RUHR-UNIVERSITÄT BOCHUM
Institute of Automation and Computer Control
Prof. Dr.-Ing. Jan Lunze

Universitätsstr. 150 Phone +49-(0)234 32-24096
D-44805 Bochum Fax +49-(0)234 32-14101

Autonomous and cooperative control of networked discrete-event systems

Markus Zgorzelski
zgorzelski@atp.rub.de

1 Introduction

A *networked discrete-event system* is a network of subsystems, each of which includes a technical process $\bar{\Sigma}_i$ together with a controller and a network unit Σ_i'. These subsystems are interconnected by physical couplings Σ_{K} and by digital communication links Σ_{N} (Fig. 1) [1]. Important characteristics of networked systems are the partial autonomy of the subsystems and a flat architecture, in which there does not exist any coordinator.

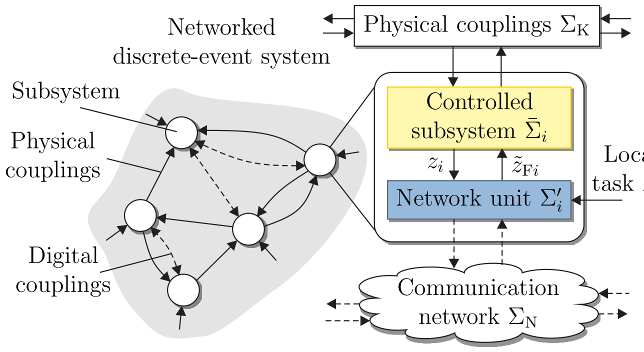

Networked discrete-event system
Physical couplings Σ_{K}
Subsystem
Physical couplings
Controlled subsystem $\bar{\Sigma}_i$
z_i
$\bar{z}_{\mathrm{F}i}$
Local task $z_{\mathrm{F}i}$
Network unit Σ_i'
Digital couplings
Communication network Σ_{N}

Figure 1: Networked discrete-event system

The autonomy of the subsystems is reflected by the fact that each subsystem individually solves its local tasks, which change during runtime. Cooperation among the subsystems becomes necessary, if physical couplings or control specifications have to be resolved by two or more subsystems in order to satisfy the local tasks. Hence, the subsystems change between an autonomous and a cooperative operation mode. On the one hand, they are able to achieve most of their local tasks autonomously without any communication. On the other hand, if it is necessary, they participate in satisfying cooperative tasks by adapting their behaviours while using the communication network.

The aim of this project is to develop design methods for networked discrete-event systems that enable the subsystems to decide upon local model information when they have to communicate over a digital communication network. The following questions have to be answered:

> *When and what information have to be exchanged by the subsystems and how should the structure of the communication network look like?*

2 Example: Networked robots

Figure 2 shows two robots, Robot 1 and Robot 2, as two subsystems of a networked discrete-event system. Figure 2(a) describes the situation, in which Robot 1 has the local task to install car doors onto a car body, while Robot 2 has the task to install bonnets. The tasks are given by the car parts using radio-frequency identification (RFID) tags and, thus, appear in arbitrary order. It is clear to see that the local tasks in Fig. 2(a) are achieved autonomously by the robots, and, thus, the subsystems need only local model information.

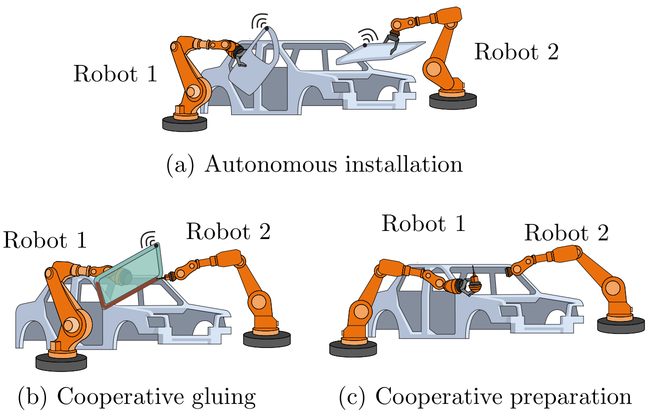


Figure 2: Autonomous and cooperative control

Figure 2(b) shows the installation of the car window. Robot 2 applies glue on a car window, while Robot 1 is grabbing and holding the car window. This cooperative task is described by a synchronous state transition in both robot models that have to be executed synchronously.

As can be seen in Fig. 2(c), Robot 2 needs a gluing tool to accomplish the cooperative task in Fig. 2(b). Therefore, both robots cooperatively have to exchange the tool of Robot 2 from grabbing to gluing, which is modelled by another synchronous state transition. Both synchronous state transitions have to be executed in the correct order, which has to be determined by the robots without any coordinator.

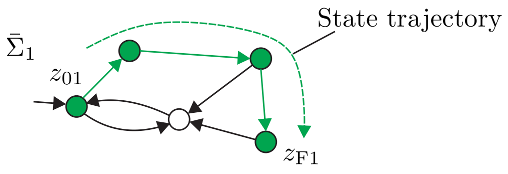
A deadlock between the robots occurs if Robot 1 holds the window for gluing, while Robot 2 waits for exchanging the tool from grabbing to gluing. Obviously, the robots have to exchange information to install the car window cooperatively.

3 Local and cooperative tasks

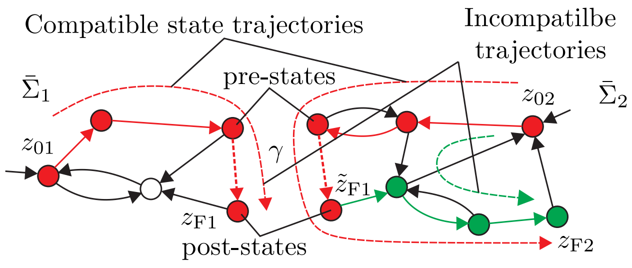
The subsystems have to solve tasks which are modelled by local target states $(z_{\mathrm{F}1}, \dots, z_{\mathrm{F}I}) \in \prod_{i=1}^{I} \mathcal{Z}_i$, for each technical process Σ_i independently of each other. They are solved if the subsystems execute state trajectories

$(Z_1(0 \dots k_e), \dots, Z_I(0, \dots, k_e))$ from their current state $(z_1, \dots, z_I) \in \prod_{i=1}^{I} \mathcal{Z}_i$ into each target state $(z_{F1}, \dots, z_{FI})$ (Fig. 3(a)).

The cooperative tasks are modelled by synchronous state transitions γ, which are defined by pairs of state transitions from the predecessor state (pre-state) into the successor state (post-state) between two subsystems, which is shown in Fig. 3(b).



(a) Local state trajectroy



(b) Compatible state trajectories

Figure 3: Local and cooperative tasks

To solve the local and cooperative tasks, the subsystem have to execute compatible state trajectories into their target states, which include a deadlock-free sequence of synchronous state transitions (Fig. 3(b)). The problem to be solved occurs if the subsystems execute incompatible state trajectories leading into a deadlock. Then the subsystems do not execute the same synchronous state transitions and cannot reach their target states.

4 Cooperative control

The aim of this project is to design a set of digitally connected network units Σ'_i (Fig. 1), which detect incompatible states trajectories and replace them by compatible ones to reach the local target states z_{Fi} (Fig. 3(b)). This control method is denoted as *cooperative target-state control.*

To enable the network unit to find compatible state trajectories the composed overall model is needed. However, the overall model does not exist, because the subsystem models are distributed among the subsystems. Obviously, only the cooperative behaviour of the networked discrete-event system has to be known being modelled by a reduced overall model, which can be applied to find the compatible state trajectories instead of the original overall model [2].

The reduced overall model is calculated by the compositional model abstraction method from [2] as shown in Fig. 4. First, each subsystem model $\bar{\Sigma}_i$ is simplified individually to the reduced subsystem model $\bar{\Sigma}_{i/\sim}$. Second, the reduced subsystem models $\bar{\Sigma}_{i/\sim}$ are composed to the abstracted overall model $\bar{\Sigma}_{/\sim}$.

To solve the cooperative target-state control problem the subsystems are steered through the abstracted state trajectory in the reduced overall model $\bar{\Sigma}_{/\sim}$ from the abstracted

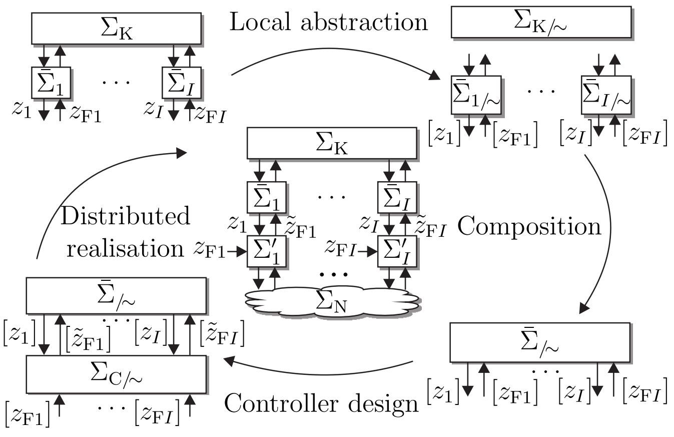


Figure 4: Compositional model abstraction and controller design

current state into the abstracted target state, which corresponds to compatible state trajectories in the original models (Fig. 5).

As an example, in Fig. 5 the subsystems $\bar{\Sigma}_1$ and $\bar{\Sigma}_2$ have to be steered from (z_1, z_2) into (z_{F1}, z_{F2}). The compatible state trajectories including γ_1 before γ_2 are determined by the abstracted state trajectory in the reduced overall model $\bar{\Sigma}_{/\sim}$ from the abstracted current state $([z_1], [z_2])$ into the abstracted target state $([z_{F1}], [z_{F2}])$.

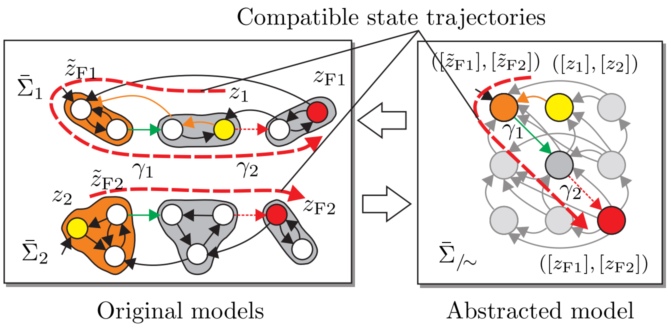


Figure 5: Cooperative control

The subsystems $\bar{\Sigma}_i$ are steered through the abstracted state trajectory by a new *cooperative controller* $\Sigma_{C/\sim}$, which is designed for $\bar{\Sigma}_{/\sim}$ as shown in Fig. 4 [3]. If incompatible state trajectories are detected by the network units Σ'_i, the distributed components $\Sigma_{Ci/\sim}$ of $\Sigma_{C/\sim}$ are used by Σ'_i to select compatible state trajectories for their controlled subsystems $\bar{\Sigma}_i$ according to the abstracted state trajectory in $\bar{\Sigma}_{/\sim}$. Obviously, the network units Σ'_i have to use the communication network Σ_N in these situations.

References

[1] M. Zgorzelski and J. Lunze. A method for the synchronisation of networked discrete-event systems, 13th International Workshop on Discrete Event Systems, Xi'an 2016, pp. 444-451

[2] M. Zgorzelski and J. Lunze. A model abstraction method for networked discrete-event systems, 17th European Control Conference, Neaples 2019, pp. 3976–3983

[3] M. Zgorzelski and J. Lunze. Cooperative tracking control in networked discrete-event systems, 6th International Conference on Control, Decision and Information Technologies 2019, Paris 2019, pp. 115-120