

A distributed NMPC scheme without stabilizing terminal constraints

Lars Grüne and Karl Worthmann

Abstract We consider a distributed NMPC scheme in which the individual systems are coupled via state constraints. In order to avoid violation of the constraints, the subsystems communicate their individual predictions to the other subsystems once in each sampling period. For this setting, Richards and How have proposed a sequential distributed MPC formulation with stabilizing terminal constraints. In this paper we show how this scheme can be extended to MPC without stabilizing terminal constraints or costs. We show theoretically and by means of numerical simulations that under a suitable controllability condition stability and feasibility can be ensured even for rather short prediction horizons.

1 Introduction

In this paper we consider a distributed nonlinear model predictive control (NMPC) algorithm for systems which are coupled via state constraints. NMPC is a controller design method which relies on the online solutions of optimal control problems on finite optimization horizons in each sampling period. In a distributed setting, the solution of this optimal control problem is distributed among the individual systems. This can be done in various ways, see [12, Chapter 6] or [15] for an overview. One way is to formulate the optimization objective in a centralized way and to solve this problem in a distributed way in each sampling period. The necessary splitting of the optimization problem can be obtained in various ways which under suitable assumptions guarantee that the performance of the distributed controller is similar to that of a centralized controller; examples can be found, e.g., in [4] or [12, Chapter 6]. The drawback of this method — which is usually called cooperative control —

Lars Grüne and Karl Worthmann,
Mathematical Institute, University of Bayreuth, 95440 Bayreuth, Germany, e-mail: lars.gruene,
karl.worthmann@uni-bayreuth.de

is that it requires numerous information exchanges between the individual systems during the iterative optimization procedure in each sampling interval.

A less demanding approach from the communication point of view is non-cooperative control, in which some information from the other systems is taken into account when a system performs its optimization but in which the optimization objectives of the individual systems are independent from each other. It is known that for this setting a solution close to the central optimum can no longer be expected; rather, the best one can get is a Nash equilibrium, see [12, Chapter 6]. However, under suitable conditions the resulting closed loop may still be stable and maintain the imposed coupling constraints. This is the situation we investigate in this paper. More precisely, we consider a specific non-cooperative distributed NMPC algorithm proposed by Richards and How [13, 14] in which each system sends information about its predicted future states once in each sampling period. Via a suitable sequential ordering of the individual optimizations it is then ensured that the coupling state constraints are maintained whenever the optimization problems are feasible, i.e., when optimal solutions exist. Clearly, requiring a strict sequential order is a drawback of this approach which we will attempt to relax in future research. Still, the numerical effort of this scheme is already significantly lower than for a centralized solution of the optimization problem, cf. the discussion after Algorithm 3.1, below.

In a stabilization setting, the optimal control problem to be solved online in the NMPC iteration usually minimizes the distance to the desired equilibrium. Often, additional stabilizing terminal constraints and costs are imposed in order to ensure asymptotic stability of the resulting closed loop. This means that the optimization on the finite horizon in each sampling instant is performed over those trajectories which — at the end of the optimization horizon — end up in the terminal constraint set which is typically a neighborhood of the equilibrium to be stabilized. These terminal constraints also play a vital role for ensuring both stability and feasibility in the scheme of Richards and How. In certain situations, however, imposing terminal constraints has the significant drawback that rather long optimization horizons are needed in order to ensure the existence of trajectories which end up in the terminal constraint sets. Furthermore, stabilizing terminal constraints may have negative effects on the performance of the scheme, see, e.g., [7, Section 8.4]. As we will see in the detailed description in Section 3, in the distributed setting the terminal constraint formulation has the additional drawback that possible conflicts between the individual systems, i.e., violations of the coupling state constraints, have to be resolved in an initialization step.

The contribution of this paper is to give sufficient conditions under which we can ensure stability and feasibility without stabilizing terminal constraints. In the non-distributed setting, several approaches for this purpose have been developed, e.g., in [5, 6, 8, 9]. Here we use the approach developed in [6, 8] which relies on an asymptotic controllability assumption taking into account the stage cost of the finite horizon optimal control problems. We will develop an extension of this condition to the distributed setting and we will verify that this condition holds for a simple test example of moving agents in a plane where the coupling constraints are formulated in order to avoid collisions between the agents. Numerical simulations

for this example illustrate that with this scheme stability can be achieved with short optimization horizons and that this scheme allows to resolve conflicts between the individual systems once they become “visible”, i.e., at the runtime of the system rather than in an initialization step.

The paper is organized as follows. In Section 2 we describe the problem formulation and in Section 3 we present the algorithm of Richards and How [13, 14] and discuss its main features. In Section 4 we recall the controllability based stability analysis for NMPC schemes from [6, 8]. Section 5 contains the main result of this paper, i.e., a distributed version of this controllability condition and the corresponding stability result. In Section 6 we investigate a simple test example theoretically and numerically. Section 7 concludes the paper and presents some ideas for future extensions of our main result.

2 Problem setup and preliminaries

We consider $P \in \mathbb{N}$ control systems described by the discrete time dynamics

$$x_p(k+1) = f_p(x_p(k), u_p(k)) \quad (1)$$

for $p = 1, \dots, P$, with $x_p(k) \in X_p$, $u_p(k) \in U_p$ and $f_p : X_p \times U_p \rightarrow X_p$, where X_p are arbitrary metric spaces and U_p are sets of admissible control values for $p = 1, \dots, P$. The solution of (1) for initial value $x_p(0) = x_p^0$ and control sequence $u_p(k) \in U_p$, $k = 0, 1, 2, \dots$ will be denoted by $x_p^u(k, x_p^0)$, i.e., we will omit the subscript p in u_p in order to simplify the notation. The combined state space of all systems will be denoted by

$$X = X_1 \times \dots \times X_P.$$

Our goal is to stabilize each system at a desired equilibrium point $x_p^* \in X_p$. This means, we are looking for feedback controllers $\mu_p(x_p(k), I_p(k)) \in U_p$ which render the respective equilibria asymptotically stable. Here the additional argument $I_p(k)$ of the controller μ_p denotes information from the other systems. We assume that for the purpose of exchanging such information the individual systems can communicate over a network with negligible delay. The precise definition of $I_p(k)$ and the controller μ_p are given in Definition 2.3 and Formula (5), below. The closed loop solutions of (1) with controller μ_p , i.e., the solutions of

$$x_p(k+1) = f_p(x_p(k), \mu_p(x_p(k), I_p(k))) \quad (2)$$

will be denoted by $x_p(k)$, i.e., in order to simplify the notation we will not explicitly include the controller μ_p , the initial value $x_p(0)$ and the additional information I_p in the notation.

Beyond ensuring stability, we want to design the controllers such that the combined state $x(k) = (x_1(k), \dots, x_P(k))$ of the closed loop systems satisfies state constraints of the form

$$x(k) = (x_1(k), \dots, x_P(k)) \in \mathbb{X} \subseteq X, \quad (3)$$

i.e., the state constraints are defined via a state constraint set \mathbb{X} . Note that these constraints induce a coupling between the — otherwise independent — systems which induces the need for passing information $I_p(k)$ between the subsystems.

Example 2.1. As an example which will be used in order to illustrate our concepts throughout this paper we consider a very simple model of $p = 1, \dots, P$ autonomous agents moving in the plane¹ \mathbb{R}^2 with state $x_p = (x_{p,1}(k), x_{p,2}(k))^T \in X_p = \mathbb{R}^2$, control $u_p \in U_p = [-\bar{u}, \bar{u}]^2 \subset \mathbb{R}^2$ for some $\bar{u} > 0$ and dynamics

$$x_p(k+1) = x_p(k) + u_p(k).$$

Thinking of $x_p(k)$ as the position of the individual agent in the plane, the state constraints can be used in order to avoid collisions of the agents. To this end, for some desired distance $\delta > 0$ we define

$$\mathbb{X} := \{(x_1, \dots, x_P)^T \in \mathbb{R}^{2P} \mid \|x_{p_1} - x_{p_2}\| \geq \delta \text{ for all } p_1, p_2 = 1, \dots, P \text{ with } p_1 \neq p_2\},$$

where $\|\cdot\|$ denotes an arbitrary norm in \mathbb{R}^2 . If we use a specific norm in the subsequent computations then this will always be explicitly stated.

Clearly, in order to be able to maintain the state constraints in closed loop, i.e., to avoid collisions in the example, the individual controllers need to have some information about the other systems and for this purpose we will use the so far undefined information $I_p(k)$. In order to define what kind of information $I_p(k)$ the systems should exchange, we first need to specify the control algorithm we are going to use. In this paper we propose to use a model predictive (or receding horizon) control approach. To this end, at each time instant k for its current state $x_p(k)$ each agent solves the optimal control problem

$$\text{minimize } J_p^N(x_p^0, u_p) = \sum_{j=0}^{N-1} \ell_p(x_p^u(j), x_p^0, u_p(j)) \quad \text{with initial value } x_p^0 = x_p(k) \quad (4)$$

over all admissible control sequences $u_p(\cdot) \in U_p^{N,ad}(k, x_p^0, I_p(k)) \subseteq U_p^N$ on the optimization horizon $N \geq 2$, where the set of admissible control sequences $U_p^{N,ad}$ will be defined in Definition 2.3, below. Here ℓ_p is a stage cost function which penalizes the distance of the state from the equilibrium and the control effort. For instance, ℓ could be $\ell_p(x_p, u_p) = \|x_p - x_p^*\| + \lambda \|u_p\|$ or $\ell_p(x_p, u_p) = \|x_p - x_p^*\|^2 + \lambda \|u_p\|^2$, where $\lambda > 0$ is a weight parameter.

We denote the optimal control sequence for (4) by $u_p^{*,k}(0), \dots, u_p^{*,k}(N-1)$ and the corresponding predicted optimal trajectory by $x_p^{u^{*,k}}(0), \dots, x_p^{u^{*,k}}(N-1)$. According to the usual receding horizon construction, the value of the MPC controller is given by the first element of the optimal control sequence $u_p^{*,k}(0)$.

¹ The example could be extended to arbitrary dimensions but for simplicity of exposition we stick to the planar case in this paper.

In order to define this MPC feedback law in a rigorous way, we need to define the set of admissible control sequences in the optimization (4) for the p -th system. To this end, we make use of the following definition.

Definition 2.2. (i) For an index set $\mathcal{P} = \{p_1, \dots, p_m\} \subseteq \{1, \dots, P\}$ with $m \in \mathbb{N}$, $m \leq P$ we define the set of *partial states* as

$$X_{\mathcal{P}} := X_{p_1} \times \dots \times X_{p_m}.$$

Elements of $X_{\mathcal{P}}$ will be denoted by $x_{\mathcal{P}} = (x_{p_1}, \dots, x_{p_m})$. The *partial state constraint set* $\mathbb{X}_{\mathcal{P}} \subset X_{\mathcal{P}}$ is defined as

$$\mathbb{X}_{\mathcal{P}} := \{x_{\mathcal{P}} \in X_{\mathcal{P}} \mid \text{there is } \tilde{x} \in \mathbb{X} \text{ with } \tilde{x}_{p_i} = x_{p_i} \text{ for } i = 1, \dots, m\}.$$

(ii) Given an index set \mathcal{P} , an element $x_{\mathcal{P}} \in X_{\mathcal{P}}$, an element $x_p \in X_p$ with $p \notin \mathcal{P}$ and a subset $\mathcal{Q} = \{q_1, \dots, q_l\} \subset \mathcal{P}$ we write

$$(x_p, (x_{\mathcal{Q}})_{\mathcal{Q}}) := (x_p, x_{q_1}, \dots, x_{q_l}) \in X_{\{p\} \cup \mathcal{Q}}.$$

The admissible control sequences over which we optimize in (4) are now defined via the information available from the other agents according to the following definition.

Definition 2.3. (i) We assume that at time instant k when optimizing (4) for $x_p^0 = x_p(k)$ the p -th agent knows prediction sequences $x_q^{k_q}(\cdot) = (x_q^{k_q}(0), \dots, x_q^{k_q}(N-1))$ for $q \in \{1, \dots, P\} \setminus \{p\}$ computed at time instant $k_q \leq k$ from the other agents. We define

$$I_p(k) := \{(k_q, x_q^{k_q}(\cdot)) \mid q \in \{1, \dots, P\} \setminus \{p\}\}.$$

Note that $I_p(k)$ lies in the set

$$\mathcal{I}_p := (\mathbb{N}_0 \times X_1^N) \times \dots \times (\mathbb{N}_0 \times X_{p-1}^N) \times (\mathbb{N}_0 \times X_{p+1}^N) \times \dots \times (\mathbb{N}_0 \times X_P^N).$$

(ii) Given a time $k \in \mathbb{N}_0$ and $I_p \in \mathcal{I}_p$ with $k_q \leq k$ for all k_q contained in I_p , we define the set of admissible control sequences for system p at time k as

$$U_p^{N,ad}(k, x_p^0, I_p) := \{u_p(\cdot) \in U_p^N \mid (x_p^u(j, x_p^0), (x_q^{k_q}(j+k-k_q))_{\mathcal{Q}_p(k,j)}) \in \mathbb{X}_{\{p\} \cup \mathcal{Q}_p(k,j)} \text{ for all } j = 0, \dots, N-1\}$$

with

$$\mathcal{Q}_p(k, j) = \{q \in \{1, \dots, P\} \setminus \{p\} \mid j+k-k_q \leq N-1\}.$$

The trajectories $x_p^u(\cdot, x_p^0)$ for $u \in U_p^{N,ad}(k, x_p^0, I_p)$ are called admissible trajectories.

In words, this definition demands that the minimization of (4) is performed over those trajectories which satisfy the state constraints together with the known predictions from the other systems for $j = 0, \dots, N-1$.

The resulting feedback law μ_p thus depends on the current state $x_p(k)$ of the p -th closed loop system and on the other systems' predictions $x_q^{k_q}(\cdot)$, $q \neq p$ available at

time k . For $I_p(k) \in \mathcal{I}_p$ the resulting MPC controller is hence given by the map

$$\mu_p : (x_p(k), I_p(k)) \mapsto u_p^{*,k}(0), \quad (5)$$

where $u_p^{*,k}(\cdot)$ is the optimal control sequence minimizing (4). For later use we define the associated optimal value function as

$$V_p^N(x_p^0, I_p) := \min_{u_p \in U_p^{N,ad}(k, x_p^0, I_p)} J_p^N(x_p^0, u_p).$$

In order not to overload the notation it does not reflect the implicit k -dependence of μ_p and V_p^N . Moreover, for simplicity of exposition, throughout the paper we assume that the minimum of this expression exists whenever $U_p^{N,ad}(k, x_p^0, I_p) \neq \emptyset$.

The important questions to be analyzed for this system are the following:

- do the resulting closed loop systems (2) maintain the state constraints (3)?
- are the optimization problems feasible in each step, i.e., is the set of admissible control sequences $U_p^{N,ad}(k, x_p^0, I_p(k))$ in the minimization of (4) non empty?
- is the closed loop system (2) asymptotically stable; in particular, do the trajectories $x_p(k)$ converge to the fixed points x_p^* as $k \rightarrow \infty$?

These are the questions we want to investigate in this paper. Clearly, the precise way of how the information $I_p(k)$ is constructed is crucial for answering these questions. To this end, in the following section we investigate an algorithm in which the construction of the sets $I_p(k)$ implies that feasibility is sufficient for maintaining the state constraints, cf. Proposition 3.2.

3 The scheme of Richards and How

In this section we define how the information $I_p(k)$ is constructed and according to which schedule the information is passed from one system to the others. To this end, we use the sequential scheme introduced by Richards and How in [13, 14]. It should be noted that the general setting in these references is different from ours: on the one hand, only linear dynamics are considered in these references, on the other hand, perturbations are explicitly included in the models considered in [13, 14] and the MPC scheme is designed to be robust against perturbations.

The main idea of the way the distributed optimization takes place, however, is independent from these details. Using the notation introduced in the last section, this idea is described in the following algorithm.

Algorithm 3.1 Let $(x_1(0), \dots, x_p(0)) \in \mathbb{X}$ be given initial values.

(0) **Initialization for $k = 0$.**

Find control sequences $u_p \in U_p^N$ such that the corresponding trajectories satisfy

$$(x_1^u(j, x_1(0)), \dots, x_p^u(j, x_p(0))) \in \mathbb{X} \quad \text{for } j = 0, \dots, N-1. \quad (6)$$

for $p = 1, \dots, P$:

Set $k_p = 0$, $x_p^{k_p}(j) = x_p^u(j)$ for $j = 0, \dots, N-1$ and send $(k_p, x_p^{k_p}(\cdot))$ to all other systems

Apply the control value $\mu_p(x_p^0) = u_p(0)$ in the first step.

end of p -loop

(1) **Control loop for $k \geq 1$.**

for $k = 1, 2, \dots$:

for $p = 1, \dots, P$:

set

$$I_p(k) := ((k, x_1^k(\cdot)), \dots, (k, x_{p-1}^k(\cdot)), (k-1, x_{p+1}^{k-1}(\cdot)), \dots, (k-1, x_P^{k-1}(\cdot)))$$

and minimize (4) for $x_p^0 = x_p(k)$ with respect to $u_p \in U_p^{N, ad}(k, x_p^0, I_p(k))$.

Denote the resulting optimal control by $u_p^{*,k}$, set $k_p = k$, $x_p^{k_p}(j) = x_p^{u_p^{*,k}}(j)$.

for $j = 0, \dots, N-1$ and send $(k_p, x_p^{k_p}(\cdot))$ to all other systems

Apply the control value $\mu_p(x_p^0, I_p(k)) = u_p^{*,k}(0)$ in the k -th step

end of p -loop

end of k -loop

This scheme is sequential in the sense that in step (1) the individual systems perform their optimization one after the other before the control values are eventually applied in all systems. Note that system p always uses the most recent available predictions of the other systems in order to construct the set of admissible control sequences $U_p^{N, ad}$, i.e., for $q < p$ the predictions x_q^k made at time k are used and for $q > p$ the predictions x_q^{k-1} computed at time instant $k-1$ are used in $I_p(k)$. In case of a large number P of systems this sequential optimization may cause rather long waiting times which may not be available in case of fast sampling. While one goal of future research will thus be to relax the strict sequential structure, see also Section 7, below, we remark that the scheme is well applicable for small values of P and, as pointed out in [14, Section 7], even for large P the scheme considerably reduces the numerical effort compared to a centralized solution of the optimization problem in each time instant.

The main advantage of the sequential scheme is that once the initialization step (0) has been performed successfully, the validity of the state constraints for the closed loop solution follows from feasibility. This is made precise in the following proposition.

Proposition 3.2. *Assume that in Algorithm 3.1 the initialization step (0) is successful in finding $u_p \in U_p^N$ satisfying (6) and that in step (1) the optimal control prob-*

lems are feasible, i.e., that $U_p^{N,ad}(k, x_p(k), I_p(k)) \neq \emptyset$ holds for all $p = 1, \dots, P$ and all $k \geq 1$. Then the closed loop maintains the state constraints (3) for all $k \geq 0$.

Proof. Condition (6) and the definition of μ_p in step (0) immediately imply (3) for $k = 1$. Now we proceed by induction over k . Assume that (3) holds for some $k \geq 1$ and that $U_p^{N,ad}(k, x_p(k), I_p(k)) \neq \emptyset$ holds for all $p = 1, \dots, P$. Then each μ_p defined in step (1) is well defined and the definition of $U_p^{N,ad}(k, x_p(k), I_p(k))$ implies

$$(x_1^{u^{*,k}}(1, x_1(k)), \dots, x_P^{u^{*,k}}(1, x_P(k))) \in \mathbb{X}.$$

By definition of the μ_p and (2) we obtain

$$x_p(k+1) = f_p(x_p(k), \mu_p(x_p(k), I_p(k))) = f_p(x_p(k), u_p^{*,k}(0)) = x_p^{u^{*,k}}(1, x_p(k))$$

for all $p = 1, \dots, P$ and thus

$$(x_1(k+1), \dots, x_P(k+1)) = (x_1^{u^{*,k}}(1, x_1(k)), \dots, x_P^{u^{*,k}}(1, x_P(k))) \in \mathbb{X}.$$

This shows (3) for $k+1$. \square

In order to ensure $U_p^{N,ad}(k, x_p(k), I_p(k)) \neq \emptyset$, in [14] a condition involving terminal constraints sets is used. The following assumption summarizes this condition in our notation and without the additional constructions needed for the robust design in [14].

Assumption 3.3 *There exist closed neighborhoods T_p , $p = 1, \dots, P$ of the equilibria x_p^* satisfying the following conditions.*

- (i) $T_1 \times \dots \times T_P \subset \mathbb{X}$.
- (ii) *On each T_p there exists a stabilizing controller K_p for x_p such that T_p is forward invariant for the closed loop system using K_p .*
- (iii) *The control functions u_p in the initialization step (0) and in the optimization of (4) in step (1) are such that $x_p^u(N, x_p(k)) \in T_p$ holds. In the optimization, this amounts to adding $x_p^u(N, x_p(k)) \in T_p$ as a further condition to the definition of the admissible control sequences $U_p^{N,ad}(k, x_p^0, I_p(k))$.*

The benefit of this condition is that if the computation of u_1, \dots, u_P satisfying (6) in step (0) is successful at time $k = 0$, then $U_p^{N,ad}(k, x_p^0, I_p(k)) \neq \emptyset$ is ensured for all subsequent times $k \geq 1$ and all $p = 1, \dots, P$. In order to see this, consider the control sequence $u_p^{*,k-1}$ from the previous time step $k-1$ in step (1) for $p = 1, \dots, P$. Then the construction of $I_q(k-1)$ for $q > p$ and $I_q(k)$ for $q < p$ ensures $u_p^{*,k-1}(\cdot+1) \in U_p^{N-1,ad}(k, x_p^0, I_p(k))$. Since $x_p^{u^{*,k-1}}(N-1, x_p(k)) = x_p^{u^{*,k-1}}(N, x_p(k-1)) \in T_k$, by setting $u_p(j) = u_p^{*,k-1}(j+1)$ for $j = 0, \dots, N-2$ and $u_p(N-1) = K_p x_p^{u^{*,k-1}}(N-1, x_p(k))$ we obtain $x_p^u(N, x_p(k)) \in T_p$. Since the predictions of all other systems $q \neq p$ also end up in their respective sets T_q and $T_1 \times \dots \times T_P \subset \mathbb{X}$, we obtain $u_p \in U_p^{N,ad}(k, x_p^0, I_p(k))$.

Besides ensuring feasibility, Assumption 3.3 also ensures stability. Indeed, a standard MPC stability proof (cf. [10] or [12, Section 2.4]) shows that under a compatibility condition between the stage cost ℓ_p and a suitably chosen terminal cost which is defined on T_p and added to J^N in (4), the optimal value function V_p becomes a Lyapunov function of the system which proves stability. For this reason, the sets T_p in Assumption 3.3 are usually called *stabilizing terminal constraints*.

In the context of Example 2.1, the stabilizing terminal constraints demand that already in the initialization step (0) we have to plan collision free trajectories for all systems from the initial value $x_p(0)$ to a neighborhood T_p of x_p^* . On the one hand, this implies that we may need to use rather large optimization horizons N if we consider initial conditions $x_p(0)$ far away from the terminal sets T_p . On the other hand, and more importantly in our distributed setting, Assumption 3.3 implies that all conflicts, i.e., possible collisions, until the “safe” terminal constraint sets T_p are reached have to be resolved in the initialization step (0). Although in each iteration in step (1) the optimization algorithm is allowed to replan the trajectory, condition (6) is crucial in order to ensure feasibility for $k = 1$ and thus — via Proposition 3.2 — to ensure that the state constraints are maintained for all $k \geq 1$.

The goal of this paper is now to relax these two drawbacks. While we will keep using Algorithm 3.1, we will not use Assumption 3.3 and in particular we will not require the solutions to end up in terminal constraint sets T_p . The hope is that this will enable us to obtain an MPC scheme which is stable and maintains the state constraints with considerably smaller optimization horizon N and — in the context of Example 2.1 — which is able to solve possible conflicts at the times $k \geq 1$ when they become visible and not necessarily in the initialization step 0.

To this end, in the next section we first revisit a stability condition for NMPC schemes without stabilizing terminal constraints.

4 Stability of NMPC without stabilizing terminal constraints

In this section we recall the stability analysis of NMPC controllers without stabilizing terminal constraints from [6, 8]. We will present the analysis for a single system of type (1). In the subsequent Section 5, we extend these results to our setting with P systems.

Since in this section we deal with a single system of type (1), we will omit the index p in all expressions as well as the dependence of V^N and μ on information from the other systems. Analogous to Definition 2.3, admissibility for a control sequence $u \in U^N$ and an initial value $x^0 \in \mathbb{X}$ means that $u(j) \in U$ and $x^u(j, x^0) \in \mathbb{X}$ for $j = 0, \dots, N-1$, i.e., that the state constraints are maintained. Since in this section we do not consider couplings between different systems, Definition 2.3(ii) simplifies to

$$U^{N,ad}(x^0) := \{u(\cdot) \in U^N \mid x^u(j, x^0) \in \mathbb{X} \text{ for all } j = 0, \dots, N-1\}. \quad (7)$$

We assume that for each $x \in \mathbb{X}$ and each $N \in \mathbb{N}$ this set satisfies $U^{N,ad}(x) \neq \emptyset$ which means that the state constraint set $\mathbb{X} \subset X$ is forward invariant or viable. This assumption provides the easiest way to ensure feasibility of the resulting NMPC scheme and is used here in order to simplify the exposition. If desired, it can be relaxed in various ways, see, e.g., [7, Sections 8.2–8.3] or [11, Theorem 3].

Stability of the NMPC closed loop is established by showing that the optimal value function V^N is a Lyapunov function for the system. More precisely, we aim at giving conditions under which for all $x \in \mathbb{X}$ we can establish the inequalities

$$\alpha_1(\|x - x^*\|) \leq V^N(x) \leq \alpha_2(\|x - x^*\|) \quad (8)$$

and

$$V^N(f(x, \mu(x))) \leq V^N(x) - \alpha \ell(x, \mu(x)) \quad (9)$$

for $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ and $\alpha \in (0, 1]$. Then, under the additional assumption that

$$\alpha_3(\|x - x^*\|) \leq \ell^*(x) \leq \alpha_4(\|x - x^*\|) \quad (10)$$

holds for all $x \in \mathbb{X}$, suitable $\alpha_3, \alpha_4 \in \mathcal{K}_\infty$ and $\ell^*(x) := \min_{u \in U} \ell(x, u)$, we can conclude asymptotic stability as stated by the following theorem.

Theorem 4.1. *Assume that the inequalities (8), (9) and (10) hold for the optimal value function V^N and the stage cost ℓ of the optimal control problem (4) for one system, i.e., for $p = P = 1$. Then the closed loop system (2) with the NMPC feedback (5) is asymptotically stable on \mathbb{X} .*

Proof. The proof follows from by standard Lyapunov function arguments using V^N as a Lyapunov function, see [6, Theorem 5.2]. \square

The inequalities (8) and (9) can be ensured by an asymptotic controllability condition of the equilibrium x^* . Here we work with the special case of exponential controllability, more general versions can be found in [6, 8].

Assumption 4.2 *Given constants $C > 0$, $\sigma \in (0, 1)$, for each $x \in X$ and each $N \in \mathbb{N}$ there exists an admissible control function $u_x \in U^{N,ad}(x)$ satisfying*

$$\ell(x^{u_x}(j, x), u_x(j)) \leq C \sigma^j \ell^*(x)$$

for all $j \in \{0, \dots, N-1\}$ with ℓ^* from (10).

Observe that the controllability condition is defined here in a slightly weaker form than in [6, 8] in the sense that the control function u_x is implicitly allowed to depend on N while in [6, 8] the existence of one u_x for all $N \in \mathbb{N}$ is assumed. However, it is straightforward to see that the weaker condition given here is sufficient for all arguments used in the proofs in these references. Note that the constant $C > 0$ allows for an increase of $\ell(x^{u_x}(j, x), u_x(j))$ for small j before it must eventually decrease. In particular, ℓ does not need to be a control Lyapunov function for the system.

Example 4.3. Consider Example 2.1 with only one system, which in particular implies that the state constraint \mathbb{X} does not include any coupling terms. Instead, we use the state constraint set $\mathbb{X} = [-1, 1]^2$. As stage cost we use $\ell(x, u) = \|x - x^*\|^2 + \lambda \|u\|^2$ for some $x^* \in [-1, 1]$ and some $\lambda \geq 0$. Moreover, let $c := \max_{x \in \mathbb{X}} \|x - x^*\|$ denote the maximal distance in \mathbb{X} from x^* .

We inductively define a control $u \in U^{N,ad}(x)$ by

$$u(k) = \kappa(x^* - x^u(k, x)) \quad \text{with} \quad \kappa = \min\{\bar{u}/c, \rho\}$$

for some design parameter $\rho \in (0, 1)$. Note that the choice of κ implies $u(k) \in [-\bar{u}, \bar{u}]^2$ for $x^u(k, x) \in \mathbb{X}$. Moreover, this definition implies

$$x^u(k+1, x) = x^u(k, x) + \kappa(x^* - x^u(k, x)) \quad (11)$$

and, as a consequence,

$$\|x^u(k+1, x) - x^*\| = (1 - \kappa)\|x^u(k, x) - x^*\|. \quad (12)$$

Due to the convexity of \mathbb{X} and $\kappa \in (0, 1)$, the identity (12) ensures feasibility of $x^u(\cdot)$. Using the definition of $u(k)$ and (12) yields

$$\begin{aligned} \ell(x^u(k, x), u(k)) &= \|x^u(k, x) - x^*\|^2 + \lambda \|u(k)\|^2 \\ &= (1 + \lambda \kappa^2) \|x^u(k, x) - x^*\|^2 \\ &= (1 + \lambda \kappa^2)(1 - \kappa)^{2k} \|x^u(0, x) - x^*\|^2 \\ &= (1 + \lambda \kappa^2)(1 - \kappa)^{2k} \ell^*(x^u(0, x)) \end{aligned}$$

which shows Assumption 4.2 for $u_x = u$ with $C = 1 + \lambda \kappa^2$ and $\sigma = (1 - \kappa)^2$. Analogously, one obtains Assumption 4.2 for $\ell(x, u) = \|x - x^*\| + \lambda \|u\|$ with $C = 1 + \lambda \kappa$ and $\sigma = 1 - \kappa$.

Under Assumption 4.2, the following properties hold.

Lemma 4.4. *Let Assumption 4.2 hold and define*

$$B^N(r) := \sum_{n=0}^{N-1} C \sigma^n r = C \frac{1 - \sigma^N}{1 - \sigma} r.$$

Then for each $x \in \mathbb{X}$ the following properties hold.

(i) *The inequality*

$$V^N(x) \leq J^N(x, u_x) \leq B^N(\ell^*(x)) \quad (13)$$

holds.

(ii) *Let u^* be an optimal control sequence for (4). Then for each $k = 0, 1, \dots, N-2$ the inequality*

$$J^{N-k}(x^{u^*}(k, x), u^*(k+\cdot)) \leq B^{N-k}(\ell^*(x^{u^*}(k, x))) \quad (14)$$

holds.

(iii) Let u^* be an optimal control sequence for (4). Then for each $j = 0, 1, \dots, N-2$ the inequality

$$V^N(x^{u^*}(1, x)) \leq J^j(x^{u^*}(1, x), u^*(1 + \cdot)) + B^{N-j}(\ell^*(x^{u^*}(1 + j, x))) \quad (15)$$

holds.

Proof. (i) This follows immediately from Assumption 4.2.

(ii) This inequality follows from (i) applied to $x = x^{u^*}(k, x)$ using the fact that by the dynamic programming principle tails of optimal trajectories are again optimal trajectories, see [6, Lemma 3.4] for details.

(iii) Follows from the inequality $V^N(x^{u^*}(1, x)) \leq J^N(x^{u^*}(1, x), \tilde{u})$ using the control function

$$\tilde{u}(n) = \begin{cases} u^*(1+n), & n \leq j-1 \\ u_x(n), & n \geq j \end{cases}$$

with u_x from Assumption 4.2 with $x = x^{u^*}(1 + j, x)$ and (i), for details see [6, Lemma 3.5]. \square

Remark 4.5. Lemma 4.4 (i) yields that under Assumption 4.2 the inequalities in (10) imply (8). Indeed, the inequality

$$V^N(x) = J^N(x, u^*) \geq \ell(x, u^*(0)) \geq \ell^*(x) \geq \alpha_3(\|x - x^*\|)$$

implies the lower inequality in (8) with $\alpha_1 = \alpha_3$ and

$$V^N(x) \leq B^N(\ell^*(x)) = C \frac{1 - \sigma^N}{1 - \sigma} \ell^*(x) \leq C \frac{1 - \sigma^N}{1 - \sigma} \alpha_4(\|x - x^*\|)$$

implies the upper inequality in (8) with $\alpha_2 = C \frac{1 - \sigma^N}{1 - \sigma} \alpha_4$.

It remains to establish (9) for which we use Lemma 4.4(ii) and (iii) in the following way.

Proposition 4.6. *Assume Assumption 4.2 and consider $N \geq 1$, a sequence $\lambda_n > 0$, $n = 0, \dots, N-1$, and a value $v > 0$. Let $x \in \mathbb{X}$ and let $u^* \in U^N$ be an optimal control sequence for (4) such that $\lambda_n = \ell(x^{u^*}(n, x), u^*(n))$ holds for $n = 0, \dots, N-1$. Then*

$$\sum_{n=k}^{N-1} \lambda_n \leq B^{N-k}(\lambda_k), \quad k = 0, \dots, N-2 \quad (16)$$

holds. If, furthermore, $v = V^N(x^{u^*}(1))$ holds then

$$v \leq \sum_{n=0}^{j-1} \lambda_{n+1} + B^{N-j}(\lambda_{j+1}), \quad j = 0, \dots, N-2 \quad (17)$$

holds.

Proof. If the stated conditions hold, then λ_n and \mathbf{v} must meet inequalities (14) and (15), which is exactly (16) and (17). \square

The conditions (16) and (17) lead to the following sufficient condition for (9).

Theorem 4.7. *Let $N \geq 1$, assume that Assumption 4.2 holds and that the optimization problem*

$$\alpha := \inf_{\lambda_0, \dots, \lambda_{N-1}, \mathbf{v}} \frac{\sum_{n=0}^{N-1} \lambda_n - \mathbf{v}}{\lambda_0} \quad (18)$$

subject to the constraints (16), (17), and

$$\lambda_0 > 0, \lambda_1, \dots, \lambda_{N-1}, \mathbf{v} \geq 0$$

has an optimal value $\alpha \in (0, 1]$. Then (9) holds for this α for each $x \in \mathbb{X}$.

Proof. The optimization objective in (18) implies that for all values $\lambda_1, \dots, \lambda_{N-1}, \mathbf{v}$ satisfying (16), (17) the inequality

$$\mathbf{v} \leq \sum_{n=0}^{N-1} \lambda_n - \alpha \lambda_0$$

holds. Proposition 4.6 then implies that for each optimal trajectory starting in some arbitrary $x \in \mathbb{X}$ the values $\lambda_n = \ell(x^{u^*}(n, x), u^*(n))$ and $\mathbf{v} = V^N(x^{u^*}(1, x))$ satisfy (16) and (17), which yields

$$\begin{aligned} V^N(x^{u^*}(1, x)) &\leq \sum_{n=0}^{N-1} \ell(x^{u^*}(n, x), u^*(n)) - \alpha \ell(x^{u^*}(0, x), u^*(0)) \\ &= V^N(x) - \alpha \ell(x, u^*(0)). \end{aligned}$$

Since by definition of the MPC feedback law we obtain $\mu(x) = u^*(0)$ and thus $f(x, \mu(x)) = x^{u^*}(1, x)$, this proves (9). \square

The characterization of α via the optimization problem (18) is particularly useful because it admits the following explicit analytic solution.

Theorem 4.8. *Under Assumption 4.2 the optimization problem (18) has the solution*

$$\alpha = 1 - \frac{(\gamma_N - 1) \prod_{k=2}^N (\gamma_k - 1)}{\prod_{k=2}^N \gamma_k - \prod_{k=2}^N (\gamma_k - 1)} \quad \text{with } \gamma_k = C \frac{1 - \sigma^k}{1 - \sigma} \quad (19)$$

for $C > 0$ and $\sigma \in (0, 1)$ from Assumption 4.2. Furthermore, for each pair of values $C > 0$ and $\sigma \in (0, 1)$ the value α in (19) satisfies $\alpha \rightarrow 1$ as $N \rightarrow \infty$.

Proof. Formula (19) follows from [8, Theorem 5.4] and the convergence $\alpha \rightarrow 1$ from [8, Corollary 6.1]. \square

Remark 4.9. An inspection of the proof of [8, Theorem 5.4] shows that some inequalities provided by Lemma 4.4 are not needed in order to prove (19) since in this proof a relaxed problem [8, Problem 5.3] with fewer constraints was used. It turns out that the inequalities not needed in this relaxed problem are exactly (14) for $k = 1, \dots, N-2$ or, equivalently, (16) for $k = 1, \dots, N-2$, see [7, Remark 6.35]. While this has no consequence for the analysis in this section since we get all inequalities in (14) “for free” from Assumption 4.2, this observation will turn out very useful in the next section.

Combining the three Theorems 4.1, 4.7 and 4.8 yields the following corollary.

Corollary 4.10. *Consider a single system of type (1) and the NMPC feedback law (5) for some $N \geq 2$. Let Assumption 4.2 and (10) hold and assume that $\alpha > 0$ holds for α from (19). Then the closed loop system (2) is asymptotically stable on \mathbb{X} .*

Using the convergence $\alpha \rightarrow 1$ for $N \rightarrow \infty$ we can use this corollary in order to conclude that when (10) and Assumption 4.2 hold, then asymptotic stability can be guaranteed for each sufficiently large optimization horizon N . Beyond this asymptotic result, however, the condition $\alpha > 0$ in (19) also gives a useful stability criterion for small optimization horizons N , as the following example shows.

Example 4.11. We reconsider Example 4.3 with $N = 2$. Formula (19) simplifies to $\alpha = 1 - (C + \sigma C - 1)^2$. Since $\kappa \in (0, 1)$, $C = 1 + \lambda \kappa^2$, $\sigma = (1 - \kappa)^2$ we obtain with $\lambda \in (0, 1)$

$$C + \sigma C - 1 = (1 + \lambda \kappa^2)(1 - \kappa^2) \leq (1 + \kappa)(1 - \kappa)^2 = (1 - \kappa^2)(1 - \kappa) < 1$$

which implies $\alpha > 0$. For instance, for $\lambda = 0.1$, $\rho = 0.5$ we obtain $\alpha \approx 0.8102$ or $\alpha \approx 0.9209$ for the Euclidean and the ∞ -norm respectively. This shows that the MPC closed loop is asymptotically stable for $N = 2$ which is the shortest possible optimization horizon given that the sum in (4) only includes the states $x^u(j, x^0)$ for $j = 0, \dots, N-1$.

More complex examples of this kind including infinite dimensional PDE models can be found, e.g., in [6, Sections 6 and 7] or [1, 2, 7]. Finally, we remark that α also allows to estimate the performance of the MPC feedback law μ in terms of an infinite horizon optimization criterion; for details see, e.g., [6, Theorem 4.2].

5 Stability of distributed NMPC without stabilizing terminal constraints

In this section we adapt the results of the previous section to the distributed MPC setting introduced in Section 2 using Algorithm 3.1. The goal is to adapt Assumption 4.2 to the distributed setting. This way we derive a sufficient condition for distributed NMPC without stabilizing terminal constraints which ensures feasibility

of the optimal control problems in Algorithm 3.1(1) — and thus via Proposition 3.2 guarantees that the state constraints are maintained — and stability of the NMPC closed loop. Stability will be guaranteed by showing that each optimal value function V_p^N will satisfy the inequalities (8) and (9), i.e., that each V_p^N is a Lyapunov function for the corresponding system.

Comparing the distributed setting of Section 2 with the non-distributed setting of Section 4, the main difference is that the set of admissible control sequences $U_p^{N,ad}$ in Definition 2.3(ii) changes with time k due to the fact that the information $I_p(k)$ in Algorithm 3.1(1) also changes with time. In contrast to this, the set $U^{N,ad}$ in (7) is constant over time. In order to include the time dependence in the controllability assumption we make use of sets of admissible control sequences according to the following definition.

Definition 5.1. (i) For $m_1 > m_2 \in \mathbb{N}$ and a control sequence $u = (u(0), \dots, u(m_1 - 1)) \in U_p^{m_1}$ we define the restriction

$$u|_{m_2} := (u(0), \dots, u(m_2 - 1)) \in U_p^{m_2}.$$

(ii) A family of sets $W_p^m \subset U_p^m$, $m \in \{1, \dots, N\}$, $N \geq 2$, of admissible control sequences is called *nested* if for all $m_1, m_2 \in \{1, \dots, N\}$ with $m_1 > m_2$ and all $u \in U_p^{m_1}$ the implication

$$u \in W_p^{m_1} \quad \Rightarrow \quad u|_{m_2} \in W_p^{m_2}$$

holds.

(iii) For a nested family of admissible control sequence sets $W_p^m \subset U_p^m$, $m \in \{1, \dots, N\}$, integers $l, m \in \mathbb{N}$, $l + m \leq N$, and a control sequence $u \in W_p^l$ we define

$$W_p[u, l, m] := \{\tilde{u} \in U_p^m \mid (u(0), \dots, u(l-1), \tilde{u}(0), \dots, \tilde{u}(m-1)) \in W_p^{l+m}\}.$$

Recalling that in our setting the admissible control sequences are derived from the state constraint sets \mathbb{X} and the predicted trajectories of the other systems contained in I_p via Definition 2.3(ii), a little computation reveals that for each time instant $k \geq 0$ the sets $W_p^m = U_p^{m,ad}(k, x_p^0, I_p)$, $m \in \mathbb{N}$ are nested and that this choice of W_p^m implies

$$W_p[u, l, m] = U_p^{m,ad}(k+l, x_p^u(l, x_p^0), I_p).$$

Another issue we take into account when adapting Assumption 4.2 is that in the distributed setting it is quite demanding to assume that controllability holds for all possible initial values. Instead, we will formulate the respective condition for fixed initial conditions. The following theorem presents this variant in an abstract setting with nested admissible control sequence sets W_p^m and \tilde{W}_p^m . In the subsequent Theorem 5.3 we will then show how this condition fits into Algorithm 3.1.

Theorem 5.2. Consider some $p \in \{1, \dots, P\}$, two families of nested admissible control sequence sets $W_p^m, \tilde{W}_p^m \subseteq U_p^m$, $m \in \{1, \dots, N\}$ for $N \geq 2$, a point $x_p^0 \in X_p$ and the optimal values

$$V^N := \min_{u_p \in W_p^N} J_p^N(x_p^0, u_p) \quad \text{and} \quad \tilde{V}^N := \min_{\tilde{u}_p \in \tilde{W}_p^N} J_p^N(\tilde{x}_p, \tilde{u}_p)$$

with $\tilde{x}_p = f_p(x_p^0, u_p^*(x_p^0))$ where $u_p^* \in W_p^N$ denotes the optimal control for V^N , i.e., $V^N = J_p^N(x_p^0, u_p^*)$.

For given constants $C > 0$, $\sigma \in (0, 1)$ assume that the following holds.

(i) The inequality $V^N \leq B^N(x_p^0)$ holds for B^N from Lemma 4.4.

(ii) The optimal control $u_p^* \in W_p^N$ satisfies $(u^*(1), \dots, u^*(N-1)) \in \tilde{W}_p^{N-1}$.

(iii) For each $j = 0, \dots, N-2$ there exists $\tilde{u} \in \tilde{W}_p[u^*(1+\cdot), j, N-j]$ with

$$\ell_p(x_p^{\tilde{u}}(s, x_p^{u^*}(1+j, x_p^0)), \tilde{u}(s)) \leq C\sigma^s \ell_p^*(x_p^{u^*}(1+j, x_p^0)), \quad s = 0, 1, \dots, N-j-1.$$

Then the inequality

$$\tilde{V}^N \leq V^N - \alpha \ell_p(x_p^0, u^*(0))$$

holds for α from (19).

Proof. It is sufficient to show (13)–(15) for $x = x_p^0$, $V^N(x) = V^N$ and $V^N(x^{u^*}(1, x)) = \tilde{V}^N$. This implies that $\lambda_n = \ell_p(x^{u^*}(n, x_p^0), u^*(n))$ and $v = \tilde{V}^N$ satisfy (16), (17) and by the same argument as in the proof of Theorem 4.7 we obtain the assertion when we use Theorem 4.8 in order to solve (18). By Remark 4.9 it is sufficient to show (14) for $k = 0$.

In order to prove these inequalities, observe that (13) and (14) for $k = 0$ follow directly from (i). In order to prove (15) we use that (iii) implies

$$J_p^{N-j}(x_p^{u^*}(1+j, x_p^0), \tilde{u}) \leq B^{N-j}(\ell_p^*(x_p^{u^*}(1+j, x_p^0))), \quad j = 0, 1, \dots, N-2 \quad (20)$$

for B^N from Lemma 4.4. Observe that (ii) is needed in order to ensure that $\tilde{W}_p[u^*(1+\cdot), j, N-j]$ in (iii) is well defined.

Now (15) follows from (20) using the inequality

$$V^N(x_p^{u^*}(1, x_p^0)) \leq J_p^j(x_p^{u^*}(1, x_p^0), u^*(1+\cdot)) + J_p^{N-j}(x_p^{u^*}(1+j, x_p^0), \tilde{u}),$$

which holds since (ii) and $\tilde{u} \in \tilde{W}_p[u^*(1+\cdot), j, N-j]$ imply $(u^*(1), \dots, u^*(j), \tilde{u}(0), \dots, \tilde{u}(N-j-1)) \in \tilde{W}_p^N$. \square

The following theorem incorporates this condition into Algorithm 3.1.

Theorem 5.3. Consider Algorithm 3.1 with optimization horizon $N \in \mathbb{N}$ in (4), let $C > 0$ and $\sigma \in (0, 1)$ and assume that the stage costs ℓ_p satisfy (10) for all $p \in \{1, \dots, P\}$ and suitable $\alpha_3, \alpha_4 \in \mathcal{K}_\infty$. Assume that step (0) of the algorithm is successful and denote the resulting control functions by $u_p^{*,0}$. Assume, furthermore, that in step (1) of the algorithm for each $k \geq 1$ and each $p \in \{1, \dots, P\}$ condition (iii) of Theorem 5.2 holds with $u^* = u_p^{*,k-1}$, $x_p^0 = x_p(k-1)$ and

$$\tilde{W}_p^m = U_p^{m,ad}(k, x_p(k), I_p(k)), \quad m = 1, \dots, N.$$

Then, the closed loop solutions maintain the state constraints (3) and there exists $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ such that the optimal value functions V_p^N satisfy

$$\alpha_1(\|x_p(k) - x_p^*\|) \leq V_p^N(x_p(k), I_p(k)) \leq \alpha_2(\|x_p(k) - x_p^*\|) \quad (21)$$

for all $k \geq 1$ and the inequality

$$V_p^N(x_p(k+1), I_p(k+1)) \leq V_p^N(x_p(k), I_p(k)) - \alpha \ell(x_p(k), \mu_p(x_p(k), I_p(k))) \quad (22)$$

holds for α from (19) and all $k \geq 1$.

In particular, if $\alpha > 0$ (which always holds for $N > 0$ sufficiently large) then the V_p^N are Lyapunov functions for the closed loop systems for $k \geq 1$ and thus asymptotic stability of the equilibria x_p^* follows.

Proof. We show that for each $k \geq 2$ the assumptions of Theorem 5.2 hold with $W_p^m = U_p^{m,ad}(k-1, x_p(k-1), I_p(k-1))$, $\tilde{W}_p^m = U_p^{m,ad}(k, x_p(k), I_p(k))$, $x_p^0 = x_p(k-1)$, $\tilde{x}_p = x_p(k)$ and $u^* = u_p^{*,k-1}$. To this end, first observe that in the discussion after Assumption 3.3 we have shown that in step (1) the relation $u_p^{*,k-1}(1 + \cdot) \in U_p^{N-1,ad}(k, x_p(k), I_p(k))$ holds which implies that condition (ii) of Theorem 5.2 is satisfied.

Condition (iii) of Theorem 5.2 holds by assumption and condition (i) of Theorem 5.2 at time k follows from condition (iii) for $j = 0$ at time $k-1$, since $x_p(k) = x_p^{u_p^{*,k-1}}(1, x_p(k-1))$ and W_p^m at time k equals \tilde{W}_p^m at time $k-1$.

Thus, Theorem 5.2 is applicable which proves (22).

Inequality (21) is then obtained with the same arguments as in Remark 4.5. Finally, since the assumed condition (iii) of Theorem 5.2 in particular demands $U_p^{N,ad}(k, x_p(k), I_p(k)) \neq \emptyset$, Proposition 3.2 yields feasibility of the problem and implies that the closed loop solutions satisfy the state constraints (3). \square

The central assumption in this theorem is that condition (iii) of Theorem 5.2 holds. In words, this assumption requires two things: first, $U_p^{N,ad}(k, x_p(k), I_p(k))$ needs to be non empty which means that given the predictions of the other systems x_q^u , $q \neq p$, contained in I_p there is still enough space to “squeeze in” a solution x_p^u . Second, the condition requires that starting from any point on the optimal open loop trajectory from the last time instant, there are solutions which approach the equilibrium x_p^* sufficiently fast in the sense of the controllability assumption. The important fact in this condition is that when the p -th system selects its control it knows the other systems’ predictions. For this reason this rather technical condition can be rigorously verified at least for simple systems, as the example in the following section shows.

Note that even though step (0) remains formally identical to Algorithm 3.1, without the additional terminal condition from Assumption 3.3(iii) and with smaller N it is much easier to satisfy (6). This is illustrated in the numerical simulations at the end of the next section, in which for most of the systems the state constraints only become relevant after several steps of the algorithm.

6 An example

In this section we first verify that Example 2.1 satisfies the conditions of Theorem 5.3 for $P = 2$ under suitable conditions. Afterwards we numerically illustrate the performance of the scheme for this example with $P = 2$ and $P = 4$.

In order to verify the conditions of Theorem 5.3, we consider Example 2.1 with $P = 2$ and show that the conditions hold for $p = 1$ and all initial values x_1^0 which are bounded by $\|x_1^0\| \leq K$ for some $K > 0$. Analogous arguments then show the condition for $p = 2$. Without loss of generality we may assume $x_1^* = 0$. Since a priori it is not clear how the predictions x_2^{*,k_2} contained in $I_1(k)$ defining the sets $U_p^{m,ad}$ in Theorem 5.3 look, we show the stronger property that the conditions hold for $p = 1$ for all possible trajectories x_2^u . The only thing we have to exclude here is that x_2^u stays too close to the equilibrium x_1^* , because then it will never be possible for x_1^u to reach x_1^* without collision and thus to reduce $\ell_1(x_1^u(k), u_1(k))$ to 0. Hence, in what follows we consider all possible trajectories $x_2^u(k)$ which stay outside a neighborhood around $x_1^* = 0$.

We show the following lemma, which implies the conditions of Theorem 5.3 whenever the trajectory x_2^{*,k_2} contained in $I_1(k)$ remains outside the neighborhood with radius $R + \delta$ around x_1^* . In order to streamline the exposition, in the following lemma the norm $\|\cdot\|$ is either the Euclidean or the ∞ -norm. Without loss of generality we furthermore assume $\delta > \bar{u}$; otherwise we can restrict ourselves to smaller control values than actually allowed.

Lemma 6.1. *We consider the stage cost $\ell_1(x_1, u_1) = \|x_1\|^2 + \lambda \|u_1\|^2$ and the state constraint set \mathbb{X} from Example 2.1 for $P = 2$. Given $K > 0$ and $R > \delta > \bar{u}$ there exists $C > 0$, $\sigma \in (0, 1)$ such that for each trajectory $x_2^u(k)$ satisfying $\|x_2^u(k)\|_\infty \geq R + \delta$ or $\|x_2^u(k)\|_2 \geq \sqrt{2}(R + \delta)$ for all $k \in \mathbb{N}_0$ and each initial value x_1^0 with $\|x_1^0\| \leq K$ and $(x_1^0, x_2^u(0)) \in \mathbb{X}$ there exists a control sequence $u_1(k) \in [-\bar{u}, \bar{u}]^2$ with $(x_1^u(k, x_1^0), x_2^u(k)) \in \mathbb{X}$ and*

$$\ell_1(x_1^u(k, x_1^0), u_1(k)) \leq C\sigma^k \ell_1^*(x_1^0) \quad \text{for all } k \in \mathbb{N}_0. \quad (23)$$

Proof. For $x_1^0 \notin T$, the fact that $\|x_2^u(k)\|_\infty \geq R + \delta$, whenever $x_1^0 \in T = [-R, R]^2$ implies that the control $u_1 = u$ constructed in Example 4.3 satisfies (23) for suitable \tilde{C} and σ since the resulting trajectory remains in T and thus $(x_1^u(k), x_2^u(k)) \in \mathbb{X}$ holds for all $k \in \mathbb{N}_0$.

For $x_1^0 \notin T$, Lemma 8.1 applied with $x^u(\cdot) = x_1^u(\cdot)$ and $y(\cdot) = x_2^u(\cdot)$ shows the existence of a constant \bar{k} and a control $u_1 = u$ such that $x_1^u(k^*) \in T$ for a $k^* \leq \bar{k}$, cf. Remark 8.2 for the Euclidean norm. Since $\|u_1(k)\|_\infty$ is bounded by \bar{u} , the trajectory $x_1^u(k, x_1^0)$ from (8.1) is bounded in the ∞ -norm by $K + \bar{k}\bar{u}$ and thus $\ell(x_1^u(k, x_1^0), u_1(k))$, $k = 0, \dots, \bar{k}$ is bounded by some constant L independent of x_1^0 . Using u from Example 4.3 from time k^* on, the resulting overall control sequence satisfies $(x_1^u(k, x_1^0), x_2^u(k)) \in \mathbb{X}$ for all $k \geq 0$,

$$\ell_1(x_1^u(k, x_1^0), u(k)) \leq L \leq L \|x_1^0\|^2 / R^2 = (L/R^2) \ell^*(x_1^0) \leq (L\sigma^{-\bar{k}}/R^2) \sigma^k \ell^*(x_1^0)$$

for $k = 0, \dots, \bar{k}$ and

$$\ell_1(x_1^u(k, x_1^0), u(k)) \leq \tilde{C} \sigma^{k-\bar{k}} \ell_1^*(x_1^u(\bar{k}, x_1^0)) \leq \tilde{C} \sigma^{k-\bar{k}} 2 \ell_1^*(x_1^0),$$

for $k \geq \bar{k}$. Here the last inequality follows from $\ell^*(x_1^u(\bar{k}, x_1^0)) = \|x_1^u(\bar{k}, x_1^0)\|^2 \leq 2 \|x_1^u(\bar{k}, x_1^0)\|_\infty^2 \leq 2 \|x_1^0\|_\infty^2 \leq 2 \|x_1^0\|^2 = 2 \ell_1^*(x_1^0)$. Together this yields

$$\ell_1(x_1^u(k, x_1^0), u(k)) \leq \max\{L\sigma^{-\bar{k}}/R^2, 2\tilde{C}\sigma^{-\bar{k}}\} \sigma^k \ell_1^*(x_1^0)$$

and thus (23) with $C = \max\{L\sigma^{-\bar{k}}/R^2, 2\tilde{C}\sigma^{-\bar{k}}\}$.

Remark 6.2. (i) The construction used in the proof of Lemma 8.1 and, thus, in this example heavily relies on the knowledge of x_2^u . Indeed, without this knowledge the construction of k^* and u_1 would not be possible. Hence, the communication of I_p in the algorithm is crucial for ensuring the conditions of Theorem 5.3 in this example.

(ii) The additional condition $\|x_2^u(k) - x_1^*\|_\infty \geq R + \delta$ (and vice versa for $x_1^u(k) - x_2^*$) could be ensured by including it into the state constraint set \mathbb{X} . However, when x_1^* and x_2^* are sufficiently far apart, then there is no incentive for the finite horizon optimal trajectory x_2^u to stay near x_1^* and vice versa. This is the situation in the following examples in which we did not explicitly include this condition in the optimization.

The following numerical simulation of the MPC control of this example confirms that the scheme yields a feasible and stable closed loop. The numerical examples were performed with MATLAB² using the stage cost $\ell_p(x_p, u_p) = \|x_p - x_p^*\| + 0.1 \|u_p\|$ converting the optimal control problems in static optimization problems which are solved with MATLAB's `fmincon`-routine. Here the control functions in step (0) of Algorithm 3.1 are computed using optimal control similar to step (1), where the admissible control sequences for the p -th system are defined via the state constraints induced by the predictions $x_1^u(k), \dots, x_{p-1}^u(k)$. In all examples we have added the additional constraints $x_p \in [-1, 1]^2$ to \mathbb{X} and we have chosen the initial conditions $x_p(0)$ at the boundary of $[-1, 1]^2$ and the desired equilibria on the opposite side of $[-1, 1]^2$, i.e., $x_p^* = -x_p(0)$. This implies that the agents meet far away from the boundary of $[-1, 1]^2$ and from the equilibria x_p^* , thus the theoretical analysis from the first part of this section remains valid. Figure 1 shows a corresponding simulation in which $N = 3$ turns out to be sufficient for stability.

Further numerical simulations show that the scheme is stable also for a larger number P of agents. However, in this case it becomes more difficult to control the individual systems to their equilibria which is reflected by the fact that for $P = 4$ systems and the initial values from Figure 2 — which extend those of Figure 1 — stability is only obtained for $N \geq 8$. However, even with $N = 8$ the horizon is considerably shorter than the optimization horizon needed in order to find predictions which end up in stabilizing terminal constraint sets around the equilibria.

² all MATLAB-Files are available on www.math.uni-bayreuth.de/~lgruene/publ/disNMPC.html

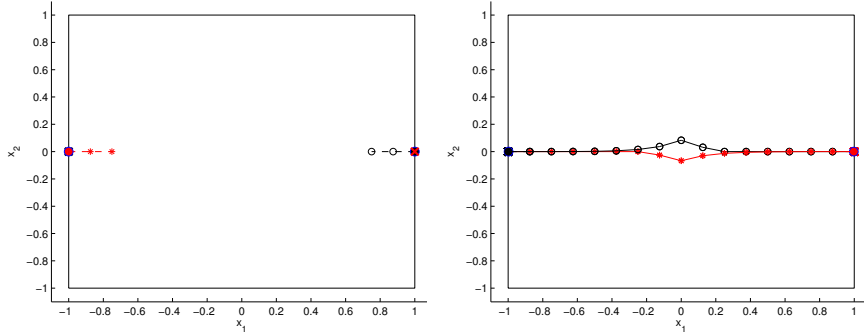


Fig. 1 MPC for Example 2.1 with $P = 2$ systems with initial values $x_1(0) = (1,0)^T$, $x_2(0) = (-1,0)^T$ and optimization horizon $N = 3$. Prediction for $k = 0$ (left) and final trajectories at $k = 16$ (right).

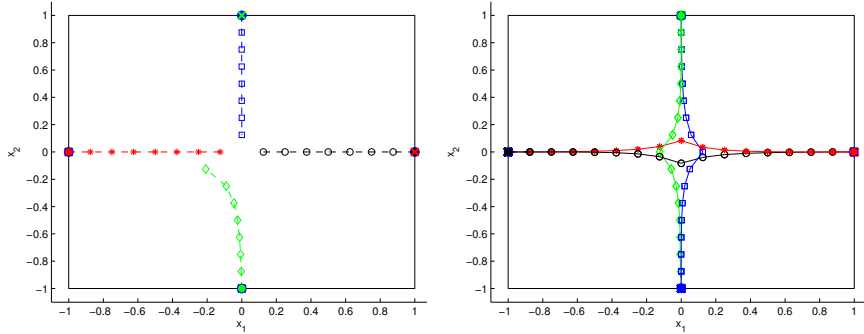


Fig. 2 MPC for Example 2.1 with $P = 4$ systems with initial values $x_1(0) = (1,0)^T$, $x_2(0) = (-1,0)^T$, $x_3(0) = (0,1)^T$, $x_4(0) = (0,-1)^T$ and optimization horizon $N = 8$. Prediction for $k = 0$ (left) and final trajectories at $k = 16$ (right).

Our final numerical experiment shows that the optimization horizon N needed in order to obtain stability of the closed loop heavily depends on the initial values, i.e., on the way how the individual agents meet and which way they are heading when they meet. Due to the fact that the control constraints $u_p \in [-\bar{u}, \bar{u}]$ are box constraints which allow the systems to move faster and more flexible in diagonal direction, it is not surprising that avoiding conflicts is easier when the agents are approaching each other in a diagonal way. This explains why resolving the conflict is easier in the situation of Figure 3 in which the optimization horizon $N = 3$ is sufficient in order to stabilize all $P = 4$ agents. A further interesting observation in this example is that the resulting trajectories are not symmetric as in Figures 1 and 2. Rather, here one sees the effect of the sequential ordering since x_1 and x_3 approach their respective equilibria directly (with x_3 performing a shorter step at time $k = 7$ in order to avoid the collision with x_1) while x_2 and x_4 are forced to take small detours.

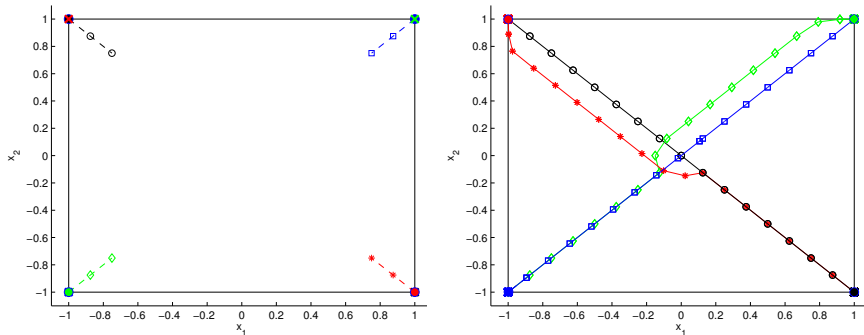


Fig. 3 MPC for Example 2.1 with $P = 4$ systems with initial values $x_1(0) = (-1, 1)^T$, $x_2(0) = (1, -1)^T$, $x_3(0) = (1, 1)^T$, $x_4(0) = (-1, -1)^T$, optimization horizon $N = 3$ and different initial condition as in Figure 2. Prediction for $k = 0$ (left) and final trajectories at $k = 18$ (right)

Due to the short horizons N , most of the conflicts, i.e., the possible collisions, are resolved by the optimization at runtime in step (1) and not in the initialization in step (0) of Algorithm 3.1. The only exception is the fourth system starting at $x_4(0) = (0, -1)^T$ in Figure 2, which at the time of its first optimization in step (0) already knows all other systems' predictions. Hence, the simulations nicely illustrate our schemes' ability to resolve conflicts at runtime.

7 Conclusion and future work

In this paper we have shown that the non-cooperative distributed model predictive control scheme from [13, 14] can be formulated without stabilizing terminal constraints. An extension of the controllability based NMPC stability analysis from [6, 8] yields a sufficient distributed controllability condition ensuring stability and feasibility. Numerical examples show that the resulting scheme is able to stabilize a test system with small optimization horizons N and illustrate the schemes' ability to resolve conflicts at runtime.

We regard the analysis in this paper as a first step which can and needs to be improved in many ways. The controllability conditions (i) and (ii) required in Theorem 5.3 are certainly difficult to check for systems more complex than Example 2.1 and even for Example 2.1 with large P a rigorous verification currently appears out of reach. In fact, the inequality (22) resulting from the controllability condition is a quite strong property by itself in the sense that in each sampling period each optimal value function V_p is decreasing. This property is most likely too demanding in many applications in which one would rather expect that in each step only some of the V_p decrease while others may even increase. ISS-small gain arguments for large systems as in [3] may be suitable for handling such situations, however, so far

it is an open questions what kind of controllability properties are needed in order to ensure the appropriate small gain properties of the NMPC closed loop systems.

Another extension will be to relax the requirement of strict sequential optimization imposed in Algorithm 3.1. A first step in this direction could be to exploit the fact that stability can also be expected if the optimization is performed only in each m -th sampling period for $m \in \{1, \dots, N-1\}$, cf. [8]. This enables us to set up a cyclic scheme in which in each sampling period only a subset of systems performs an optimization. While this reduces the waiting time, it does not completely remove the sequential order of the optimization. The development of a scheme in which the optimization is performed completely in parallel remains a major challenge.

8 Appendix

In this section we prove an auxiliary result which is needed for the analysis of the example in Section 6. In Lemma 8.1 we assume that the norm used in the definition of \mathbb{X} is the maximum norm $\|\cdot\|_\infty$. For the Euclidean norm see Remark 8.2.

Lemma 8.1. *Let $K > 0$, $R > \delta > \bar{u}$ and a trajectory $y(\cdot)$ with $\|y(k)\|_\infty \geq R + \delta$, $k \in \mathbb{N}_0$ be given and define the set \mathbb{X} according to Example 2.1 for $P = 2$. Then there exists $\bar{k} \in \mathbb{N}$ such that for each $x^0 \notin T = [-R, R]^2$ with $\|x^0\| \leq K$ and $(x^0, y(0)) \in \mathbb{X}$ there is a finite number $k^* \in \mathbb{N}$ with $k^* \leq \bar{k}$ and a control $u(\cdot)$ satisfying $(x^u(k, x^0), y(k)) \in \mathbb{X}$ for all $k = 0, \dots, k^*$ and $x^u(k^*, x^0) \in T$.*

Proof. Note that $(x^0, y(0)) \in \mathbb{X}$ implies $\|x^0 - y(0)\|_\infty \geq \delta$. We construct $u(\cdot)$ in three steps and start by finding a control sequence such that

$$\min\{|x_1^u(k_1)|, |x_2^u(k_1)|\} < \bar{u} \quad (24)$$

holds for some $k_1 \in \mathbb{N}$. If this inequality holds for $k = 0$, then there is nothing left to do. Otherwise, at each time $k = 0, 1, 2, \dots$ such that (24) does not hold we choose one of the following control sequences. If $|x_1^0 - y_1(0)| \geq \delta$ then we set

$$u(k) = \begin{pmatrix} y_1(k+1) - y_1(k) \\ -\bar{u} \operatorname{sign}(x_2^u(k)) \end{pmatrix}.$$

which implies $|x_1^u(k) - y_1(k)| = |x_1^0 - y_1(0)| \geq \delta$ and $|x_2^u(k)| = |x_2^0| - k\bar{u}$ for $k \leq k_1$. Thus, the trajectory is admissible and (24) is satisfied for $k_1 = \lfloor |x_2^0|/\bar{u} \rfloor$. Note that $|u_1(k)| \leq \bar{u}$ since $y_1(k+1) - y_1(k)$ equals the control applied at time k in order to generate $y(\cdot)$. If $|x_1^0 - y_1(0)| < \delta$ then $\|x^0 - y(0)\|_\infty \geq \delta$ implies that $|x_2^0 - y_2(0)| \geq \delta$ holds and we can do the analogous construction exchanging x_1^0 and x_2^0 which implies (24) for $k_1 = \lfloor |x_1^0|/\bar{u} \rfloor$. All in all this shows that (24) can always be satisfied with $k_1 \leq \lfloor \|x^0\|_\infty/\bar{u} \rfloor$. Furthermore, note that $\|x^u(k_1)\|_\infty \leq \|x^0\|_\infty + \lfloor \|x^0\|_\infty/\bar{u} \rfloor \bar{u} \leq 2\|x^0\|_\infty$ holds.

In the following second step we construct $u(k_1), u(k_1+1), \dots, u(k_2)$ such that we approach T as fast as possible until we either reach T or are blocked by $y(\cdot)$. To this

end, we assume without loss of generality $|x_1^u(k_1)| < \bar{u}$ and $x_2^u(k_1) > 0$; otherwise we use symmetric arguments in the sequel. We set

$$u(k) = \begin{pmatrix} -x_1^u(k) \\ -\bar{u} \end{pmatrix}$$

for $k = k_1, k_1 + 1, \dots, k_2$ where $k_2 \geq k_1$ is the minimal time at which either $x^u(k_2) \in T$ or $\|x^u(k_2 + 1) - y(k_2 + 1)\|_\infty < \delta$ holds. Note that since $x_1^u(k_2) \in (-\bar{u}, \bar{u}) \subset [-R, R]$ and $x_2^u(k_2) = x_2^u(k_1) - (k_2 - k_1)\bar{u} \in [-R, R]$ for $(k_2 - k_1)\bar{u} \in [x_2^u(k_1) - R, x_2^u(k_1) + R]$, we obtain $k_2 \leq k_1 + \lceil \|x^u(k_1)\|_\infty / \bar{u} \rceil \leq k_1 + \lceil 2\|x^0\|_\infty / \bar{u} \rceil$.

If $x^u(k_2) \in T$ then we set $k^* = k_2$ and are done, otherwise, we continue with the third part of our construction for $u_1(k_2), u_1(k_2 + 1), \dots$. To this end we distinguish two cases. The first case is that $x_2^u(k_2) \geq y_2(k_2) + \delta$ holds. In this case we set

$$u(k_2) = \begin{pmatrix} -x_1^u(k_2) \\ y_2(k_2 + 1) + \delta - x_2^u(k_2) \end{pmatrix}$$

which implies $x_2^u(k_2 + 1) = y_2(k_2 + 1) + \delta$ and thus $\|x^u(k_2 + 1) - y(k_2 + 1)\|_\infty = \delta$. Furthermore we have $\|x^u(k_2 + 1)\|_\infty \leq \|x^u(k_2)\|_\infty + \bar{u}$. Observe that by choice of k_2 the relation $x^u(k_2) \notin T$ implies $x_2^u(k_2) > y_2(k_2) - \delta$.

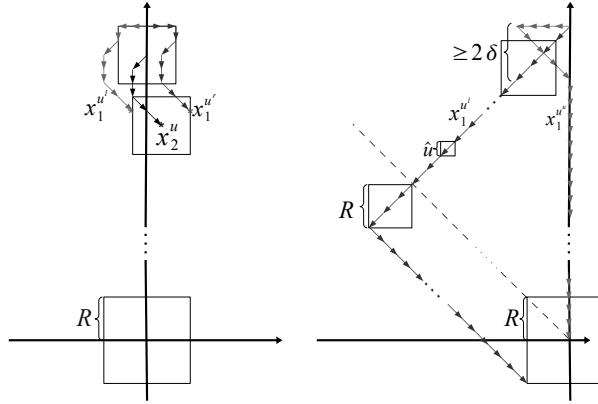


Fig. 4 Illustration of the trajectories used in the construction of x^u .

Now we continue by constructing two alternative trajectories $x^{u^r}(k)$ and $x^{u^l}(k)$ for $k \geq k_2 + 1$, cf. Figure 4 (left). At least one of them is admissible and reaches T in a predetermined number of steps. The first elements of the corresponding control sequences are

$$u^r(k) = \begin{pmatrix} \bar{u} \\ y_2(k+1) - y_2(k) \end{pmatrix} \quad \text{and} \quad u^l(k) = \begin{pmatrix} -\bar{u} \\ y_2(k+1) - y_2(k) \end{pmatrix}.$$

As soon as $|x_1^{u^j}(k+1) - y_1(k+1)| \geq \delta$ holds for $j \in \{l, r\}$ and the following control values, we change the control inputs to

$$u_1^r(k) = \begin{cases} -\min\{\bar{u}, \max\{x_1^{u^r}(k) - \delta, -\bar{u}\}\}, & y_1(k) + \delta \leq \delta \\ y_1(k+1) + \delta - x_1^u(k), & \text{otherwise} \end{cases}$$

and $u_2^r(k) = -\bar{u}$ if $x_2^{u^r}(k) > R$ and $u_2^r(k) = 0$ otherwise, and

$$u_{11}^l(k) = \begin{cases} \min\{\bar{u}, \max\{-x_1^{u^l}(k) - \delta, -\bar{u}\}\}, & y_1(k) - \delta \geq -\delta \\ y_1(k+1) - \delta - x_1^u(k), & \text{otherwise} \end{cases}$$

and $u_2^l(k) = -\bar{u}$ if $x_2^l(k) > R$ and $u_2^l(k) = 0$ otherwise, respectively. We denote the times at which this is done as k_3^j and k_3^r . Observe that at least for one $j \in \{l, r\}$ we have $k_3^j \leq k_2 + \lceil \delta/\bar{u} \rceil$ and for the corresponding trajectory we get $x_2^{u^j}(k_3^j) \leq x_2^u(k_2) + \delta + \bar{u}$. Moreover, the choice of $u_1^j(k)$ implies that $u^j(k)$ is admissible for all $k \geq k_3^j$, $j \in \{r, l\}$ and that for each $k \geq k_2 + 1$ we have $x_1^{u^j} \in [-\delta, \delta]$ for at least one $j \in \{r, l\}$. Since after the times k_3^j the trajectories move down as fast as possible, we obtain $x_2^{u^j}(k) \in [-R, R]$ for all $k \geq k_4^j$ with $k_4^j \leq k_3^j + \left\lfloor \frac{x_2^u(k_2) + \delta + \bar{u}}{\bar{u}} \right\rfloor \leq k_3^j + \left\lfloor \frac{2\|x^0\|_\infty + \delta + \bar{u}}{\bar{u}} \right\rfloor$. Without loss of generality let $k_4^l \leq k_4^r$ which implies $k_4^l \leq k_2 + \lceil \delta/\bar{u} \rceil + \left\lfloor \frac{2\|x^0\|_\infty + \delta + \bar{u}}{\bar{u}} \right\rfloor$. Then, by construction of u^l we have that either $x^{u^l}(k_4^l) \in T$ or $y_1(k_4^l) < 0$. In the latter case our construction implies $k_3^r \leq k_4^l$. This, in turn, implies $k_4^r \leq k_4^l + \left\lfloor \frac{2\|x^0\|_\infty + \delta + \bar{u}}{\bar{u}} \right\rfloor + k_4^l - k_2 \leq k_2 + 3 \left\lfloor \frac{2\|x^0\|_\infty + \delta + \bar{u}}{\bar{u}} \right\rfloor + 2\lceil \delta/\bar{u} \rceil$. At time k_4^r we have $x_2^{u^j}(k_4^r) \in [-R, R]$ for all $j \in \{r, l\}$ which together with $x_1^{u^j} \in [-\delta, \delta]$ for at least one $j \in \{r, l\}$ implies $x_1^{u^j}(k_4^r) \in T$ for at least one $j \in \{r, l\}$. Hence, at least one of the trajectories reaches T in the time $k^* = k_4^r \leq \left\lfloor \frac{\|x^0\|_\infty}{\bar{u}} \right\rfloor + \left\lfloor \frac{2\|x^0\|_\infty}{\bar{u}} \right\rfloor + 3 \left\lfloor \frac{2\|x^0\|_\infty + \delta + 2\bar{u}}{\bar{u}} \right\rfloor + 2\lceil \delta/\bar{u} \rceil$.

It remains to deal with the case $x_2^u(k_2) < y_2(k_2) + \delta$ and $|x_1^u(k_2)| \leq \bar{u}$. Again we construct two alternative trajectories for $k \geq k_2$, cf. Figure 4 (right). Recall that from the construction of k_2 and $x_1^u(k_2) \notin T$ the inequality $x_2^u(k_2) > y_2(k_2) - \delta$ follows. Since $\|x^u(k_2) - y(k_2)\|_\infty \geq \delta$ this implies $|x_1^u(k_2) - y_1(k_2)| \geq \delta$. Without loss of generality we assume $x_1^u(k_2) \leq y_1(k_2) - \delta$. We define two different control sequences for $k \geq k_2$ whose first elements are given as

$$u^u(k) = \begin{pmatrix} -\bar{u} \\ 0 \end{pmatrix} \quad \text{and} \quad u^l(k) = \begin{pmatrix} -\bar{u} \\ -\bar{u} \end{pmatrix}.$$

Since x^{u^j} , $j \in \{u, l\}$, moves left with maximal speed, $x_1^{u^j}(k) \leq y_1(k) - \delta$ holds implying feasibility, i.e., $(x^{u^j}(k), y(k)) \in \mathbb{X}$. Furthermore, this choice implies $x_2^{u^u}(k_3) - x_2^{u^l}(k_3) \geq 2\delta$ at time $k_3 = k_2 + \lceil 2\delta/\bar{u} \rceil$. Then, for $k \geq k_3$, we use the controls

$$u^u(k) = \begin{pmatrix} \min\{-x_1^{u^u}(k), \bar{u}\} \\ -\bar{u} \end{pmatrix}$$

and

$$u_1^l(k) = \begin{cases} -\bar{u}, & -x_1^{u^l}(k) \leq x_2^{u^l}(k) + 2(R - \bar{u}), \\ \bar{u}, & \text{otherwise} \end{cases}, \quad u_2^l(k) = -\bar{u}.$$

By k^l we denote the minimal time at which $u_1^l(k^l) = \bar{u}$ holds. Provided that the resulting trajectories are admissible, they both reach T before time $k_4 \leq k_2 + \lceil (2\|x^0\| + R)/\bar{u} \rceil$. Furthermore, they satisfy $x_2^{u^u}(k) - x_2^{u^l}(k) \geq 2\delta$ for $k \in \{k_3, \dots, k_4\}$. Since x^{u^l} moves left with maximal speed for $k = k_3, \dots, k^l$ and satisfies $x_1^{u^l}(k_3) \leq y_1(k_3) - \delta$, we obtain that the trajectory is admissible at least until time $k = k^l$.

Now assume that there exists $\tilde{k} \in \{k^l + 1, \dots, k_4 - 1\}$ at which x^{u^l} is not admissible, i.e., at which $\|x^{u^l}(\tilde{k}) - y(\tilde{k})\|_\infty < \delta$ holds. Since x^{u^l} moves downwards with maximal speed, this is only possible if $y_2(k) \leq x_2^{u^l}(k) + \delta$ holds for $k \in \{k_3, \dots, \tilde{k}\}$. Since this implies that $x_2^{u^u}(k) \geq x_2^{u^l}(k) + 2\delta \geq y_2(k) + \delta$ holds, $x^{u^u}(k)$ is admissible for $k = k_3, \dots, \tilde{k}$. On the other hand, $\|x^{u^l}(\tilde{k}) - y(\tilde{k})\|_\infty < \delta$ for $\tilde{k} \geq k^l + 1$ and a little computation based on the definition of k^l reveals that $y_1(k) \leq -\delta$ holds as long as $x^{u^u}(k) \notin T$. Hence, x^{u^u} is admissible until it reaches T and consequently, at least one of the trajectories is admissible and reaches T in a time $k^* \leq k_4 \leq \left\lfloor \frac{\|x^0\|_\infty}{\bar{u}} \right\rfloor + \left\lfloor \frac{2\|x^0\|_\infty}{\bar{u}} \right\rfloor + \left\lceil \frac{2\|x^0\|_\infty + R}{\bar{u}} \right\rceil$.

Summarizing, we have shown that u and a finite number k^* with the required properties required exist. Since $\|x_1^0\|_\infty$ is bounded from above by K , the corresponding time k^* satisfies $k^* \leq \bar{k}$ for some suitable \bar{k} depending on K and R which completes the proof.

Remark 8.2. Using the Euclidean norm $\|\cdot\|_2$ in Lemma 8.1 is also possible. To this end, one has to assume $\|y(k)\|_2 \geq \sqrt{2}(R + \delta)$ which implies $\|y(k)\|_\infty \geq (R + \delta)$. Moreover, an additional preliminary step is required since $\|x^0 - y(0)\|_2 \geq \delta$ does not imply $\|x^0 - y(0)\|_\infty \geq \delta$. However, since $y(\cdot)$ is known, this can be easily obtained in a finite number of steps. Then, one may proceed as in the proof of Lemma 8.1 since $\|x^u(k) - y(k)\|_\infty \geq \delta$ implies $(x^u(k), y(k)) \in \mathbb{X}$ for the Euclidean norm $\|\cdot\|_2$ as well. Apart from this initialization step, the term $\|x^0\|_\infty$ in the estimates for k^* remains the same.

References

1. Altmüller, N., Grüne, L., Worthmann, K.: Performance of NMPC schemes without stabilizing terminal constraints. In: M. Diehl, F. Glineur, E. Jarlebring, W. Michiels (eds.) Recent Advances in Optimization and its Applications in Engineering, pp. 289–298. Springer-Verlag (2010)

2. Altmüller, N., Grüne, L., Worthmann, K.: Receding horizon optimal control for the wave equation. In: Proceedings of the 49th IEEE Conference on Decision and Control, pp. 3427–3432. Atlanta, Georgia (2010)
3. Dashkovskiy, S., Rüffer, B.S., Wirth, F.R.: An ISS small gain theorem for general networks. *Math. Control Signals Systems* **19**(2), 93–122 (2007)
4. Giselsson, P., Rantzer, A.: Distributed model predictive control with suboptimality and stability guarantees. In: Proceedings of the 49th IEEE Conference on Decision and Control, pp. 7272–7277. Atlanta, Georgia (2010)
5. Grimm, G., Messina, M.J., Tuna, S.E., Teel, A.R.: Model predictive control: for want of a local control Lyapunov function, all is not lost. *IEEE Trans. Automat. Control* **50**(5), 546–558 (2005)
6. Grüne, L.: Analysis and design of unconstrained nonlinear MPC schemes for finite and infinite dimensional systems. *SIAM J. Control Optim.* **48**, 1206–1228 (2009)
7. Grüne, L., Pannek, J.: *Nonlinear Model Predictive Control. Theory and Algorithms*. Springer-Verlag, London (2011). To appear
8. Grüne, L., Pannek, J., Seehafer, M., Worthmann, K.: Analysis of unconstrained nonlinear MPC schemes with varying control horizon. *SIAM J. Control Optim.* **48**, 4938–4962 (2010)
9. Jadbabaie, A., Hauser, J.: On the stability of receding horizon control with a general terminal cost. *IEEE Trans. Automat. Control* **50**(5), 674–678 (2005)
10. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Sokaert, P.O.M.: Constrained model predictive control: stability and optimality. *Automatica* **36**, 789–814 (2000)
11. Primbs, J.A., Nevistić, V.: Feasibility and stability of constrained finite receding horizon control. *Automatica* **36**(7), 965–971 (2000)
12. Rawlings, J.B., Mayne, D.Q.: *Model Predictive Control: Theory and Design*. Nob Hill Publishing, Madison (2009)
13. Richards, A., How, J.: A decentralized algorithm for robust constrained model predictive control. In: Proceedings of the American Control Conference, pp. 4261–4266. Boston, Massachusetts (2004)
14. Richards, A., How, J.P.: Robust distributed model predictive control. *Internat. J. Control* **80**(9), 1517–1531 (2007)
15. Scattolini, R.: Architectures for distributed and hierarchical Model Predictive Control - A review. *J. Process Control* **19**(5), 723–731 (2009)