde{x}_{il} = 0, l \in \{1, 2, \ldots, M\}$. Therefore, the centralized MPC problem can be written as:

$$\begin{aligned}
\mathbb{P}_N(x) : \min_{\mathbf{u}} & \; V_N(x, \mathbf{u}) \\
\text{s.t. } & \underline{x}_l(j+1) = \underline{A}x_l(j) + \underline{B}_l(j) u_l(j) && j = \{0, 1, 2, \ldots, N-1\}, \\
& && l = \{1, 2, \ldots, M\} \\
& x_i(j) = \hat{C}_i \hat{x}_i(j) && j = \{1, 2, \ldots, N\}, \\
& && i = \{1, 2, \ldots, M\} \\
& u_l(j) \in \mathbb{U} && j = \{0, 1, 2, \ldots, N-1\}, \\
& && l = \{1, 2, \ldots, M\} \\
& x_i(0) = x_i && i = \{1, 2, \ldots, M\} \\
& \underline{x}_l(N) = 0 && l = \{1, 2, \ldots, M\}
\end{aligned} \tag{18}$$

Optimization problem (18) does not have any coupled constraints as the $\underline{x}_l$ are the set of sub-states that depend only on one input. An alternate explanation is that the each subsystem input stabilizes all the sub-states that it influences in the inner optimizations of the cooperative MPC algorithm. Since there are no coupled constraints in the optimization problem, the Jacobi algorithm converges to the centralized solution.

Finally, we require that the non-minimal realization satisfies the following assumption to ensure that (i) the input $\mathbf{u}_i$ can be used to zero all the states that u_i influences (feasibility of (18)), (ii) all the sub-states can be reconstructed from the outputs (so that a state estimator can reconstruct x_{il} from measurements of x_i).

Assumption 19 (Subsystem stabilizability)**.**

- *The system* $(\underline{A}_i, \underline{B}_i)$ *is stabilizable.*

- *The system $(\hat{A}_i, \hat{C}_i)$ is detectable.*

Remark 20. The decomposition into the substate models (1) can be obtained from the Kalman decomposition of the original input/output y_i, u_i model [2, p.270]. The drawback however, is that not all centralized stabilizable models have a corresponding substate non-minimal realization that satisfies Assumption 19. One such example is the system of integrators (like supply chain models [36]).

3.1.2 Relaxing the terminal region

This relaxation was proposed by Rawlings et al. [28], and has been used for nonlinear suboptimal/ distributed / economic MPC in Stewart et al. [35], Pannocchia et al. [25], Amrit et al. [1] as well as linear MPC in Subramanian et al. [36].

In this relaxation, we (i) Modify the terminal region $\mathbb{X}_f$ to be the sub level set of the terminal cost given by (19) and (ii) Modify the objective function by magnifying the terminal cost using a parameter $\beta \geq 1$ as shown in (20)

$$\mathbb{X}_f := \{x \mid V_f(x) \leq a, a > 0\} \tag{19}$$

$$V_N^\beta(x, \mathbf{u}) = \sum_{j=0}^{N-1} \ell(x(i), u(i)) + \beta V_f(x(N)) \tag{20}$$

Note that we choose the parameter $a > 0$ such that the set set $\mathbb{X}_f$ given by (19) satisfies the Assumption 3.

The online optimization problem that is solved is given by:

$$\begin{aligned} \mathbb{P}_N(x) : \min_{\mathbf{u}} & V_N^\beta(x, \mathbf{u}) \\ & \text{s.t } x(j+1) = Ax(j) + Bu(j) && \forall j \in 0, 1, \ldots, N-1 \\ & u(j) \in \mathbb{U}, && \forall j \in 0, 1, \ldots, N-1 \\ & x(0) = x \\ & |\mathbf{u}|_i \leq d_i |x|, && \forall i \in 1, 2, \ldots, M, \\ & && \forall x \in \mathcal{B}_r \end{aligned} \tag{21}$$

Note that in contrast to the suboptimal MPC online problem (10), the optimization problem based on relaxing the terminal region (21) has a magnified terminal penalty (given by $V_N^\beta(x, \mathbf{u})$) but *no* terminal constraints. Therefore, the Jacobi parallel optimization algorithm on (21) will converge to the optimal solution.

However, to ensure closed-loop properties, we need to ensure that every feasible solution to (10) satisfies the terminal region constraint, that is (19). To do so, we design the feasible set of the problem (21) by appropriate choice of the parameter β. Recall that for suboptimal and optimal MPC, the feasible set was the set of $(x, \mathbf{u})$ pair for which the optimization problems (6) or (10) was feasible. In the case of this relaxation, we define the feasible set as the set of $(x, \mathbf{u})$ pairs for which the objective cost $V_N^\beta(x, \mathbf{u})$ is less than a parameter $\bar{V} > 0$. That is:

$$\mathbb{Z}_N^\beta = \left\{(x, \mathbf{u}) \mid V_N^\beta(x, \mathbf{u}) \leq \bar{V}, \mathbf{u} \in \mathbb{U}^N\right\} \tag{22}$$

Proposition 21 gives guidelines for the choice of the parameter β based on the parameters $a > 0, \bar{V} > 0$, such that any $(x, \mathbf{u})$ that lies in the set $\mathbb{Z}_N^\beta$ automatically satisfies the terminal region constraint $V_f(x(N; x, \mathbf{u}) \leq a$.

Proposition 21. Let the cost function be given by $V_N(x, \mathbf{u})$ (20). For $\bar{V} \geq a$, define $\bar{\beta} := \bar{V}/a$. Then, for any $\beta \geq \bar{\beta}$ and $(x, \mathbf{u}) \in \mathbb{Z}_N^\beta$ given by (22), we have that $\phi(N; x, \mathbf{u}) \in \mathbb{X}_f$ in which $\mathbb{X}_f$ is given by (19).

The proof is by contradiction and is presented in Subramanian et al. [36].

Since the problem (21) has no coupled constraints, we can ensure that the inner-loop of Algorithm 12 converges to the optimal solution of (21).

Remark 22. This relaxation based on magnifying the terminal penalty and designing the feasible set is applicable to any system which is stabilizable. In Pannocchia et al. [25], it was shown that the feasible set $\mathbb{Z}_N^\beta$ can be made as large as $\mathbb{Z}$(the feasible set for the original online MPC optimization problem (6)) by increasing $\bar{V}$. The drawback however is that the Hessian of the subsequent optimization problem can become ill-conditioned.

An important requirement for the cooperative MPC algorithm is that an initial feasible input sequence $\mathbf{u}^{(0)}$ be provided to initialize the system. During implementation, the warm start ensures that there is a feasible input sequence for the subsequent state. However, in the presence of uncertainty (model uncertainty) or additive noise, the warm start may not be feasible for the actual subsequent state. In such cases, we have to find a feasible $\mathbf{u} \in \mathbb{U}^N$ such that $V_N^\beta(x, \mathbf{u}) \leq \bar{V}$. This problem has coupled constraints, and to the best of our knowledge, no algorithm exists to find a feasible solution in parallel without requiring a coordinator. In the next section, we present an cooperative MPC algorithm based on robust MPC using tubes to overcome this drawback.

3.2 Robust cooperative MPC

In this section we build a cooperative MPC algorithm based on tube based MPC [15] that can be used even when the warm start becomes infeasible.

We consider the system (3) subject to bounded additive disturbance as follows:

$$x^+ = Ax + Bu + w \tag{23}$$

We make the following assumption on the disturbance:

Assumption 23 (Bounded disturbance)**.** *The additive disturbance w lies in a compact set $\mathbb{W}$ containing the origin in its interior.*

The dynamics without the additive disturbance is called the nominal system. The state of the nominal system is denoted by z and input by v. That is the nominal system dynamics is given by:

$$z^+ = Az + Bv \tag{24}$$

Denoting error $e = x - z$, we can write the error dynamics as (for the same input signal u).

$$e^+ = Ax + Bu + w - Az + Bu = Ae + w \tag{25}$$

Therefore, given the same initial condition and input sequence, the error at time k, $e(k)$ can be shown to lie in the following set:

$$e(k) \in S(i) := \sum_{j=0}^{k-1} A^j \mathbb{W} = \mathbb{W} \oplus A\mathbb{W} \oplus \ldots \oplus A^{k-1}\mathbb{W} \tag{26}$$

in which $A^j\mathbb{W}$ indicates set multiplication. That is,

$$A^j\mathbb{W} := \left\{A^j w \mid \forall w \in \mathbb{W}\right\}$$

The symbol $\oplus$ indicates set addition. That is,

$$\mathbb{W} \oplus A\mathbb{W} := \{w_1 + w_2 \mid w_1 \in \mathbb{W}, w_2 \in A\mathbb{W}\}$$

For stable A, it can be shown that the set $S(\infty)$ exists and is positive invariant for the system (25) [14].

In tube based robust MPC, (i) rolling horizon optimization based control that is designed to guarantee closed-loop asymptotic convergence is applied to the nominal system and (ii) based on the inputs to the nominal system, input to the process is obtained such that the error signal $e(k)$ converges to a known set. Since $x(k) = z(k) + e(k)$, we can then ensure that the state $x(k)$ as $k \to \infty$ lies in a bounded set that can be computed offline.

To do so, we use model predictive control for the nominal system to obtain $v(k) = \kappa_s(z(k))$. Then the input signal to the plant is based on the error signal at time k as $u(k) = v(k) + Ke(k)$. The gain K is chosen so that $A + BK$ is stable. For this choice of $(v(k), u(k))$, the error dynamics can be written as:

$$e^+ = x^+ - z^+ = Ax + Bv + BK(x - z) + w - Az - Bv = A_K e + w \tag{27}$$

By choice of K, the matrix $A_K := A + BK$ is stable and hence we can use theorems in [14] to ascertain that $e(k) \in \sum_{j=0}^{k-1} A_K^j \mathbb{W}$ and that the set $S_K(\infty) := \sum_{j=0}^{\infty} A_K^j \mathbb{W}$ exists and is positive invariant for (27).

The following observations help establish closed-loop properties of the robust control algorithm:

- The origin is asymptotically stable for the nominal system by choice of (sub)optimal MPC control law $v = \kappa_s(z)$.

- The error is designed to lie in the set $S_K(\infty)$
- The actual state $x(k) = z(k) + e(k)$ hence lies in the set $\{0\} \times S_K\infty$ as $k \to \infty$.

Finally, we have to ensure that the optimization based input $v = \kappa_s(z)$ is such that input that is injected to the plant after correcting for the disturbance is feasible. Therefore, the inputs v is constrained to lie in the set:

$$\mathbb{V}(k) := \mathbb{U} \ominus KS_K(k) \tag{28}$$

The tighter set follows from the fact that $e = (x - z) \in S_K(k)$.

The nominal MPC problem is defined as:

$$\begin{aligned}
\tilde{\mathbb{P}}_N(z) : \min_{\mathbf{u}} &V_N(z, \mathbf{v}) \\
&\text{s.t } z(j+1) = Az(j) + Bz(j) && \forall j \in 0, 1, \ldots, N-1 \\
&v(j) \in \mathbb{V}, && \forall j \in 0, 1, \ldots, N-1 \\
&z(0) = z \\
&z(N) \in \mathbb{Z}_f \\
&|\mathbf{v}_i| \leq d_i |z|, && z \in \mathcal{B}_r
\end{aligned} \tag{29}$$

in which $\mathbb{Z}_f$ is a terminal set that satisfies Assumption 3 and $V_N(z, \mathbf{v})$ is the cost function defined by (4). Observe that the optimization problem for robust MPC is the same as the optimization problem for suboptimal MPC (10). Therefore, following the results established in Section 3, Cooperative MPC for the nominal system can be employed to guarantee the same closed-loop properties as given by tube based MPC. However, the calculation to find the tightened input constraint set (28) could lead to input constraint set $\mathbb{V}$ that is not uncoupled. The performance guarantee (that the inner loop optimizations in the cooperative MPC algorithm converged to the optimal) can then be achieved by using one of the two relaxations proposed in the previous section and making an additional offline computation. This additional computation is that of an hyper box $\tilde{\mathbb{V}} \subseteq \mathbb{V}$ so that the input constraints to (29) are uncoupled.

As noted earlier, in robust MPC, the optimizations are performed based on the nominal state information, while the actual state could have drifted far from the nominal state because of the disturbances. We therefore use the modified version of the robust MPC algorithm presented in Rawlings and Mayne [27, P.234].

Algorithm 24 (Robust cooperative MPC)**.**

```
Data: Starting state $x(0)$, initial guess $(\tilde{\mathbf{u}}_1(0), \tilde{\mathbf{u}}_2(0), \ldots, \tilde{\mathbf{u}}_M(0))$ so that
      $V_N(x, \tilde{\mathbf{u}}) \leq \bar{V}$ $\bar{p} \geq 1$ and $\omega_i \in (0,1)$ such that $\sum_{i=0}^{M} \omega_i = 1$
Offline: Perform the following computations and share with every
subsystem; begin
    Compute $K$ so that $A + BK$ is stable
    Compute $S_K(\infty)$
    Compute $\mathbb{V} = \mathbb{U} \ominus KS_K(\infty)$
    Compute $\tilde{\mathbb{V}}_i$ so that $\tilde{\mathbb{V}}_1 \times \tilde{\mathbb{V}}_2 \times \ldots \times \tilde{\mathbb{V}}_M \subseteq \mathbb{V}$
end
Online: begin
    set $z(0) \leftarrow x(0); \tilde{\mathbf{v}}(0) \leftarrow \tilde{\mathbf{u}}(0)$
    set $k \leftarrow 0$
    while $k \geq 0$ do
        Set $p \leftarrow 0$
        Set $\mathbf{v}_i^{(p)} \leftarrow \tilde{\mathbf{v}}_i(k)$ for $i = 1, 2, \ldots, M$
        Broadcast current subsystem inputs $\tilde{\mathbf{v}}_i(k)$ to other subsystems
        while $p < \bar{p}$ do
            if $V_N(x(k), \tilde{\mathbf{v}}) \leq V_N(z(k), \tilde{\mathbf{v}}) \leq \bar{V}$ then
                Reset $z(k) \leftarrow x(k)$
            end
            Solve $\tilde{\mathbb{P}}_N^i(z(k), \mathbf{v}_{-i})$ to obtain $\mathbf{v}_i^0$ for i in $1, 2, \ldots, M$
            Set $\mathbf{v}_i^{(p+1)} \leftarrow \omega_i \mathbf{v}_i^{(p)} + (1 - \omega_i)\mathbf{v}_i^0$ for i in $1, 2, \ldots, M$
        end
        Set $\mathbf{v} \leftarrow (\mathbf{v}_1^{(p)}, \mathbf{v}_2^{(p)}, \ldots, \mathbf{v}_M^{(p)})$ and find $z(k+N) \leftarrow \phi(N; z(k), \mathbf{v})$
        Obtain $v_+ = (v_{1+}, v_{2+}, \ldots, v_{M+}) \leftarrow \kappa_f(z(k+N))$
        Obtain warm start $\tilde{\mathbf{v}}_i(k+1) = (\mathbf{v}_i^{(p)}(1), \mathbf{v}_i^{(p)}(2), \ldots, v_{i+})$ for
        $i = 1, 2, \ldots, M$.
        Set input as $v(k) = (\mathbf{v}_1^{(p)}(0), \mathbf{v}_2^{(p)}(0), \ldots, \mathbf{v}_M^{(p)}(0))$
        Evolve nominal state from $z(k)$ to $z(k+1)$ under input $v(k)$
        Set input $u(k) = v(k) + K(x(k) - z(k))$
        Evolve state from $x(k)$ to $x(k+1)$ under input $u(k)$
    end
end
```

The "If loop" in Algorithm 24 states that if the actual state is such that the warm start constructed from the nominal state is (i) feasible and, (ii) satisfies cost-drop, we "reset" the calculations to be based from the actual state. Otherwise, we continue to implement MPC for the nominal system. In this way, the convergence property of the closed-loop nominal state is guaranteed (since the cost-drop is satisfied all the time) and feedback is incorporated into the system.

4 Related Work

Cooperative MPC has evolved as a attractive architecture for distributed control because it solves the centralized control problem, and inherits the desirable closed-loop properties of centralized control. In Cooperative MPC, the centralized optimization problem is solved directly using parallel optimization architectures. Liu et al. [16] use an algorithm similar to cooperative MPC to solve the centralized control problem for non-linear systems. They obtain the warm start by using a closed form controller $\mathbf{u} = h(x)$ that satisfies all the criteria in the Lyapunov theorem to establish asymptotic stability. This controller is called the reference controller and the online controller is designed to improve the performance of the reference control. The authors propose algorithms based on the Jacobi algorithm (all subsystems optimize in parallel) and the Gauss-Seidel algorithm (subsystems optimize in sequence). The stability theorem is based on suboptimal MPC. However, since the centralized problem has coupled constraints, their algorithm does not provide performance guarantee. Cooperative MPC based on relaxing the terminal region has been proposed for non-linear systems by Stewart et al. [35]. The parallel optimization algorithm is an modification of the Jacobi algorithm. The modification is based on a sequential evaluation of objective functions to obtain a descent direction (without requiring a coordinator).

As mentioned earlier, an important requirement for cooperative MPC is that the subsystems share objective functions and models with each other. For situations in which it is not feasible for the subsystem to share objectives and models, Maestre et al. [18] propose a distributed MPC algorithm based on agent negotiation. In this algorithm, each subsystem solves its local optimization problem, but on the entire input sequence (including the inputs from other subsystems). In the second phase of the algorithms, the subsystems share their proposed solution with other subsystems. Finally, each subsystem calculates the objective value and constraint violation of the solutions from every other subsystem and broadcasts them. The subsystems then negotiate to obtain a feasible input sequence. By using warm start and algorithm design to accept only input sequences that decrease the cost, the algorithm establishes closed-loop properties by using suboptimal MPC. The drawback of the proposed architecture is that the agents have to solve larger optimization problems (because they have to optimize over all the inputs that affect their state).

Maestre et al. [19] use a game theory based distributed optimization algorithm to implement distributed MPC. Similar to Maestre et al. [18], the subsystems do not share models or objectives. In this algorithm, each subsystem broadcasts their current iterate to all the other subsystems. Upon receiving the iterates, the subsystems optimize for their input decisions keeping the other inputs fixed. Next, these optimized inputs are broadcast to the other subsystems. Next, each subsystem evaluates all possible combinations of previous iterate and current optimal solutions to come up with a list of (centralized input, local objective) pair that is broadcast. Each subsystem can now sum over the local objectives to find the centralized input with best cost-drop. While, stability is guaranteed by design of warm start and design of the terminal region, there are no convergence guarantees of the algorithm.

Müller et al. [22] decouple the terminal region constraint by providing a method that finds time varying local terminal regions that when satisfied by each subsystem, also satisfies the centralized terminal region. In their algorithm, the authors assume that subsystems do not share models and objectives. Hence, after performing local optimizations, input directions that do not reduce cost or input directions that are infeasible are discarded. Stability is established by using suboptimal MPC theory.

In the methods mentioned above, the parallel optimization algorithm has been designed to minimize a centralized cost when the subsystems do not know the centralized cost and/or constraints. These optimization algorithms are also designed satisfy the requirements of suboptimal MPC (feasible iterates, cost-drop at each iterate). However, the convergence properties and performance guarantees from these algorithms have not be studied.

Problems like multi-vehicle synchronization have a unique feature, in that, the dynamics of the different subsystems are uncoupled (x_i depends only on u_i). In such problems, the objective function is separable, and the only complicating constraint is the consensus point (which is similar to terminal equality constraint). In Johansson et al. [13], a primal decomposition is used to solve such control problems. The algorithm uses a coordinator to resolve the coupled constraint.

The distributed MPC algorithms mentioned above focus on designing a parallel optimization algorithm that satisfies the requirements of suboptimal MPC. As mentioned earlier, another family of methods for distributed MPC focus on solving the centralized optimization problem using a parallel optimization algorithm efficiently, and utilizing optimal MPC stability theory to provide guaranteed close-loop properties. These methods, typically focus on optimizing the dual of the online optimization problem using parallel solvers (with each solver representing a subsystem). The algorithms based on the dual decomposition converge to the centralized optimal solution for convex optimization problem with coupled constraints . The main disadvantage of these parallel optimization algorithms are that they (i) employ a coordinator that updates the Lagrange multiplier based on the subsystem solutions and, (ii) provide no guarantees on the feasibility of intermediate iterates. The second inequality implies that we have to wait till convergence before obtaining a feasible input.

There are many distributed algorithms using dual decomposition based on different parallel optimization algorithms used.

Subgradient methods are used in Cheng et al. [4],Ma et al. [17], Wakasa et al. [40], Marcos et al. [20].

Moroşan et al. [21] use Benders decomposition to solve a building control problem.

Scheu and Marquardt [32] augment the local subsystem objective function with the sensitivity of the objectives and constraints of other subsystems to obtain updates for the dual variables along with the primal variables. Thus, their algorithm does not require a coordinating layer.However, this method generates a feasible solution only upon convergence.

Giselsson et al. [11], Giselsson and Rantzer [10] propose a dual decomposition algorithm with a stopping criteria based on the objective value to ensure stability. They advocate the use of long prediction horizon along with results obtained in Grüne [12] to determine bounds on the value of the objective function so that stability can be guaranteed.

Doan et al. [7] modified the Han's algorithm which is a dual decomposition based algorithm for the special structure of the MPC problem. Although the method uses communication between directly connected subsystems, stability is guaranteed only upon convergence. Necoara et al. [23] use a smoothing technique to simplify the dual problem. With the smoothing technique, the coordinator problem for finding the Lagrange multiplier updates becomes easier. The algorithm also gives bounds on the number of iterations so that the optimal solution and constraint violation are within a pre-specified limit (ϵ approximation of the centralized problem). The main advantage is that the dual decomposition based on proximal center is order of magnitude faster than other sub-gradient based methods. Finally, Doan et al. [8], propose a primal feasible dual gradient approach, that generates a primal feasible solution that achieves cost-drop in a finite number of iterations based on an averaging scheme of the primal variables at each iteration.

Christofides et al. [5] is a recent review of different algorithms for distributed MPC. Necoara et al. [24] provides an excellent overview of the different optimization problems and parallel solution strategies that are seen in control and estimation.

Trodden and Richards [37, 38] propose a tube based robust distributed MPC algorithm. In their method, at each sampling time, only one subsystem performs optimization. The subsystem optimizes only over its decision variables, keeping all other subsystem decisions fixed from the previous iteration.. Richards and How [29] present a robust tube-based MPC for systems with decoupled dynamics. The coupling constraints are coupled output constraints. Their algorithm is based on the Gauss-Siedel iterations.

5 Conclusions

We propose a cooperative MPC algorithm in which each agent solves the centralized optimization problem subject to its own inputs. The algorithm uses a primal decomposition to solve the centralized problem using a Jacobi algorithm. Stability is ensured by designing appropriate terminal region and penalties and using suboptimal MPC theory. The advantages of using the cooperative MPC algorithm are (i) Guaranteed closed-loop properties, (ii) The optimization problems solved by the subsystems are of the same complexity if they subsystems were just optimizing its local control problem (in terms of the size of the optimization problem), (iii) No coordinators are required and (iv) There are no requirements on number of iterations to be made. In fact. the cooperative MPC is truly parallel in the sense that one or more subsystems can go offline and we can still find feasible inputs. The drawbacks of the cooperative MPC algorithm are that (i) Each subsystem has to share objective and constraints and (ii) Performance guarantee, that of converging to the centralized optimal solution, can only be guaranteed to a special class of problems that have uncoupled constraints. We believe that for many applications, sharing objectives and constraints would not be a bottleneck. For example, coordinating controllers inside a single plant. For the second drawback, we review two relaxations of the centralized online control problem so that the relaxations do not have coupled constraints. While the first relaxation based on sub-states is not applicable to all systems, the second relaxation based on magnifying the terminal penalty is applicable to all systems. The relaxations, however, add restrictions on the feasible set over which the centralized problem is defined. That is, compared to the centralized control problem, the relaxed centralized control problems have smaller region of attraction. Finally, we used techniques from MPC using tubes to propose a robust cooperative MPC controller that has both stability and performance guarantees.

References

[1] R. Amrit, J. B. Rawlings, and D. Angeli. Economic optimization using model predictive control with a terminal cost. *Annual Rev. Control*, 35: 178–186, 2011.

[2] P. J. Antsaklis and A. N. Michel. *Linear Systems*. McGraw-Hill, New York, 1997.

[3] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1989.

[4] R. Cheng, J. F. Forbes, and W. S. Yip. Price-driven coordination method for solving plant-wide MPC problems. *J. Proc. Cont.*, 17(5):429–438, 2007.

[5] P. Christofides, R. Scattolini, D. de la Peña, and J. Liu. Distributed model predictive control: A tutorial review and future research directions. *Comput. Chem. Eng.*, 2012.

[6] H. Cui and E. W. Jacobsen. Performance limitations in decentralized control. *J. Proc. Cont.*, 12:485–494, 2002.

[7] D. Doan, T. Keviczky, I. Necoara, M. Diehl, and B. De Schutter. A distributed version of han's method for dmpc using local communications only. *Contr. Eng. and App. Info.*, 11(3):6–15, 2009.

[8] M. Doan, T. Keviczky, and B. De Schutter. A dual decomposition-based optimization method with guaranteed primal feasibility for hierarchical mpc problems. In *18th IFAC World Congress*, 2011.

[9] E. G. Gilbert and K. T. Tan. Linear systems with state and control constraints: The theory and application of maximal output admissible sets. *IEEE Trans. Auto. Cont.*, 36(9):1008–1020, September 1991.

[10] P. Giselsson and A. Rantzer. Distributed model predictive control with suboptimality and stability guarantees. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 7272–7277. IEEE, 2010.

[11] P. Giselsson, M. Doan, T. Keviczky, B. De Schutter, and A. Rantzer. Accelerated gradient methods and dual decomposition in distributed model predictive control. *Automatica*, 2012.

[12] L. Grüne. Analysis and design of unconstrained nonlinear mpc schemes for finite and infinite dimensional systems. *SIAM J. Cont. Opt.*, 48(2): 1206–1228, 2009.

[13] B. Johansson, A. Speranzon, M. Johansson, and K. Johansson. Distributed model predictive consensus. In *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems*, pages 2438–2444, 2006.

[14] I. Kolmanovsky and E. G. Gilbert. Theory and computation of disturbance invariant sets for discrete-time linear systems. *Math. Probl. Eng.*, 4(4):317–367, 1998.

[15] W. Langson, I. Chryssochoos, S. V. Raković, and D. Q. Mayne. Robust model predictive control using tubes. *Automatica*, 40:125–133, January 2004.

[16] J. Liu, X. Chen, D. Muñoz de la Peña, and P. D. Christofides. Sequential and iterative architectures for distributed model predictive control of nonlinear process systems. *AIChE J.*, 56(5):2137–2149, 2010.

[17] Y. Ma, G. Anderson, and F. Borrelli. A distributed predictive control approach to building temperature regulation. In *American Control Conference (ACC), 2011*, pages 2089–2094. IEEE, 2011.

[18] J. Maestre, D. Muñoz de la Peña, E. Camacho, and T. Alamo. Distributed model predictive control based on agent negotiation. *J. Proc. Cont.*, 21(5): 685–697, 2011.

[19] J. M. Maestre, D. Muñoz de la Peña, and E. F. Camacho. Distributed model predictive control based on a cooperative game. *Optimal Cont. Appl. Meth.*, 32(2):153–176, 2011.

[20] N. I. Marcos, J. F. Forbes, and M. Guay. Coordination of distributed model predictive controllers for constrained dynamic processes. In *ADCHEM 2009, International Symposium on Advanced Control of Chemical Processes*, Istanbul, Turkey, July 12-15, 2009.

[21] P. Moroşan, R. Bourdais, D. Dumur, and J. Buisson. A distributed mpc strategy based on benders decomposition applied to multi-source multizone temperature regulation. *J. Proc. Cont.*, 21(5):729–737, 2011.

[22] M. Müller, M. Reble, and F. Allgöwer. Cooperative control of dynamically decoupled systems via distributed model predictive control. *Int. J. Robust and Nonlinear Control*, 2012.

[23] I. Necoara, D. Doan, and J. Suykens. Application of the proximal center decomposition method to distributed model predictive control. In *Proceedings of the IEEE Conference on Decision and Control*, Cancun, Mexico, December 9-11 2008.

[24] I. Necoara, V. Nedelcu, and I. Dumitrache. Parallel and distributed optimization methods for estimation and control in networks. *J. Proc. Cont.*, 21(5):756–766, 2011.

[25] G. Pannocchia, J. B. Rawlings, and S. J. Wright. Conditions under which suboptimal nonlinear MPC is inherently robust. *Sys. Cont. Let.*, 60:747–755, 2011.

[26] S. J. Qin and T. A. Badgwell. A survey of industrial model predictive control technology. *Control Eng. Prac.*, 11(7):733–764, 2003.

[27] J. B. Rawlings and D. Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison, WI, 2009. 576 pages, ISBN 978-0-9759377-0-9.

[28] J. B. Rawlings, B. T. Stewart, S. J. Wright, and D. Q. Mayne. Suboptimal MPC: On replacing the terminal contraint by a terminal cost, 2010. Internal communication.

[29] A. Richards and J. How. A decentralized algorithm for robust constrained model predictive control. In *Proceedings of the American Control Conference*, Boston, Massachusetts, June 2004.

[30] N. R. Sandell Jr., P. Varaiya, M. Athans, and M. Safonov. Survey of decentralized control methods for large scale systems. *IEEE Trans. Auto. Cont.*, 23(2):108–128, 1978.

[31] R. Scattolini. Architectures for distributed and hierarchical model predictive control - a review. *J. Proc. Cont.*, 19(5):723–731, May 2009. ISSN 0959-1524.

[32] H. Scheu and W. Marquardt. Sensitivity-based coordination in distributed model predictive control. *J. Proc. Cont.*, 21(5):715–728, 2011.

[33] P. O. M. Scokaert, D. Q. Mayne, and J. B. Rawlings. Suboptimal model predictive control (feasibility implies stability). *IEEE Trans. Auto. Cont.*, 44(3):648–654, March 1999.

[34] B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, and G. Pannocchia. Cooperative distributed model predictive control. *Sys. Cont. Let.*, 59: 460–469, 2010.

[35] B. T. Stewart, S. J. Wright, and J. B. Rawlings. Cooperative distributed model predictive control for nonlinear systems. *J. Proc. Cont.*, 21:698–704, 2011.

[36] K. Subramanian, J. B. Rawlings, and C. T. Maravelias. Integration of control theory and scheduling methods for supply chain management. Proceedings of Foundations of Computer-AidedProcess Operations (FOCAPO) 2012 and Chemical Process Control (CPC) VIII, Savannah, GA, 2012.

[37] P. Trodden and A. Richards. Robust distributed model predictive control using tubes. In *American Control Conference, 2006*, pages 6–pp. IEEE, 2006.

[38] P. Trodden and A. Richards. Robust distributed model predictive control with cooperation. In *Proceedings of the European Control Conference, 2007*, pages 2172–2178, 2007.

[39] A. N. Venkat. *Distributed Model Predictive Control: Theory and Applications*. PhD thesis, University of Wisconsin–Madison, October 2006. URL `http://jbrwww.che.wisc.edu/theses/venkat.pdf`.

[40] Y. Wakasa, M. Arakawa, K. Tanaka, and T. Akashi. Decentralized model predictive control via dual decomposition. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 381–386. IEEE, 2008.