Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

A distributed model predictive control scheme for networks with communication failure

Rainer M. Schaich * Matthias A. Müller ** Frank Allgöwer **

* *Department of Engineering Science, University of Oxford, OX1 3PJ, UK. (e-mail: rainer.schaich@eng.ox.ac.uk).*
** *Institute of Systems Theory and Automatic Control, University of Stuttgart, Stuttgart, Germany. (e-mail: {matthias.mueller,frank.allgower}@ist.uni-stuttgart.de)*

Abstract: A method to ensure recursive feasibility and asymptotic stability for a distributed model predictive control scheme in case of temporary communication failure is presented. The considered networks are made up of dynamically uncoupled nonlinear systems, coupled through constraints and objectives. The proposed method substitutes affected coupling constraints in such a way that the scheme can not become infeasible, while a decrease in the cost is guaranteed.

Keywords: Distributed model predictive control; Communication failure; Collision avoidance.

1. INTRODUCTION

In recent years the control of networks, consisting of interacting dynamical systems (multi agent systems) that work together to achieve a cooperative task has increasingly attracted researchers and produced various interesting results, see Olfati-Saber et al. [2007] for a short overview. To optimize the network in its task we use the framework of distributed model predictive control (DMPC). Over the past years a variety of DMPC algorithms have been proposed, see [Scattolini, 2009] and [Christofides et al., 2013] for an overview. These DMPC algorithms vary in the way they handle interconnectivity in coupling costs as well as in coupling constraints. The way in which adjacent systems (systems coupled by constraints and costs) optimize is important to ensure the recursive feasibility of the DMPC algorithm. A parallel optimization scheme requires either an additional penalization of the systems' trajectories to guarantee the feasibility of its neighbours (see e.g. [Dunbar, 2007]), or each system has to solve its optimization problem and exchange information with its neighbours multiple times at each time step, such as in [Doan et al., 2011]. An alternative way to ensure recursive feasibility is to optimize sequentially, i.e. adjacent systems do not optimize in parallel and exchange trajectories in intermediate optimization steps, so that coupling constraints are respected at all times; such sequential algorithms have been used in DMPC schemes, e.g. by Richards and How [2007] and Müller et al. [2012].

In this paper we discuss networks of $p \geq 2$ systems which are dynamically decoupled, but are coupled through the objective and constraints. We follow the work presented in Müller et al. [2012] to obtain a distributed model predictive control scheme, that we extend to be able to handle communication failure robustly. As in Müller et al. [2012] we use a sequential algorithm to be able to respect coupling constraints without additional penalizing cost terms and without the burden of intermediate communication and multiple optimization runs for each system in each time step as in Doan et al. [2011]. In a sequential optimization algorithm each system will only optimize once in each time step (as opposed to multiple times in iterative parallel schemes) and therefore the required communication load is reduced. In this paper we propose a way to handle the temporary absence of communication of adjacent systems by means of substituting the associated coupling constraints by constraints that restrict subsequent optimized trajectories to be inside a tube around the last transmitted trajectory. The size of the tube allows us to ensure the decrease of the overall objective function in addition to ensuring the recursive feasibility. A similar method to handle communication failures in a DMPC scheme has been presented in Alessio and Bemporad [2008] for linear systems without state constraints and hence without feasibility issues in absence of communication. Furthermore, in Heidarinejad et al. [2011], a Lyapunov-based distributed MPC scheme was presented, where communication failures can be treated under the assumption that some Lyapunov-based auxiliary controller is known. The control scheme presented here can be used for decoupled, nonlinear system dynamics, coupled pairwise through constraints.

The remainder of this paper is structured as follows: In section 2 we present the considered setup and the algorithm we use to solve DMPC problems, in section 3 we extend the algorithm presented in section 2 to the case with communication failures and sketch the ideas behind the proposed way to ensure feasibility and stability in absence of communication. Simulation results of a collision avoidance example are presented in section 4, and we conclude the paper in section 5.

1.1 Notation

For some non-negative integer $q \in \mathbb{N}_0$ let $\mathcal{I}(q) := \{0, 1, \ldots, q-1, q\}$. For a real number $a \in \mathbb{R}$ we denote its absolute value by $|a|$, for the real vector $v \in \mathbb{R}^n$ we define the ℓ_p norm and ℓ_∞ norm as $\|v\|_p := (\sum_{k=1}^{n} |v_k|^p)^{\frac{1}{p}}$ and $\|v\|_\infty := \max_k |v_k|$ respectively. We call a continuous, strictly increasing function $f \in C([0,b))$ a $\mathcal{K}$ function, if it satisfies $f(0) = 0$. If it is also unbounded we call it a $\mathcal{K}_\infty$ function.

The network of systems will be described by an undirected graph $G = (V, E)$, where V is the vertex set and E is the edge set $E \subset V \times V$. The neighbourhood set of a vertex $s_i \in V$ is denoted by $\mathcal{N}_i := \{s_j \in V \,:\, (s_j, s_i) \in E\}$.

2. NETWORKS WITHOUT COMMUNICATION FAILURE

Each agent is described by a nonlinear, discrete time difference equation

$$x_i[k+1] = f_i(x_i[k], u_i[k]) \tag{1}$$

with the state $x_i[k] \in \mathbb{R}^{n_i}$ and the input $u_i[k] \in \mathbb{R}^{m_i}$ and a continuous function $f_i : \mathbb{R}^{n_i} \times \mathbb{R}^{m_i} \to \mathbb{R}^{n_i}$. The control objective is to stabilize each system's set point $\hat{x}_i$ while respecting coupling constraints and optimizing some possibly coupled performance criterion, i.e. we consider the DMPC optimization problems

$$\min_{\mathbf{u}_i \in U_i^{N_h}} J_i(\mathbf{x}_i^{\mathbf{u}_i}[l], \mathbf{u}_i, \mathbf{x}_{\mathcal{N}_i}^p[l]) \tag{2}$$

subject to the constraints:

$$x_i^{\mathbf{u}_i}[0|l] = x_i[l] \tag{3a}$$

$$x_i^{\mathbf{u}_i}[k+1|l] = f_i(x_i^{\mathbf{u}_i}[k|l], u_i[k]) \tag{3b}$$

$$c_{ii}(x_i^{\mathbf{u}_i}[k|l]) \leq 0 \;\forall\, k \in \mathcal{I}(N_h - 1) \tag{3c}$$

$$c_{ij}(x_i^{\mathbf{u}_i}[k|l], x_i^{p}[k|l]) \leq 0 \;\forall\, s_j \in \mathcal{N}_i \wedge k \in \mathcal{I}(N_h - 1) \tag{3d}$$

$$d_{ii}(x_i^{\mathbf{u}_i}[N_h|l]) \leq 0. \tag{3e}$$

The input constraint set U_i is assumed to be compact in $\mathbb{R}^m$, the coupling constraints are assumed to be pairwise equivalent, i.e. $c_{ij}(x_i, x_j) \equiv c_{ji}(x_j, x_i)$ for all admissible x_i and x_j. The functions c_{ii} and d_{ii} define the local state constraint set X_i and the decoupled terminal constraint set X_i^f respectively. We assume that $c_{ij}(x_i, x_j) \leq 0$ for $x_i \in X_i^f$ and $x_j \in X_j^f$. In (2) and (3) we used the shortened notation $\mathbf{x}_i^{\mathbf{u}_i}[l] = (x_i^{\mathbf{u}_i}[0|l], \ldots, x_i^{\mathbf{u}_i}[N_h|l])$ which is the predicted state trajectory starting at $x_i[l]$ and using $\mathbf{u}_i := (u_i[0], \ldots, u_i[N_h - 1])$ as an input, whereas $\mathbf{x}_{\mathcal{N}_i}^p[l] = \{\mathbf{x}_j^p[l]\}_{s_j \in \mathcal{N}_i}$ denotes the predicted trajectories of all adjacent systems for which we will discuss later how to obtain them.

In (2) the objective function itself is given by:

$$J_i(\mathbf{x}_i^{\mathbf{u}_i}[l], \mathbf{u}_i, \mathbf{x}_{\mathcal{N}_i}^p[l]) = \sum_{k=0}^{N_h-1} L_{ii}(x_i^{\mathbf{u}_i}[k|l], u_i[k]) + \sum_{s_j \in \mathcal{N}_i} L_{ij}(x_i^{\mathbf{u}_i}[k|l], x_j^p[k|l]) + F_{ii}(x_i^{\mathbf{u}_i}[N_h|l]), \tag{4}$$

where L_{ii} denotes the self associated stage cost, L_{ij} denotes the coupling stage cost and F_{ii} denotes the terminal cost. It is important that there is no coupling in the terminal cost. We assume all functions in (3) and (4) to be locally Lipschitz continuous. For the individual terms in (4) we assume that for all $s_i \in V$ and all $(s_i, s_j) \in E$ there exist $\gamma_i, \Gamma_i, \phi_i, \Phi_i, \theta_i, \Theta_i, b_{ij}, B_{ij} \in \mathcal{K}_\infty$ such that

$$\gamma_i(\|x_i - \hat{x}_i\|_2) + \phi(\|u_i - \hat{u}_i\|_2) \leq L_{ii}(x_i, u_i) \leq \Gamma_i(\|x_i - \hat{x}_i\|_2) + \Phi(\|u_i - \hat{u}_i\|_2) \tag{5a}$$

$$\theta(\|x_i - \hat{x}_i\|_2) \leq F_{ii}(x_i) \leq \Theta(\|x_i - \hat{x}_i\|_2) \tag{5b}$$

$$0 \leq L_{ij}(x_i, x_j) \leq b_{ij}(\|x_i - \hat{x}_i\|_2) + b_{ij}(\|x_j - \hat{x}_j\|_2) \tag{5c}$$

for all $x_i \in X_i \subseteq \mathbb{R}^{n_i}$, $x_j \in X_j \subseteq \mathbb{R}^{n_j}$, and $u_i \in U_i \subset \mathbb{R}^{m_i}$. In addition to (5) we require the coupling stage cost to be dominated by the self associated stage costs inside the terminal region:

$$L_{ij}(x_i, x_j) \leq \tilde{c}_{ij}^i\, L_{ii}(x_i, u_i) + \tilde{c}_{ij}^j\, L_{jj}(x_j, u_j) \tag{6}$$

for all $x_j \in X_j^f$, $x_j \in X_j^f$ and $u_i \in U_i$. For all $s_i \in V$ define $\tilde{c}^i := \sum_{s_j \in \mathcal{N}_i} \tilde{c}_{ij}^i + \tilde{c}_{ji}^i$.

Remark 1. In (5c) we do not bound the coupling cost L_{ij} from below in terms of $\|x_i - \hat{x}_i\|_2$ and $\|x_j - \hat{x}_j\|_2$. We allow the coupling cost to vanish for some constellation of x_i and x_j. This is for example of interest for collision avoidance constraints, where we do not penalize states that are sufficiently far apart.

Furthermore we assume to have a local auxiliary controller k_i^{loc} that satisfies for all $x_i \in X_i^f$

(i) the input constraints, i.e. $k_i^{loc}(x_i) \in U_i$,
(ii) the inequality $F_{ii}(f_i(x_i, k_i^{loc}(x_i))) - F_{ii}(x_i) \leq -(1 + \tilde{c}^i) \cdot L_{ii}(x_i, k_i^{loc}(x_i))$ and
(iii) renders the terminal region closed-loop invariant, i.e. $f_i(x_i, k_i^{loc}(x_i)) \in X_i^f$

The assumptions (i-iii) are standard assumptions in the MPC context, see Mayne et al. [2000]; assumption (ii) is slightly modified to enable us to compute terminal controllers in a distributed fashion. We define the appended and shifted input sequence $a(\mathbf{u}_i)[l+1] = (u_i[1|l], \ldots, u_i[N_h|l], k_i^{loc}(x_i^{\mathbf{u}_i}[N_h|l]))$, $a^2(\mathbf{u}_i)[l+2] = a(a(\mathbf{u}_i))[l+2]$. Note that $a(\mathbf{u}_i)[l+1]$ is an input sequence admissible at time step $l+1$ while $\mathbf{u}_i = \mathbf{u}_i[l]$. We neglect the second time argument since appending the local controller is only useful for the previous input sequence.

Remark 2. With (6) and the second assumption on the terminal controller we are able to compute independent terminal controllers and therefore to decouple the problem inside the terminal region (see Müller et al. [2012, section 4] for a particular example). This fact will be used later to construct feasible trajectories with previously transmitted trajectories.

2.1 A distributed model predictive control algorithm for networks without communication failure

In this section we present a DMPC algorithm for constant communication topologies, which is a modification of the first algorithm presented in Müller et al. [2012]. Systems optimize sequentially and exchange data to ensure a decrease of the sum of all objective functions. In each time step, i.e. between $l+1$ and l, all optimizations have to take place, hence we use fractions of the sampling time T to describe the intermediate times; $t = lT$ denotes the continuous time instant for the discrete time instant l. We assume to have initially feasible input sequences $\mathbf{u}_{i,init}$, which for example can be obtained by a centralized computation or by a suitable distributed optimization algorithm. We assume that the initial input sequences are such that $\sum_{s_i \in V} J_i(\mathbf{x}_i^{\mathbf{u}_{i,init}}[0], \mathbf{u}_{i,init}, \mathbf{x}_{\mathcal{N}_i}^p[0]) \le \sum_{s_i \in V} \delta_i(\|\tilde{x}_i - \hat{x}_i\|)$ for some $\delta_i \in \mathcal{K}_\infty$ and all $\tilde{x}_i \in X_i^f$, for all $s_i \in V$; herein $\mathbf{x}_{\mathcal{N}_i}^p[0] = \{\mathbf{x}_j^{\mathbf{u}_{j,init}}[0]\}_{s_j \in \mathcal{N}_i}$. This is a technical assumption which we need in order to prove stability and not only asymptotic convergence of the closed loop trajectories. Note that this assumption is not restrictive, since it is for example satisfied if for all $\tilde{x}_i \in X_i^f$, $\mathbf{u}_{i,init}$ is given as an appended sequence of the local auxiliary controller k_i^{loc}. The assumption then follows from assumption (ii) on the local controller k_i^{loc}. Since we use a sequential algorithm, we assume to have the optimization schedule $0 = t_{s_{i_1}} \le t_{s_{i_2}} \le \cdots \le t_{s_{i_q}} = T - \rho$, where $q \le p$ is the number of optimizations per time step determined by the parallelizability of the network, i.e. time slots $[t_i, t_{i+1})$ in which individual systems optimize. It is important that adjacent systems do not optimize at the same time, otherwise the optimization may be parallelized, hence q can be smaller than the number of systems p.

Algorithm The algorithm itself has four different steps:

- Initialization: Every system $s_i \in V$ evaluates $\mathbf{x}_i^{\mathbf{u}_{i,init}}[0]$ and sends it to all systems in its neighbourhood $s_j \in \mathcal{N}_i$. All neighbours $s_j \in \mathcal{N}_i$ send their trajectory $\mathbf{x}_j^{\mathbf{u}_{j,init}}[0]$ to system s_i, which stores them in its local memory $\mathbf{x}_{\mathcal{N}_i}^p[0] = \{\mathbf{x}_j^{\mathbf{u}_{j,init}}[0]\}_{s_j \in \mathcal{N}_i}$. Define $\mathbf{u}_{i,MPC} = \mathbf{u}_{i,init}$ and proceed to:
- Optimization: During the time slot $t \in [t_{s_i} + lT, t_{s_i+1} + lT)$ system s_i solves its optimization problem (2)–(3) to obtain the minimizer $\mathbf{u}_i^*$ and the associated trajectory $\mathbf{x}_i^{\mathbf{u}_i^*}[l]$, which it sends to all neighbours. System s_i computes
$$\delta_i = J_i(\mathbf{x}_i^{\mathbf{u}_i^*}[l], \mathbf{u}_i^*, \mathbf{x}_{\mathcal{N}_i}^p[l]) - J_i(\mathbf{x}_i^{\mathbf{u}_{i,MPC}}[l], \mathbf{u}_{i,MPC}, \mathbf{x}_{\mathcal{N}_i}^p[l]). \quad (7)$$
When all neighbours have returned the change in their objective (9) i.e. δ_j, system s_i computes
$$\ell_i = \delta_i + \sum_{s_j \in \mathcal{N}_i} \delta_j. \quad (8)$$
If $\ell_i \le 0$ then system s_i sets $\mathbf{u}_{i,MPC} = \mathbf{u}_i^*$ and sends out $\mathbf{x}_i^{\mathbf{u}_i^*}[l]$ again.
- Refreshing: Outside of the assigned time slot $t \in [lT, t_{s_i} + lT) \cup [t_{s_i+1} + lT, (l+1)T - \rho)$ system s_i receives potential trajectories from its neighbours to compute the change in objective, i.e. $s_j \in \mathcal{N}_i$ sends $\mathbf{x}_j^*$ to s_i, system s_i computes
$$\delta_i = \sum_{k=0}^{N_h-1} L_{ij}(x_i^{\mathbf{u}_{i,MPC}}[k|l], x_j^*[k|l]) - L_{ij}(x_i^{\mathbf{u}_{i,MPC}}[k|l], x_j^p[k|l]) \quad (9)$$
and returns δ_i to system s_j. If system s_j sends $\mathbf{x}_j^*$ again system s_i sets $\mathbf{x}[l]_j^p = \mathbf{x}_j^*$ in its local memory.
- Implementation & updating: During the time interval $t \in [(l+1)T - \rho, (l+1)T)$ system s_i implements $u_{i,MPC}[0]$ and shifts and appends all trajectories in its local memory, its own and its neighbours' , i.e.

$$\mathbf{x}_j^p[l+1] = (x_j^p[1|l], \ldots \ldots, x_j^p[N_h|l], f_j(x_j^p[N_h|l], k_j^{loc}(x_j^p[N_h|l]))) \quad (10a)$$
and
$$\mathbf{u}_{i,MPC} = a(\mathbf{u}_{i,MPC}). \quad (10b)$$

For this algorithm we can prove

Theorem 3. If we initialize the algorithm with feasible initializers $\mathbf{u}_{i,init}$ for all $s_i \in V$, then the algorithm is recursively feasible and the resulting closed loop is asymptotically stable with respect to the set points $\hat{x}_i$ for all $s_i \in V$.

Proof. In the following, we only sketch the proof of Theorem 3; for more details, the interested reader is referred to [Müller et al., 2012]. The recursive feasibility of the algorithm is ensured by the assumption that the optimization schedule does not allow adjacent systems to optimize at the same time and that we use equivalent constraints $c_{ij}(x_i, x_j) \equiv c_{ji}(x_j, x_i)$. Therefore all intermediate trajectories will be admissible for all neighbouring systems.

For the asymptotic stability we use the abstract sum of all objectives with perfect knowledge about all states, i.e. the function
$$J(\mathbf{x}^{\mathbf{u}_{MPC}}[l], \mathbf{u}_{MPC}[l]) = \sum_{s_i \in V} J_i(\mathbf{x}_i^{\mathbf{u}_{i,MPC}}[l], \mathbf{u}_{i,MPC}[l], \mathbf{x}_{\mathcal{N}_i}^{\mathbf{u}_{\mathcal{N}_i,MPC}}[l]) \quad (11)$$
as Lyapunov function candidate with respect to the set points. By standard MPC argumentation, we know that at time instant $t = lT$ we have
$$\underbrace{J(\mathbf{x}^{a(\mathbf{u}_{MPC})}[l], a(\mathbf{u}_{MPC})[l])}_{J_0[l]} \le \underbrace{J(\mathbf{x}^{\mathbf{u}_{MPC}}[l-1], \mathbf{u}_{MPC}[l-1])}_{J_e[l-1]} - \Bigg(\sum_{s_i \in V} \Big(L_{ii}(x_i^{\mathbf{u}_{i,MPC}}[0|l-1], u_{i,MPC}[0|l-1]) + \sum_{s_j \in \mathcal{N}_i} L_{ij}(x_i^{\mathbf{u}_{i,MPC}}[0|l-1], x_j^{\mathbf{u}_{j,MPC}}[0|l-1])\Big)\Bigg) \le J_e[l-1] - \sum_{s_i \in V} L_{ii}(x_i^{\mathbf{u}_{i,MPC}}[0|l-1], u_{i,MPC}[0|l-1]). \quad (12)$$
We only use an optimized input sequence if it decreases the value of the sum of all neighbours objectives. So that at the end of the optimizations for time step l, i.e. at $t = (l+1)T - \rho$, we have
$$J_e[l] = J_0[l] + \sum_{\ell_i \le 0} \ell_i. \quad (13)$$
Thus we know that the sequence $J_e[l]$ is strictly decreasing. From here, asymptotic stability of the closed-loop system with respect to the set points $\hat{x}_i$ can be established as in standard MPC stability proofs (see e.g. Mayne et al. [2000]).

Remark 4. In the updating step each system updates its own as well as its neighbour's trajectories, this requires every system to have the local controller and system dynamics of all adjacent systems.

The presented algorithm is a modified version of algorithm 1 in Müller et al. [2012] and can so far only handle systems with perfect communication.

3. NETWORKS WITH COMMUNICATION FAILURE

In this section we extend the previously introduced algorithm to networks where communication between adjacent systems is temporarily lost. We require the communication to be working at the initialization. If the communication link between two systems s_α and s_β fails, we substitute the affected coupling constraints in order to ensure that the coupling constraints are still fulfilled. We choose a substitute constraint in such a way that even though the new predicted trajectories cannot be transmitted and hence $c_{\alpha\beta}$ and $c_{\beta\alpha}$ in the optimization problem (2)–(3) can not be evaluated, we can still guarantee recursive feasibility by a local state constraint. Other than that the algorithm presented above still works in the same way, except that now δ_α and δ_β can no longer be exchanged. Furthermore, in the updating step of the algorithm, equation (10a) to be precise, systems s_α and s_β just update the last predicted trajectory of the respective neighbour which was successfully transmitted. As soon as the communication between the systems s_α and s_β is reestablished the systems exchange their latest trajectories and store them in their local memory. For clarity of presentation in the following we will show in more detail how feasibility and stability can still be maintained under the failure of one communication link only; we then remark how the presented method can be generalized to multiple communication failures.

3.1 Feasibility during communication absence

As stated above, during the absence of communication between the two systems s_α and s_β, both affected systems update the last predicted trajectory of the respective neighbour which was successfully transmitted via (10a). However, both systems do not have to follow their last transmitted trajectory, they only need to stay close enough to it that they can not make the other system's optimization problem become infeasible. We restrict the optimized trajectories to be in balls around the last transmitted trajectory. Suppose the last time instant of communication was at Q, we define

$$\tilde{h}_\alpha[k|Q+q] = \max \epsilon \tag{14a}$$

subject to

$$c_{\alpha\beta}(\kappa_\alpha, \kappa_\beta) \le 0 \tag{14b}$$

for all

$$\kappa_\alpha \in S_\epsilon(x_\alpha^{a^q(\mathbf{u}_{\alpha,MPC})}[k|Q+q]) \tag{14c}$$

$$\kappa_\beta \in S_\epsilon(x_\beta^{a^q(\mathbf{u}_{\beta,MPC})}[k|Q+q]) \tag{14d}$$

where $S_r(y) = \{x : \|x-y\|_2 \le r\}$ is the ball of radius r centred at y, and $x_\beta^{a^q(\mathbf{u}_{\beta,MPC})}$ is the last successfully transmitted trajectory (at time Q) updated q times via (10a). The values $\tilde{h}_\alpha[k|Q+q]$ can be interpreted to be the radius of the largest ball around the last transmitted predicted state in which states will not violate the coupling constraint. Since the coupling constraints are equivalent, i.e. $c_{ij}(x_i, x_j) \equiv c_{ji}(x_j, x_i)$, the radii will be the same as well, i.e. $\tilde{h}_\alpha[k|Q+q] = \tilde{h}_\beta[k|Q+q]$.

We now substitute the coupling constraints $c_{\alpha\beta}(x_i^{\mathbf{u}_\alpha}[k|Q+q], x_\beta^p[k|Q+q]) \le 0$ in (3d) at time $l = Q+q$ by

$$\|x_\alpha^{\mathbf{u}_i}[k|Q+q] - x_\alpha^{a^q(\mathbf{u}_{\alpha,MPC})}[k|Q+q]\|_2 \le h_\alpha[k|Q+q] \tag{15}$$

using $h_\alpha[k|Q+q] = \tilde{h}_\alpha[k|Q+q]$ for all $k \in \mathcal{I}(N_h)$ and do the same for s_β to ensure that both optimization problems remain recursively feasible without communicating. Note that (15) does not use the other systems state x_β. Nevertheless, according to the definition of $\tilde{h}_\alpha$ and $\tilde{h}_\beta$ in (14), the substitute constraint (15) guarantees that the original coupling constraint (3d) is not violated. Thus, even without communication between systems s_α and s_β feasibility is guaranteed. This means that when communication is reestablished between the systems s_α and s_β, the original optimization problem (i.e. using the original coupling constraint in (3d)) is still feasible, and hence recursive feasibility of the presented algorithm is guaranteed.

Remark 5. For the case of multiple connection failures for one system we extend (14) slightly by adding all coupling constraints, which have to be satisfied, i.e.

$$\tilde{h}_\alpha[k|Q+q] = \max \epsilon \tag{16a}$$

$$\text{s.t. } c_{\alpha\beta_i}(\kappa_\alpha, \kappa_{\beta_i}) \le 0 \,\forall\, i \in \mathcal{I}_\alpha \tag{16b}$$

for all

$$\kappa_\alpha \in S_\epsilon(x_\alpha^{a^q(\mathbf{u}_{\alpha,MPC})}[k|Q+q]) \tag{16c}$$

$$\kappa_{\beta_i} \in S_\epsilon(x_{\beta_1}^{a^q(\mathbf{u}_{\beta_i,MPC})}[k|Q+q]) \,\forall\, i \in \mathcal{I}_\alpha \tag{16d}$$

where $\mathcal{I}_\alpha$ is the set of all failing communication links at system s_α. However we will substitute all failing coupling constraints by only one, namely (15), with $h_\alpha = \tilde{h}_\alpha$.

3.2 Stability at reconnection

As in the proof of Theorem 3, we consider the Lyapunov function candidate $J_e[l] = J(\mathbf{x}^{\mathbf{u}_{MPC}}[l], \mathbf{u}_{MPC}[l])$ as defined in (11), but with $\mathbf{x}_\beta^p[Q+q] = \mathbf{x}_\beta^{a^q(\mathbf{u}_{\beta,MPC})}[Q+q]$ in J_α and $\mathbf{x}_\alpha^p[Q+q] = \mathbf{x}_\alpha^{a^q(\mathbf{u}_{\alpha,MPC})}[Q+q]$ in J_β, for all $q \ge 1$ until the communication between systems s_α and s_β is reestablished. Note that with this definition, inequality (12) is satisfied as long as all communication links, apart from the one between s_α and s_β, are working. Now suppose that at some time $Q+N_l$ with $N_l \ge 2$, connection between the systems s_α and s_β is reestablished. At this time instant the value of J_α could increase since s_α was previously using the latest successfully communicated state sequence (transmitted at time Q), i.e. $\mathbf{x}_\beta^p[Q+k] = \mathbf{x}_\beta^{a^k(\mathbf{u}_{\beta,MPC})}[Q+k]$ for $0 \le k \le N_l - 1$, and will now update to $\mathbf{x}_\beta^p[Q+N_l] = \mathbf{x}_\beta^{\mathbf{u}_{\beta,MPC}}[Q+N_l]$, which is the latest predicted trajectory for s_β at $Q+N_l$. While communication is working, recall that the computation of δ_β in (9) during the s_α optimization step in the algorithm prevents input sequences to be implemented if they increase the sum of all objective functions. On the other hand, while communication is absent, δ_β and δ_α can not be exchanged, so that the update of $\mathbf{x}_\beta^p[Q+N_l]$ to $\mathbf{x}_\beta^{\mathbf{u}_{\beta,MPC}}[Q+N_l]$ could increase the value of J_α compared to its value when using $\mathbf{x}_\beta^{a^{N_l}(\mathbf{u}_{\beta,MPC})}[Q+N_l]$ instead of $\mathbf{x}_\beta^{\mathbf{u}_{\beta,MPC}}[Q+N_l]$ by δ_β, i.e. by

$$\left| \sum_{k=0}^{N_h-1} L_{\alpha\beta}(x_\alpha^{\mathbf{u}_{\alpha,MPC}}[k|Q+N_l], x_\beta^{\mathbf{u}_{\beta,MPC}}[k|Q+N_l]) \right.$$

$$-L_{\alpha\beta}(x_\alpha^{\mathbf{u}_\alpha,MPC}[k|Q+N_l], x_\beta^{a^{N_l}(\mathbf{u}_{\beta,MPC})}[k|Q+N_l])\Bigg|. \tag{17}$$

The value of (17) can be bounded by using the triangle inequality and by using the Lipschitz property of $L_{\alpha\beta}$ in the second variable; since the state x_α is in a compact ball we obtain a uniform Lipschitz constant $l_{\alpha\beta}^k$ depending on $x_\beta^{a^{N_l}(\mathbf{u}_{\beta,MPC})}[k|Q+N_l]$ and on $h_\beta[k|Q+N_l]$. So that the bound is termwise given by

$$\begin{aligned}&\Big|L_{\alpha\beta}(x_i^{\mathbf{u}_i,MPC}[k|Q+N_l], x_\beta^{\mathbf{u}_\beta,MPC}[k|Q+N_l])\\ &\quad -L_{\alpha\beta}(x_i^{\mathbf{u}_i,MPC}[k|Q+N_l], x_\beta^{a^{N_l}(\mathbf{u}_{\beta,MPC})}[k|Q+N_l])\Big|\\ &\leq l_{\alpha\beta}^k \underbrace{\left\|x_\beta^{\mathbf{u}_\beta,MPC}[k|Q+N_l] - x_\beta^{a^{N_l}(\mathbf{u}_{\beta,MPC})}[k|Q+N_l]\right\|_2}_{\leq h_\beta[k|Q+N_l]}.\end{aligned} \tag{18}$$

This means that each of the values $h_\beta[k|Q+N_l]$ in (15) can be used to bound the maximal deviation of one term in (17). The analogous holds for J_β. Note that even though $l_{\alpha\beta}^k$ depends on the state of s_β and on h_β, we could easily obtain a conservative upper bound

$$l_{\alpha\beta} = \max_{k\in\mathcal{I}(N_h)} l_{\alpha\beta}^k. \tag{19}$$

In total, the sum over the prediction horizon (17) is bounded by

$$(17) \leq \sum_{k=0}^{N_h-1} l_{\alpha\beta}^k \cdot h_\beta[k|Q+N_l] \tag{20}$$

As the values $h_\beta[k|Q+N_l]$ are design parameters, we can, if necessary, use a smaller value than $\tilde{h}_\beta[k|Q+N_l]$ (which was needed in order to ensure feasibility) in order to keep the maximal possible increase of the objective function after reconnection small enough. Furthermore, as we assume that the duration of communication failure N_l is a priori unknown, the necessary values of h_β have to be computed as soon as the communication fails, i.e. at time Q. We compute them in such a way that the possible increase of J_α at time $Q+N_l$ is smaller than the guaranteed decrease of J_β at time Q and vice versa. Namely, we have to ensure that

$$\begin{aligned}&\sum_{k=0}^{N_h-1} l_{\alpha\beta}^k \cdot h_\beta[k|Q+k]\\ &\leq (1-\epsilon)\Bigg(L_{\beta\beta}(x_\beta^{\mathbf{u}_\beta,MPC}[0|Q], u_{\beta,MPC}[0|Q])\\ &\qquad + \sum_{s_j\in\mathcal{N}_\beta} L_{\beta j}(x_\beta^{\mathbf{u}_\beta,MPC}[0|Q], x_j^p[0|Q])\Bigg)\end{aligned} \tag{21}$$

for all $k \geq 1$ and some $0 < \epsilon < 1$. If this is satisfied, it follows that

$$J_e[l] \leq J_e[Q] - \epsilon \sum_{k=Q}^{l-1} \sum_{s_i\in V} L_{ii}(x_i^{\mathbf{u}_i,MPC}[0|k], u_{i,MPC}[0|k]) \tag{22}$$

for all $l \geq Q+1$, from where again, just like in (12), asymptotic stability follows. One possible way to ensure (21) is to choose

$$h_\beta[k|Q+q] = \min\{\tilde{h}_\beta[k|Q+q], \frac{L_{\beta 0}}{l_{\alpha\beta}^k N_h}\}, \tag{23}$$

for all $k \in \mathcal{I}(N_h)$ and all $q \in \mathcal{I}(N_l)$, where $L_{\beta 0}$ denotes the right hand side of (21).

Note that both $L_{\beta 0}$ and the values $\tilde{h}_\beta[k|Q+q]$, appearing on the right hand side of (23), can be determined by system s_β. Furthermore, the Lipschitz constants $l_{\alpha\beta}^k$ can also be determined by system s_β if for example it knows the function $L_{\alpha\beta}$, or if a (global) Lipschitz constant is exchanged between the systems at the beginning. Hence the values $h_\beta[k|Q+q]$ can be determined locally by system s_β (i.e. without further communication) as soon as communication fails, as required.

With the previous analysis, we have derived

Theorem 6. Assume that at time $l = 0$, there is no communication failure and that the algorithm is initially feasible. If the communication link between two systems s_α and s_β fails at some time $Q+1$, recursive feasibility and asymptotic stability of the closed-loop system with respect to the set points $\hat{x}_i$ is maintained if the coupling constraint $c_{\alpha\beta}$ in (3d) (and similarly $c_{\beta\alpha}$) is substituted by (15) with $h_\alpha[k|Q+q]$ satisfying

$$h_\alpha[k|Q+q] \leq \tilde{h}_\alpha[k|Q+q] \tag{24}$$

as well as (21) for some $0 < \epsilon < 1$ and for all $k \in \mathcal{I}(N_h)$ and all $q \in \mathcal{I}(N_l)$.

Remark 7. Handling multiple connection failures is straight forward but not completely obvious. If for a single system connections fail sequentially, i.e. the connection to s_{β_1} fails at Q and the connection to s_{β_2} fails at $Q+q$ and so on, we can treat each failing connection individually with the presented method, using for $o \geq 0$ the minimum of the two values $h_\beta[k|Q+q+o]$ computed at time Q and $Q+q$ for the two communication failures, respectively. If multiple connections fail at one time step, i.e. system s_α loses contact to both s_{β_1} and s_{β_2} after time step Q, then the radii h_α have to satisfy:

$$\begin{aligned}&\left(\sum_{k=0}^{N_h-1}\sum_{\beta_i} l_{\beta_i\alpha}^k h_\alpha[k|Q+N_l]\right)\\ &\leq (1-\epsilon)\Bigg(L_{\alpha\alpha}(x_\alpha^{\mathbf{u}_\alpha,MPC}[0|Q], u_{\alpha,MPC}[0|Q])\\ &\qquad + \sum_{s_j\in\mathcal{N}_\alpha} L_{\alpha j}(x_\alpha^{\mathbf{u}_\alpha,MPC}[0|Q], x_j^p[0|Q])\Bigg).\end{aligned} \tag{25}$$

Note that the radii h_α in (25) can again be computed locally, similar to what was discussed above in case of one communication failure.

4. SIMULATION RESULTS

We present a simple integrator example in a collision avoidance setup. The dynamics we use are:

$$x_i[k+1] = \begin{pmatrix} p_{1i}[k+1] \\ p_{2i}[k+1]\end{pmatrix} = \begin{pmatrix} p_{1i}[k] + u_{1i}[k] \\ p_{2i}[k] + u_{2i}[k]\end{pmatrix}, \tag{26}$$

the objective consists of the quadratic self associated stage costs

$$L_{ii}(x_i, u_i) = \|x_i - \hat{x}_i\|_2^2 + \|u_i\|_2^2 \tag{27a}$$

the coupled stage costs with compact support

$$L_{ij}(x_i, x_j) = 20 \exp\left(\frac{\|\hat{x}_i - \hat{x}_j\|_\infty}{\|x_i - x_j\|_2^2 - \|\hat{x}_i - \hat{x}_j\|_\infty} + 1\right) \tag{27b}$$

for $\|x_i - x_j\|_2^2 < \|\hat{x}_i - \hat{x}_j\|_\infty$ and zero elsewhere, and the quadratic, decoupled terminal costs

$$F_{ii}(x_i) = \|x_i - \hat{x}_i\|_2^2 . \tag{27c}$$

The constraints are given by decoupled box constraints

$$c_{ii}(x_i) = \|x_i\|_\infty - 10, \tag{28a}$$

the quadratic coupling constraints

$$c_{ij}(x_j) = 1 - \|x_i - x_j\|_2 , \tag{28b}$$

the quadratic, decoupled terminal constraints

$$d_{ii}(x_i) = \|x_i - \hat{x}_i\|_2 - \sqrt{2}, \tag{28c}$$

and the box input constraints

$$U_i = \{u \in \mathbb{R}^2 : \|u\|_\infty \leq 1\}. \tag{28d}$$

We use four systems $s_1, \ldots, s_4$. As terminal controllers we use

$$k_i^{loc}(x_i) = \begin{pmatrix} \text{sign}(\hat{x}_{1i} - x_{1i}) \min\{1, |\hat{x}_{1i} - x_{1i}|\} \\ \text{sign}(\hat{x}_{2i} - x_{2i}) \min\{1, |\hat{x}_{2i} - x_{2i}|\} \end{pmatrix}, \tag{29}$$

for initial states $x_1[0] = (-7, 6.5)^T$, $x_2[0] = (-7, -6)^T$, $x_3[0] = (5.5, -6)^T$, $x_4[0] = (6, 6.5)^T$ and set points $\hat{x}_1 = (2, -1.5)^T$, $\hat{x}_2 = (1.5, 1.5)^T$, $\hat{x}_3 = (-2.5, 2)^T$ and $\hat{x}_4 = (-1.5, -1)^T$.

The neighbourhoods are such, that all systems communicate with each other.

After one iteration the communication between s_1 and s_2 fails for eight samples, i.e. for $l = 2, \ldots, 10$ there is no communication between s_1 and s_2.

The results of the simulation is shown in figure 1. We can see, that the deviation between the predicted trajectory at time $Q = 2$ and the actual state trajectory is minimal and we conclude from that, that the additional restriction imposed to guarantee asymptotic stability does not affect the optimization's overall behaviour in a severe way. Even while the restrictions are active, the actual trajectory, i.e. the result of the optimized solution, is not using all its freedom.

Remark 8. Since (27b) is defined to be $L_{ij}(x_i, x_j) = L_{ji}(x_j, x_i)$ for all $s_i, s_i \in V$, we can avoid the intermediate exchange of trajectories to compute δ_i in (9) in the algorithm as all neighbouring systems $s_j \in \mathcal{N}_i$ can compute the value δ_i by themselves.

5. CONCLUSION

This paper proposes a simple modification of an existing DMPC algorithm to make it robust with respect to communication failure. Future work might reduce the conservatism introduced through (21) and might even extend results to couplings of multiple systems, i.e. no pairwise coupling.

REFERENCES

A. Alessio and A. Bemporad. Stability conditions for decentralized model predictive control under packet drop communication. In *American Control Conference, 2008*, pages 3577–3582, 2008.

P. D. Christofides, R. Scattolini, D. Muñoz de la Peña, and J. Liu. Distributed model predictive control: A tutorial review and future research directions. *Computers & Chemical Engineering*, 51:21–41, 2013.

M. D. Doan, T. Keviczky, and B. De Schutter. A distributed optimization-based approach for hierarchical MPC of large-scale systems with coupled dynamics and constraints. In *Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference*, pages 5236–5241, 2011.

W.B. Dunbar. Distributed receding horizon control of cost coupled systems. In *Decision and Control, 2007 46th IEEE Conference on*, pages 2510–2515, 2007.

M. Heidarinejad, J. Liu, D. Muñoz de la Peña, J. F. Davis, and P. D. Christofides. Handling communication disruptions in distributed model predictive control. *Journal of Process Control*, 21(1):173 – 181, 2011.

D.Q. Mayne, J.B. Rawlings, C.V. Rao, and P.O.M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789 – 814, 2000.

Matthias A. Müller, Marcus Reble, and Frank Allgöwer. Cooperative control of dynamically decoupled systems via distributed model predictive control. *International Journal of Robust and Nonlinear Control*, 22(12):1376–1397, 2012.

R. Olfati-Saber, J.A. Fax, and R.M. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

A. Richards and J. P. How. Robust distributed model predictive control. *International Journal of Control*, 80 (9):1517–1531, 2007.

Riccardo Scattolini. Architectures for distributed and hierarchical model predictive control – a review. *Journal of Process Control*, 19(5):723 – 731, 2009.

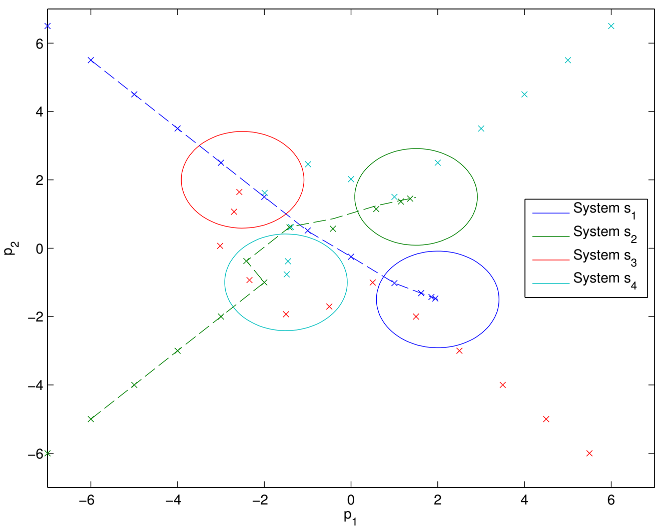

Fig. 1. Terminal regions are depicted by circles, the Xs mark the the trajectory of the closed loop systems, whereas the dashed lines depict the predicted trajectories of system s_1 and system s_2 respectively at the time instant of last communication. System s_2 deviates from the previously predicted trajectory notably, while system s_1 follows it's previously predicted trajectory almost perfectly.