Preprints of the 10th IFAC International Symposium on Dynamics and Control of Process Systems
The International Federation of Automatic Control
December 18-20, 2013. Mumbai, India

Set-point tracking using distributed MPC

Markus Kögel * **Rolf Findeisen** *

* *Institute for Automation Engineering, Otto-von-Guericke-University Magdeburg, Germany. e-mail: {markus.koegel, rolf.findeisen}@ovgu.de.*

Abstract: We consider the output tracking of a set-point using cooperative distributed model predictive control. We propose a framework to avoid the loss of feasibility, guarantee stability, constraint satisfaction as well as convergence to admissible set-points. To enable a distributed implementation of the model predictive control law we utilize a cyclic varying horizon length. A simulation example illustrates the approach and its applicability for interconnected systems.

Keywords: Model predictive control, Distributed Model predictive control, Set-point tracking.

1. INTRODUCTION

Model predictive control (MPC) is frequently used to efficiently control systems subject to constraints, see Maciejowski (2002); Qin and Badgwell (2003). In MPC an optimal control problem is solved at each sampling instance and the first part of the resulting input is applied until the next sampling instance. Solving this optimization problem can be challenging, in particular for large-scale systems. Therefore, there is an increasing trend to derive distributed, predictive control schemes.

There are by now various MPC approaches for the control of coupled systems, see Scattolini (2009). One straightforward approach is to use a centralized MPC controller, which centrally controls all systems, c.f. top of Figure 1. However this often results in a large optimization problem and requires the communication with and calculation in one central controller, which might be prone to failures and errors.

Further, it is possible to control each subsystem by a single MPC controller. If no communication between the controllers is present, then this is frequently referred as decentralized MPC, see e.g. Scattolini (2009). Although the computational demand is reduced, this will usually result in bad performance.

Distributed MPC refers to the case that each system features an MPC controller, but the separate controllers communicate with the other controllers, see bottom of Figure 1. We focus on so-called cooperative distributed MPC (see e.g. Scattolini (2009)), i.e. the distributed controllers solve the same optimization problem as the centralized controller using distributed optimization methods (c.f. Bertsekas and Tsitsiklis (1989)) and communicate to achieve this goal. In principle, the distributed controllers can achieve performance similar or very close to centralized MPC, while overcoming the problem of high central computational demand and offering increased reliability with respect to communication and system faults.

Partial support by the International Max Planck Research School Magdeburg and the German Research Foundation, grant FI 1505/3-1, are gratefully acknowledged.

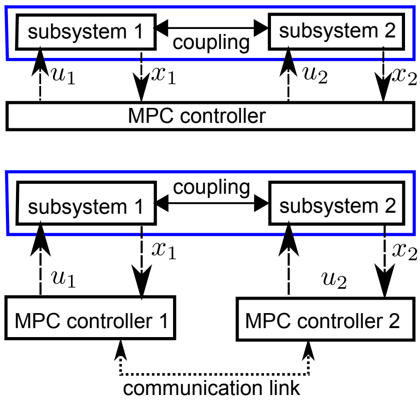


Fig. 1. Top: Centralized MPC, Bottom: Distributed MPC.

By now many approaches exist based on distributed optimization for cooperative distributed MPC of physically coupled systems. For example Wakasa et al. (2008); Giselsson and Rantzer (2010); Conte et al. (2012b) consider dual decomposition for cooperative distributed MPC. A nonlinear Jacobi algorithm is presented in Venkat et al. (2005); Stewart et al. (2010) considering only input constraints. A distributed version of Han's method is proposed in Doan et al. (2010). The work Scheu and Marquardt (2011) considers a sensitivity driven approach. An augmented Lagrangian based approach is investigated in Negenborn et al. (2009). Finally, the alternating direction multiplier method is investigated in for example Conte et al. (2012b); Kögel and Findeisen (2012).

Moreover, different stability criteria have been used for distributed MPC. Giselsson and Rantzer (2010) uses a strong controllability assumption to derive stopping and stability criteria. In Stewart et al. (2010) feasibility, stability and convergence criteria are presented, but only for the case of input constraints. Terminal equality constraint are proposed in Doan et al. (2010). Conte et al. (2012a) present a stability criterion using a distributed invariance approach. Kögel and Findeisen (2012) present a stability approach for distributed MPC using cyclically varying horizons for linear systems, which is extended to nonlinear systems in Kögel and Findeisen (2013).

Note that these works consider only the stabilization of fixed steady states / set-points (usually the origin). However, especially in chemical applications, often the steady state changes due to product quality and raw material changes. Thus, it is necessary to track the system to the new steady state without violating the constraints. Therefore, it is of interest to develop MPC schemes to track set-points as good as possible, which can be implemented in a distributed fashion to allow an efficient solution.

Several approaches for MPC based set-point tracking exist; for example off-set free MPC (see Muske and Badgwell (2002); Pannocchia and Rawlings (2003); Maeder and Morari (2010) and the references therein). We are motivated by the tracking scheme proposed in Limon et al. (2008); Ferramosca et al. (2009); Limon et al. (2012), since it easily allows to guarantee convergence and recursive feasibility. This setup has been generalized to distributed, tracking MPC by using a centralized terminal set in Ferramosca et al. (2013) and in Conte et al. (2013) using distributed, globally coupled terminal sets and distributed invariance.

Contribution: We present a framework for set-point tracking MPC using cyclic varying horizons, which have been previously proposed for regulation in Kögel and Findeisen (2012, 2013). Using a cyclic varying horizon guarantees that we can always enforce structure into the terminal set and constraints to allow a distributed implementation. In particular, we outline that the considered tracking scheme has similar properties as the centralized approach, i.e. it guarantees constraint satisfaction, recursive feasibility and tracking of the reference set-point. Furthermore, it is in some cases less restrictive than Conte et al. (2013); Ferramosca et al. (2013), c.f. Kögel and Findeisen (2012).

Paper structure: The remainder of the paper is structured as follows. In Section 2 we outline the problem setup. Section 3 presents the proposed distributed, predictive control for set-point tracking. The design of suitable terminal constraints and cost is discussed in Section 4. Section 5 outlines the applicability of the approach.

Notation: The notation is mainly standard. By $rem(a,b)$, ($a \in \mathbb{Z}$, $b \in \mathbb{N}$) we denote the remainder function. By $\mathbb{N}_{a,b}$ we denote the integers $a, a+1, \ldots, b$. For a matrix M, $M^\dagger$ is the Moore-Penrose pseudo-inverse. A matrix $M = \begin{pmatrix} M_{1,1} & M_{1,2} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$ consists of the block matrices $[M]_{i,j} = M_{i,j}$. With $\overline{u}(k)$ we the optimization variable corresponding to $u(k)$ and by $\overline{u}^\star(k|k-1))$ the optimal value of $\overline{u}(k)$ at time instance $k-1$.

2. PROBLEM SETUP

This section outlines the considered class of interconnected systems and our objective: set-point tracking.

Considered system class We consider linear, discrete-time, time-invariant systems of the form

$$x(k+1) = Ax(k) + Bu(k) \tag{1a}$$
$$y(k) = Cx(k) + Du(k), \tag{1b}$$

where $x(k) \in \mathbb{R}^n$ is the state, $u(k) \in \mathbb{R}^p$ the input and $y(k) \in \mathbb{R}^q$ the output and (A,B) is controllable.

The system consists of S interconnected subsystems. The state of subsystem i $x_i(k) \in \mathbb{R}^{n_i}$, the input $u_i(k) \in \mathbb{R}^{p_i}$ and the output $y(k) \in \mathbb{R}^{q_i}$ are parts of the overall state $x(k)$, input $u(k)$ and output $y(k)$, respectively, e.g. In detail:

$$x(k)^T = \big(x_1(k)^T \; x_2(k)^T \; \ldots \; x_S^T(k)\big) \tag{2}$$

In general not all subsystems might have an influence onto every subsystem. To account for the influences between the systems we introduce the sets $\mathcal{N}_i$ to denote the subsystems with an influence onto the state or output of subsystem i:

$$\mathcal{N}_i = \{j \text{ s. t. } A_{i,j} \neq 0, B_{i,j} \neq 0, C_{i,j} \neq 0 \text{ or } D_{i,j} \neq 0\}. \tag{3}$$

Thus, the dynamics of subsystem i is given by

$$x_i(k+1) = \sum_{j \in \mathcal{N}_i} \left(A_{i,j}x_j(k) + B_{i,j}u_j(k)\right) \tag{4a}$$
$$y_i(k) = \sum_{j \in \mathcal{N}_i} \left(C_{i,j}x_j(k) + D_{i,j}u_j(k)\right), \tag{4b}$$

where $A_{i,j}$, $B_{i,j}$, $C_{i,j}$ and $D_{i,j}$ are partitioned as

$$A = \begin{pmatrix} A_{1,1} & \ldots & A_{1,S} \\ \vdots & \ddots & \vdots \\ A_{S,1} & \ldots & A_{S,S} \end{pmatrix} \qquad B = \begin{pmatrix} B_{1,1} & \ldots & B_{1,S} \\ \vdots & \vdots & \ddots & \vdots \\ B_{S,1} & \ldots & B_{S,S} \end{pmatrix} \tag{5}$$
$$C = \begin{pmatrix} C_{1,1} & \ldots & C_{1,S} \\ \vdots & \ddots & \vdots \\ C_{S,1} & \ldots & C_{S,S} \end{pmatrix} \qquad D = \begin{pmatrix} D_{1,1} & \ldots & D_{1,S} \\ \vdots & \vdots & \ddots & \vdots \\ D_{S,1} & \ldots & D_{S,S} \end{pmatrix}. \tag{6}$$

For the constraints we assume that for each subsystem the state $x_i(k)$ and input $u_i(k)$ has to be in a convex, nonempty polytope $\mathcal{Z}_i$, which contains the origin

$$(x_i(k), u_i(k)) \in \mathcal{Z}_i \tag{7a}$$
$$\mathcal{Z}_i = \{x_i, u_i \text{ s.t. } X_i x_i + U_i u_i \leq z_i\}. \tag{7b}$$

Note the constraints do not involve states and inputs of other subsystems, i.e. they are separable constraints.

Set-point Tracking The objective is the tracking of a set-point: the output $y(k)$ should converge to a given reference y^{ref} or at least as close as possible, while satisfying the constraints (7).

We propose a framework for distributed MPC ensuring convergence to a steady state set-point y^s given by

$$0 = (A-I)x^s + Bu^s \tag{8a}$$
$$y^s = Cx^s + Du^s \tag{8b}$$
$$(x_i^s, u_i^s) \in (1-\epsilon)\mathcal{Z}_i, \; \forall i \in \mathbb{N}_{1,S}, \tag{8c}$$

which is as close as possible to the reference set-point y^{ref}. In particular, if y^{ref} satisfies for some x, u (8), then $y(k)$ converges to y^{ref}. The parameter ϵ in (8) with $0 < \epsilon < 1$ is a design parameter to guarantee that the steady state stays strictly in the feasible set and thus guarantees that stabilizability is maintained. ϵ can be freely chosen.

3. COOPERATIVE SET-POINT TRACKING MPC USING CYCLIC VARYING HORIZONS

This section presents the main result: the proposed MPC scheme for distributed tracking and some of its properties.

3.1 CST-MPC - Cyclic-horizon set-point tracking MPC

We propose a control scheme for distributed, tracking MPC, which we call cyclic-horizon set-point tracking MPC

(CST-MPC). The scheme allows tracking of a set-point, while avoiding constraint violations. Moreover, it enables a distributed implementation of the underlying optimization problem resulting in a distributed predictive controller.

As key element we use a cyclic varying horizon $N(k)$

$$N(k) = \underline{N} + M - \mathrm{rem}(k, M), \tag{9}$$

where $\underline{N} \geq 1$ is the minimum horizon length and $M \geq 1$ is the horizon period. $M = 1$ corresponds to the standard fixed horizon length of $\underline{N}$. For $M > 1$ the horizon varies between $\underline{N}$ and the maximum length $\underline{N}+M-1$: at $k = iM$, $i \in \mathbb{N}^0$ it has the maximum length ($\underline{N}+M-1$) and shrinks the next time instances. At $k = (i+1)M$ the horizon is restored to its maximum length, see Fig. 2.

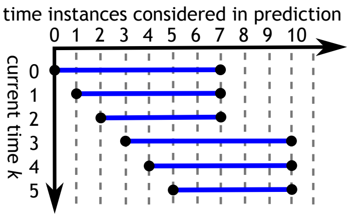


Fig. 2. Illustration of cyclic horizon for $\underline{N} = 5$, $M = 3$.

Thus at time k we optimize a state/input trajectory $\overline{\mathbf{x}}/\overline{\mathbf{u}}$

$$\overline{\mathbf{x}} = \{\overline{x}(k), \ldots, \overline{x}(k + N(k))\} \tag{10a}$$
$$\overline{\mathbf{u}} = \{\overline{u}(k), \ldots, \overline{u}(k + N(k) - 1)\}, \tag{10b}$$

and the steady state (u^s, x^s, y^s) subject to the dynamic (1), constraints (7) and steady-state map (8). Moreover, our objective is that the distance between the reference output y^{ref} and the steady-state output y^s is as small as possible, that recursive feasibility as well as stability are guaranteed, and that the controllers can be implemented in a distributed way to solve this problem.

The proposed CST-MPC scheme to solve this problem is based on the solution of the following optimal control problem denoted by $\mathcal{O}(x(k), y^{ref}, k)$

$$\min_{\overline{\mathbf{u}},\overline{\mathbf{x}},u^s,x^s,y^s,\alpha} J(\overline{\mathbf{u}}, \overline{\mathbf{x}}, u^s, x^s, y^s, y^{ref}) \tag{11a}$$

subject to

$$(A - I)x^s + Bu^s,\ y^s = Cx^s + Du^s \tag{11b}$$
$$(x_i^s, u_i^s) \in (1 - \alpha_i)\mathcal{Z}_i,\ \forall i \in \mathbb{N}_{1,S} \tag{11c}$$
$$\overline{x}(j + 1) = A\overline{x}(j) + B\overline{u}(j),\ \forall j \in \mathbb{N}_{k,k+N(k)-1} \tag{11d}$$
$$\overline{x}(k) = x(k) \tag{11e}$$
$$(\overline{x}_i(j), \overline{u}_i(j)) \in \mathcal{Z}_i,\ \forall i \in \mathbb{N}_{1,S},\ \forall j \in \mathbb{N}_{k,k+N(k)-1} \tag{11f}$$
$$\overline{x}_i(k + N(k)) - x_i^s \in \alpha_i\mathcal{T}_i,\ \forall i \in \mathbb{N}_{1,S} \tag{11g}$$
$$\epsilon \leq \alpha_i \leq 1,\ \forall i \in \mathbb{N}_{1,S},\ W\alpha \leq 0 \tag{11h}$$

where u^s, x^s, y^s denote an (artificial) steady state, $\epsilon > 0$ is a design parameter. The cost function J is given below and depends on the reference y^{ref}. $W\alpha \leq 0$ denotes a set of inequalities called terminal coupling inequalities and $\mathcal{T}_i$ are terminal sets, which we discuss later-on in detail.

Note that each optimization problem (11) depends on three parameters: the current overall system state $x(k)$, the reference output y^{ref} and the current time k (due to the time-varying length of the horizon $N(k)$). If (11) is feasible, then it delivers an optimal input sequence $\overline{\mathbf{u}}^\star$ depending on y^{ref}, x_k, k from which the first part $u(k) = u^\star(k)$ is used as feedback.

We assume that the cost function J is given by

$$\begin{aligned} J = &\sum_{j=k}^{k+N(k)-1} l(\overline{x}(j) - x^s, \overline{u}(j) - u^s) \\ &+ V^F(\overline{x}(k + N(k)) - x^s) + V^O(y^s - y^{ref}), \end{aligned} \tag{12}$$

where we consider convex quadratic functions for the stage cost $l(x, u)$, the terminal cost V^F and the offset cost V^O. Moreover, we assume that the cost is separable, i.e. the overall cost is given by the sum of the costs of each subsystem i. Consequently, we have that

$$l(x, u) = x^TQx + u^TRu = \sum_{i=1}^{S} x_i^TQ_ix_i + u_i^TR_iu_i \tag{13a}$$
$$V^F(x) = x^TFx = \sum_{i=1}^{S} x_i^TF_ix_i \tag{13b}$$
$$V^O(x) = y^THy = \sum_{i=1}^{S} y_i^TH_iy_i, \tag{13c}$$

where $Q = Q^T \geq 0$, $(A, Q^{\frac{1}{2}})$ detectable, $R = R^T > 0$, $F = F^T > 0$ and $H = H^T > 0$.

Remark 1. (Interpretation of optimization problem (11)) In the optimization problem (11) the equality constraints (11d), (11e) guarantee that the input sequence $\overline{\mathbf{u}}$ and state trajectory $\overline{\mathbf{x}}$ are consistent with the the dynamic (1) and (11f) imply satisfaction of the constraints (7). The constraints (11b), (11c) guarantee that the steady state (u^s, x^s, y^s) satisfies (8). The steady state is coupled with the terminal state $\overline{x}(N(k))$ by the constrains (11g), (11h).

In the cost function (12) the stage cost l and terminal cost V^f weight the difference between $\overline{\mathbf{u}}$, $\overline{\mathbf{x}}$ and the artificial steady state, the offset cost V^O penalizes differences between the artificial steady state and the reference y^{ref}.

Remark 2. (Application to distributed MPC) The cost function J (12) is separable. Moreover, all inequality constraints are separable except possible $W\alpha \leq 0$. However, it is possible to reformulate $W\alpha \leq 0$ as a set of equalities and separable constraints. This can for example be achieved by introducing copies of α for all subsystems and enforcing via equality constraints consensus, i.e. requiring that all copies have similar values.

Our setup allows to easily use tailored optimization methods to solve (11) in a distributed manner and thus facilitates distributed model predictive control, see e.g. Doan et al. (2010); Kögel and Findeisen (2012); Conte et al. (2012b). Note that one advantage of using a cyclic varying horizon is that under rather mild assumptions it is always possible to choose separable terminal constraints $\mathcal{T}_i$ with nonzero volume and separable terminal costs F_i, see Kögel and Findeisen (2013).

Remark 3. (Addition of a smoothing cost) One can add to the cost function J (12) an additional term to take the difference between the previous predicted optimal input and current input to be optimized, i.e. $\overline{u}(k) - \overline{u}^\star(k|k-1)$, into account and similarly for the state, which provides additional degrees of freedom. So it is possible to use for $k \geq 1$ the cost function

$$\begin{aligned} \tilde{J} =& J + (\overline{u}(k) - \overline{u}^\star(k|k-1))^TY(\overline{u}(k) - \overline{u}^\star(k|k-1)) \\ &+ (\overline{x}(k) - \overline{x}^\star(k|k-1))^TZ(\overline{x}(k) - \overline{x}^\star(k|k-1)), \end{aligned} \tag{14}$$

where $Y = Y^T \geq 0$ and $Z = Z^T \geq 0$. This results in the modified optimal control problem $\tilde{\mathcal{O}}(x(k), y^{ref}, k, \overline{x}^\star(k|k-1), \overline{u}^\star(k|k-1))$ given by

$$\min_{\overline{\mathbf{u}},\overline{\mathbf{x}},u^s,x^s,y^s,\alpha} \tilde{J} \qquad \text{subject to (11b)-(11h).} \tag{15}$$

3.2 System theoretic properties of CST-MPC

We can guarantee recursive feasibility, constraint satisfaction, stability and convergence of the proposed scheme under certain conditions as discussed in the following.

First let us assume that the terminal cost, terminal constraints, terminal coupling matrix and horizon period are designed such that they satisfy the following conditions.

Assumption 4. (Conditions on terminal conditions) *The terminal weighting matrices F, the terminal control gain K, the closed loop matrix $\tilde{A} = A + BK$, the terminal sets $\mathcal{T}_i$, horizon period M and terminal coupling matrix W satisfy*

a) $\forall x$ *with* $x_i \in \mathcal{T}_i$

$$x^T F x \geq \sum_{j=0}^{M-1} x^T (\tilde{A}^j)^T (Q + K^T R K) \tilde{A}^j x + x^T (\tilde{A}^M)^T F \tilde{A}^M x \tag{16}$$

b) *If* $x_i \in \alpha_i \mathcal{T}_i$, $i \in \mathbb{N}_{1,S}$ *and* $W\alpha \geq 0$, *then for* $l \in \mathbb{N}_{1,S}$ $(\tilde{x}(j)_l, \tilde{u}(j)_l) \in \alpha_l \mathcal{Z}_l$, $j \in \mathbb{N}_{0,M-1}$ *and* $\tilde{x}(M)_l \in \alpha_l \mathcal{T}_l$. *where* $\tilde{x}(0) = x$, $\tilde{x}(i+1) = \tilde{A}\tilde{x}(i)$ *and* $\tilde{u} = K\tilde{x}$.

Given these conditions it is possible to establish stability, convergence and recursive feasibility:

Theorem 5. (Recursive feasibility, convergence) *Let the terminal weighting matrices F, the terminal control gain K, the terminal sets $\mathcal{T}_i$, horizon period M and terminal coupling matrix W be designed such that Assumption 4 holds. Let either every $\mathcal{T}_i$ have a nonzero volume or $N \geq n$. If the optimization problem $\mathcal{O}(x(0), y^{ref}, 0)$ (11) is feasible, then we have:*

- *Recursive feasibility:* *Using the optimal feedback $u^\star(x(i), \overline{y}^{ref}(i), i)$ all optimization problems $\mathcal{O}(x(i), \overline{y}^{ref}(i), i)$ are feasible for any $\overline{y}^{ref}(i)$, where $x(i+1) = Ax(i) + Bu^\star(x(i), \overline{y}^{ref}(i), i)$.*
- *Convergence to the reference/stability:* *Using the optimal feedback $u^\star(x_i, y^{ref}, i)$ $x(k)$, $y(k)$ converges to x^a and y^a satisfying*

$$(x^a, u^a, y^a) \in \arg\min_{x^s,u^s,y^s} V^O(y^s - y^{ref}) \ \text{s.t. (8)}. \tag{17}$$

In particular, if there are x^s, u^s and y^s such that (8) holds with $y^s = y^{ref}$, then $y(k)$ converges to y^{ref}.

For space limitations the proof is avoided here. It builds upon results of Limon et al. (2008); Ferramosca et al. (2009) and Kögel and Findeisen (2013).

Note that in (17) y^a is unique (since V^0 is positive definite), but x^a might be non-unique, see e.g. Limon et al. (2012). Observe that the recursive feasibility is not lost if the reference changes. Moreover as usual in MPC (Scokaert et al. (1999)), it is not necessary to obtain the exact optimizers: it is possible to use suboptimal solutions if feasibility and a sufficiently large cost decrease (if possible) are guaranteed.

Using an additional smoothing cost term, i.e. solving (15) instead of (11), yields similar results. In particular, if $Y > 0$, then also $u(k)$ converges to a u^a satisfying (17), which might be for certain cases not guaranteed without a smoothing cost.

4. DESIGN OF TERMINAL ENTITIES

We derived conditions to guarantee convergence and feasibility. In the following we outline for some cases how to design suitable terminal control gain K, terminal weighting matrix P, horizon period M and the terminal coupling matrix W.

4.1 Symmetric constraints

Let us consider symmetric constraints, i.e. that the constraints (7) are given by $\|X_i x + U_i u_i\|_\infty \leq 1$, where $X_i \in \mathbb{R}^{m_i \times n_i}$, $U_i \in \mathbb{R}^{p_i \times n_i}$. Moreover, we aim to design terminal sets with the structure

$$\mathcal{T}_i = \{x_i \text{ s.t. } \|T_i x_i\|_\infty \leq c,\ i \in \mathbb{N}_{1,S}\} \tag{18a}$$

$$\mathcal{T} = \{x \text{ s.t. } x_i \in \mathcal{T}_i\} = \{x \text{ s.t. } \|Tx\|_\infty \leq c\} \tag{18b}$$

where $c > 0$ is some parameter to be determined and $T_i \in \mathbb{R}^{m_i \times n_i}$ are given with $m_i \geq n_i$. We assume that each $\mathcal{T}_i$ is bounded, i.e. T_i has n_i independent columns, and that $A + BK = \tilde{A}$ is asymptotic stable.

First let us consider $\alpha_i = 1, \forall i$. In order to have $x \in \mathcal{T} \Rightarrow \tilde{A}^M x \in \mathcal{T}$ we need to have

$$\|T\tilde{A}^M x\|_\infty \leq \|T\tilde{A}^M T^\dagger\|_\infty \|Tx\|_\infty \leq c \tag{19}$$

using the fact that the pseudo-inverse $T^\dagger$ satisfies $T^\dagger T = I$ (independent columns). So $\|T\tilde{A}^M T^\dagger\|_\infty \leq 1$ needs to hold, which can be achieved by choosing M large enough. So a suitable M can be computed by increasing M until the above condition is satisfied.

For the constraints we need $\|X\tilde{A}^l x + UK\tilde{A}^l x\|_\infty \leq 1$ for all $x \in \mathcal{T}$ and all $l \in \mathbb{N}_{0,M-1}$. Consequently, we need

$$\|X\tilde{A}^l T^\dagger + UK\tilde{A}^l T^\dagger\|_\infty \leq \frac{1}{c} \tag{20}$$

which can be satisfied by choosing c, $c > 0$ small enough.

Moreover to satisfy Assumption (4) we need to have that if $x_i \in \alpha_i \mathcal{X}_i$ and $W\alpha \geq 0$ $\forall j \in \mathbb{N}_{1,S}$, then $\tilde{A}^M x_i \in \alpha_i \mathcal{X}_i$ $\forall i \in \mathbb{N}_{1,S}$. So

$$\|T_i \tilde{x}_i^M\|_\infty = \sum_{j \in \mathcal{N}_i} \|[T\tilde{A}^M]_{i,j} x_j\|_\infty \tag{21}$$

$$\leq \sum_{j \in \mathcal{N}_i} \|[T\tilde{A}^M T^\dagger]_{i,j}\|_\infty \alpha_j c. \tag{22}$$

Note that $\|T_i \tilde{x}_i^M\|_\infty \leq \alpha_i c$ holds if $W_{i,j} \geq \|[T\tilde{A}^M T^\dagger]_{i,j}\|_\infty$, $i \neq j$ and $W_{i,i} \geq \|[T\tilde{A}^M T^\dagger]_{i,i}\|_\infty - 1$, since we have $\alpha_i \geq \sum_{i=1,j\neq i}^{j} W_{i,j}\alpha_j + (W_{i,i}+1)\alpha_i$. The constraints of Assumption 4 require that

$$\|[X\tilde{A}^l x + UK\tilde{A}^l x]_i\|_\infty = \|\sum_{j \in \mathcal{N}_i} [X\tilde{A}^l + UK\tilde{A}^l]_{i,j} x_j\|_\infty \leq \alpha_i, \tag{23}$$

which is true if $\frac{\|[X\tilde{A}^l T^\dagger + UK\tilde{A}^l T^\dagger]_{i,j}\|_\infty}{c} \leq W_{i,j}$, $i \neq j$ and $\frac{\|[X\tilde{A}^l T^\dagger + UK\tilde{A}^l T^\dagger]_{i,i}\|_\infty}{c} - 1 \leq W_{i,i}$. Hence choosing W such that all of the above conditions on W hold guarantee that Assumption 4 b) is satisfied.

Note that the above derivations can easily be extended to non-symmetric constraints: one method is to design the terminal cost, terminal constraints, terminal coupling matrix and horizon period by restricting (7) to a symmetric inner approximation of the constraints.

The terminal cost F can be determined straightforwardly by solving the linear matrix inequality given by (16) and restricting F to be block diagonal.

4.2 Terminal equality constraints

The simplest choice of terminal sets is $\mathcal{T}_i = \{0\}$, i.e. to enforce that the terminal state $\overline{x}(k + N(k))$ is equal to the steady state x^s. Clearly, Assumption 4 is satisfied for $M = 1$, $W = 0$, any K, F and independently of α_i. Thus, we can fix $\alpha_i = \epsilon$. However, this choice of the terminal sets often results in rather poor control performance.

4.3 Coupling via inputs, asymptotic stable systems

Another special case is given by asymptotic stable plants coupled via the inputs. Without a terminal control law, i.e. $K = 0$ and $\tilde{A} = A$, there are terminal constraints $\mathcal{T}_i$ such that for all $i \in \mathbb{N}_{1,S}$ and any $\alpha_i \in [0, 1]$

$$x_i \in \alpha_i \mathcal{T}_i \Rightarrow A_{i,i} x_i \in \alpha_i \mathcal{T}_i \text{ and } (x_i, 0) \in \alpha_i \mathcal{Z}_i. \qquad (24)$$

As a consequence $\alpha_i \mathcal{T}_i$ are positive invariant, i.e. $x_i \in \alpha_i \mathcal{T}_i$ implies that $A_{i,i} x_i \in \alpha_i \mathcal{T}_i$, $\forall i$. Moreover, the second part of the above equation (24) implies that if $x_i \in \mathcal{T}_i$ for all i, then the open-loop system will not violate the constraints. Different choices for $\mathcal{T}_i$ are possible, e.g. small enough sublevel set of the open loop cost.

In order to satisfy condition a) of Assumption 4 one can use as terminal costs F_i the solution of the Lyapunov equations

$$F_i = A_{i,i}^T F_i A_{i,i} + Q_i = 0.$$

In a summary, for this special case one can select terminal sets $\mathcal{T}_i$ and terminal costs F_i satisfying the conditions of Assumption 4 for $K = 0$, $M = 1$ and with $W = 0$. However using $K = 0$ might lead to bad performance.

5. SIMULATION EXAMPLE

We consider as simulation examples a reactor chain with a non-adiabatic flash separator. The system (taken from Venkat et al. (2005)) consists of two continuous stirred tank reactors (CSTRs) followed by a non-adiabatic flash as illustrated in Figure 3. Two irreversible reactions take place in both reactors: $A \to B$ and $B \to C$, where B is the desired product and C a side product. In the flash separator A is separated from B and C. The vapor consists mostly of A and is therefore partly recycled and the liquid bottom phase containing mostly B and C is removed.

From the nonlinear model we obtained, similar as in Venkat et al. (2005), a linearized model. The flows F_0, F_1, D and heat exchangers Q_r, Q_m, Q_b can be manipulated. The system is splitted into three subsystems: the first incorporates the dynamics of the left CSTR and controls the input flow F_I and the heat flow Q_r, the second subsystem is the right CSTR has as controls the flow F_{II} and Q_m. The last subsystem is given by the dynamic of the flash and the controls H_m and D.

As constraints we assume that the states and input need to be limited to $\pm 10\%$ of the steady state value. For the MPC setup we use $Q = I$ and $R = 100I$, $Y = Z = 0$ and $\underline{N} = 20$.

We transformed the system into new coordinates such that the state constraints correspond to $\|x\|_\infty \leq 1$. Using K as the LQR gain and choosing $M = 5$ we compute

$$W = \begin{pmatrix} -0.992 & 0.082 & 0.060 \\ 0.032 & -0.919 & 0.289 \\ 0.0129 & 0.1210 & -0.9321 \end{pmatrix}$$

$$\mathcal{T} = \{x \text{ s.t. } \|x\|_\infty \leq 0.14\}$$

$$F_1 = \begin{pmatrix} 589 & -0.043 & -0.013 & -0.20 \\ -0.043 & 300.09 & 0.024 & -0.068 \\ -0.013 & 0.024 & 300 & -0.020 \\ -0.20 & -0.068 & -0.020 & 1261 \end{pmatrix}$$

$$F_2 = \begin{pmatrix} 686 & -4.19 & -1.29 & -18.76 \\ -4.19 & 306 & 2.47 & -5.67 \\ -1.29 & 2.4 & 300 & -2.35 \\ -18.8 & -5.67 & -2.35 & 1287 \end{pmatrix}$$

$$F_3 = \begin{pmatrix} 300 & 0 & 0 & 0 \\ 0 & 300 & 0 & 0 \\ 0 & 0 & 301 & 0 \\ 0 & 0 & 0 & 1261 \end{pmatrix},$$

which satisfy the condition of Assumption 4.

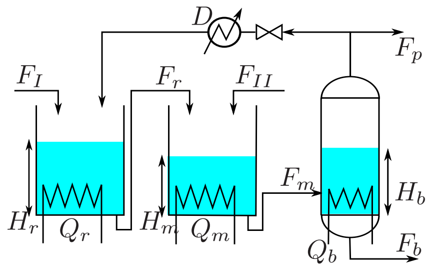


Fig. 3. Example: Two reactors followed by a flash drum.

We consider the tracking of the concentration of A and B in the flash (outputs y^I and y^{II}, respectively). We use $\epsilon = 0.001$ and $H = \text{diag}(2, 200)$. As illustrated in Figure 4 tracking works well in the first 150 steps. However, afterwards due to the saturation of the inputs, this is no longer the case.

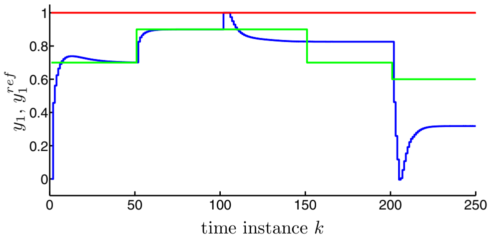
In summary, for this example the proposed tracking scheme using distributed MPC works well.

6. SUMMARY AND POSSIBLE EXTENSIONS

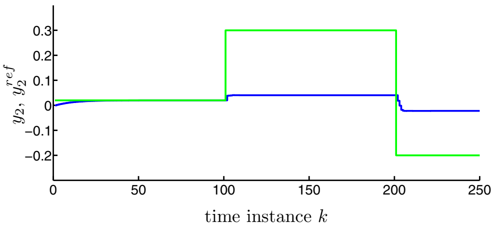
We presented a set-point tracking scheme for model predictive control. We outlined how using a cyclic varying horizon enables a distributed implementation of this scheme. A simulation example illustrated the proposed approach.

Extension to uncertain system Often the plant is uncertain, e.g. disturbances act on the plant. For bounded, additive disturbances tube-based robust MPC, see e.g. Mayne et al. (2011), can be used, c.f. Limon et al. (2012).

Nonlinear systems An extension to nonlinear systems based on Limon et al. (2012) and the references therein and Kögel and Findeisen (2012) seems to be possible.



Output y_1 (blue), reference y_1^{ref} (green), constr. (red).



Output y_2 (blue), reference y_2^{ref} (green).

Fig. 4. Plots for CSTRs flash system.

Utilization of an additional prediction horizon Note that enforcing an equality constraint $x(k+1) = (A+BG)x(k)$ where $\tilde{A} = A + BG$ is Schur stable and the constraints (7) can be done within distributed optimization methods, especially if G is sparse. Under mild conditions, one can choose N_p large enough such that the set

$$\mathcal{S} = \{x | x(i+1) = \tilde{A}x(i), x(0) = x, (x_j(i), u_j(i)) \in \mathcal{Z}_j \\ i \in \mathbb{N}_{0,N_p-1}\},$$

is invariant, i.e. can be used as terminal set and might be significantly larger than separable sets. In particular, for large enough N_p this set might be the maximum admissible set, see Rawlings and Muske (1993).

REFERENCES

Bertsekas, D.P. and Tsitsiklis, J.N. (1989). *Parallel and Distributed Computation - Numerical Methods*. Prentice-Hall.

Conte, C., Voellmy, N., Zeilinger, M., Morari, M., and Jones, C. (2012a). Distributed Synthesis and Control of Constrained Linear Systems. In *Proc. Amer. Contr. Conf.*, 6017–6022.

Conte, C., Summers, T., Zeilinger, M.N., Morari, M., and Jones, C.N. (2012b). Computational aspects of distributed optimization in model predictive control. In *Proc. IEEE Conf. on Dec. and Contr.*, 6819–6824.

Conte, C., Zeilinger, M.N., Morari, M., and Jones, C.N. (2013). Cooperative Distributed Tracking MPC for Constrained Linear Systems: Theory and Synthesis. Proc. IEEE Conf. on Dec. and Contr.

Doan, D., Keviczky, T., and Schutter, B.D. (2010). An improved distributed version of Han's method for distributed MPC of canal systems. In *Proc. 12th IFAC Symp. on Large Scale Systems*.

Ferramosca, A., Limon, D., Alvarado, I., and Camacho, E. (2013). Cooperative distributed MPC for tracking. *Automatica*, 49(4), 906 – 914.

Ferramosca, A., Limon, D., Alvarado, I., Alamo, T., and Camacho, E. (2009). MPC for tracking with optimal closed-loop performance. *Automatica*, 45(8), 1975–1978.

Giselsson, P. and Rantzer, A. (2010). Distributed Model Predictive Control with Suboptimality and Stability Guarantees. In *Proc. IEEE Conf. on Dec. and Contr.*, 7272 – 7277.

Kögel, M. and Findeisen, R. (2012). Cooperative distributed MPC using the alternating direction multiplier method. In *8th IFAC International Symp. on Advanced Control of Chemical Processes*, 445–450.

Kögel, M. and Findeisen, R. (2013). Stability of NMPC with cyclic horizons. In *Proc. IFAC Symp. on Nonlinear Control Systems 2013*, 809–814.

Limon, D., Alvarado, I., Alamo, T., and Camacho, E. (2008). MPC for tracking piecewise constant references for constrained linear systems. *Automatica*, 44(9), 2382–2387.

Limon, D., Ferramosca, A., Alamo, T., Gonzalez, Alejandro, H., and Odloak, D.J. (2012). Model Predictive Control for Changing Economic Targets. In *Proc. IFAC Conf. on Nonlinear Model Predictive Control*, 384–391.

Maciejowski, J.M. (2002). *Predictive Control with Constraints*. Prentice Hall, Upper Saddle River, New Jersey.

Maeder, U. and Morari, M. (2010). Offset-free reference tracking with model predictive control. *Automatica*, 46(9), 1469–1476.

Mayne, D.Q., Kerrigan, E.C., and Falugi, P. (2011). Robust model predictive control: advantages and disadvantages of tube-based methods. In *Proc. 18th IFAC World Congress*, 191–196.

Muske, K.R. and Badgwell, T.A. (2002). Disturbance modeling for offset-free linear model predictive control. *Jour. Proc. Contr.*, 12(5), 617–632.

Negenborn, R.R., van Overloop, P.J., Keviczky, T., and De Schutter, B. (2009). Distributed model predictive control for irrigation canals. *Networks and Heterogeneous Media*, 4(2), 359–380.

Pannocchia, G. and Rawlings, J.B. (2003). Disturbance models for offset-free model-predictive control. *AIChE Journal*, 49(2), 426–437.

Qin, S. and Badgwell, T. (2003). A survey of industrial model predictive control technology. *Contr. eng. prac.*, 11(7), 733–764.

Rawlings, J.B. and Muske, K. (1993). The stability of constrained receding horizon control. *IEEE Tran. on Auto. Contr.*, 38, 1512–1516.

Scattolini, R. (2009). Architectures for distributed and hierarchical model predictive control - a review. *Jour. Proc. Contr.*, 19(5), 723 – 731.

Scheu, H. and Marquardt, W. (2011). Sensitivity-based coordination in distributed model predictive control. *Jour. of Proc. Contr.*, 21, 715–728.

Scokaert, P.O.M., Mayne, D.Q., and Rawlings, J.B. (1999). Suboptimal model predictive control (feasibility implies stability). *IEEE Tran. on Auto. Contr.*, 44(3), 648–654.

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

Venkat, A., Rawlings, J., and Wright, S. (2005). Stability and optimality of distributed model predictive control. In *Proc. IEEE Conf. on Dec. and Contr.*, 6680–6685.

Wakasa, Y., Arakawa, M., Tanaka, K., and Akashi, T. (2008). Decentralized model predictive control via dual decomposition. In *Proc. IEEE Conf. on Dec. and Contr.*, 381 – 386.