Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

Cooperative control of a flexible manufacturing system

Markus Zgorzelski * **Jan Lunze** *

* *Institute of Automation and Computer Control, Ruhr-Universität Bochum, Universitätsstrasse 150, D-44780 Bochum, Germany (e-mail: {zgorzelski, lunze}@atp.rub.de)*

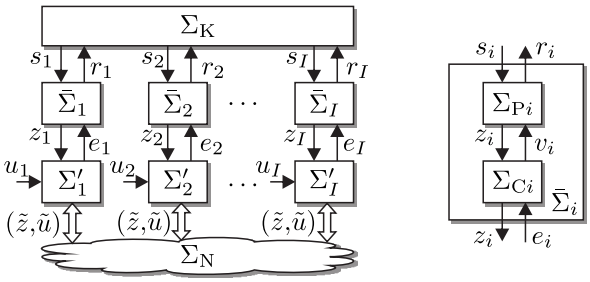
Abstract: This paper considers networked discrete-event systems. Local state-feedback controllers enable each subsystem to reach local target states. Due to the physical restrictions between the subsystems, the local target states might not be reachable autonomously, but cooperatively with the help of other subsystems. Therefore, each subsystem is extended by a network unit, which detects and resolves possible physical restrictions. If cooperation is necessary, the network units temporarily modify their local target states while applying situation-dependent communication. The proposed method is applied to the flexible manufacturing system HANS and its applicability to real-life systems is demonstrated by experimental results.

Keywords: Networked discrete-event systems, cooperative tracking control

1. INTRODUCTION

1.1 Networked discrete-event systems

Networked discrete-event systems (Fig. 1(a)) consist of controlled subsystems $\bar{\Sigma}_i$, which are able to use a digital communication network Σ_N. Each subsystem has a local state-feedback controller to reach a target state $z_{\mathrm{F}i}$. The physical couplings Σ_K are represented as synchronous state transitions of neighbouring subsystems. If they become active, two subsystems have to execute these synchronous state transitions together before the subsystems are able to reach their local target states. The task to follow the synchronous state transitions necessitates a cooperation among the controllers.



(a) Networked discrete-event system (b) Controlled subsystem

Fig. 1. Networked system

This paper experimentally evaluates a cooperative control method for networked discrete-event systems. If a subsystem has to execute a cooperative task to satisfy its local task, then the subsystems apply the communication network to organise the execution of this cooperation. That means, each subsystem is steered into target states $z_{\mathrm{F}i}$ to solve its local tasks. If a subsystem has to execute a synchronous state transition to reach $z_{\mathrm{F}i}$, then the subsystems temporarily modify the local target states $z_{\mathrm{F}i}$ to ensure the execution of this synchronous state transition.

⋆ This work was supported by the German Research Foundation (DFG) under grant LU 462/42.

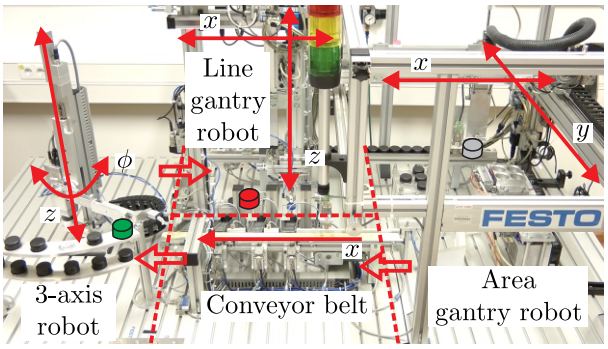


Fig. 2. Flexible manufacturing system HANS

To this aim, the network units Σ'_i in Fig. 1(a) are introduced to enable each controlled subsystem $\bar{\Sigma}_i$ to exchange situation-dependent information among each other via the digital communication network Σ_N. Based on a reduced overall model, which mainly describes the physical restrictions among the subsystems and abstracts from the autonomous behaviour, the network units cooperatively determine the modified target states $\tilde{z}_{\mathrm{F}i}$ for their subsystems.

As an example, the flexible manufacturing system HANS in Fig. 2 is considered (Zgorzelski and Lunze, 2018a) (Modular production system MPS® of the company FESTO), which is a pick-and-place system for handling workpieces. It consists of four subsystems: The area-gantry robot, the line-gantry robot, the 3-axis robot and the conveyor belt. Each subsystem can transport the workpieces autonomously in the direction indicated by the solid red arrows (autonomous mode). However, in the networked system, the subsystem can cooperatively transfer the workpiecess between the subsystems, which is shown by the red double arrows (Fig. 2). It is clear to see that for the transfer the subsystem have to cooperate temporarily (cooperative mode), which is solved by the proposed method.

The first contribution of this paper is an extension of the cooperative control method of Zgorzelski and Lunze (2019) from two subsystems to an arbitrary number of coupled subsystems. In order to do that, the method from Zgorzelski

and Lunze (2019) is generalised through the application of extended Input/Output (I/O)-automata (Drüppel et al., 2008), which introduces a new form of model describing the system structure. The second contribution is the experimental application of the proposed method to the cooperative pick-and-place process HANS with four subsystems (Fig. 2), which will demonstrate the applicability of the presented method to real-life systems.

1.2 Literature

The supervisory control of discrete-event system proposed by Ramadge and Wonham (1989) has been investigated to connect decentralised supervisors via a digital communication network (Barrett and Lafortune, 2000),(Rudie et al., 2003) to restrict the plant behaviour to a given specification. Moreover, in the multi-agent supervisory control (Su and Lennartson, 2017) and distributed supervisory control (Seow et al., 2009), (Cai and Wonham, 2010) the communication network is used for the synchronous event passing between the local supervisors. Other works on networked supervisory control (Wang et al., 2016) consider a communication network between the plant and the supervisor, which causes a delayed transmission of events and control actions.

In contrast, our control objective is the state-feedback control of I/O automata (Zgorzelski and Lunze, 2017) to steer physically coupled subsystems into local target states. Although the state attraction problem (Brave and Heymann, 1989) of the supervisory control is related to this control problem, Brave and Heymann (1989) did not discuss how to cope with more than one subsystem and a control input that flexibly allows the subsystems to adapt to its environment for reaching different target states given at runtime. Moreover, our communication depends on the plant behaviour but also on the control input, which requires more flexibility.

The cooperative tracking control proposed in this paper has similarities to the distributed optimal controller proposed by Stursberg and Hillmann (2017), which has the control objective to minimize the state trajectory cost of subsystems into their goal states within a distributed architecture with a priority structure. The difference is that we do not want to minimize the costs and we do not restrict our system to a certain priority structure. Moreover, the physical restrictions in this paper have a more generalised description and the solution presented in this paper is given for arbitrary couplings and any number of subsystems. Hence, in the literature, the networked discrete-event systems appear with a different structure or a different control objective.

1.3 Organisation of the paper and notation

In Section 2, the physically coupled controlled subsystems are presented and the cooperative control problem is formulated. The main result is presented in Section 3 that gives the solution of the cooperative control problem for networked discrete-event systems. Finally, Section 4 presents the modelling and the cooperative control of the flexible manufacturing system HANS.

In the following, sets are denoted by calligraphic letters (e.g. $\mathcal{Z}$) and scalars by lower cases italics (e.g. z_0). To denote sequences of scalars, upper case italics are used (e.g. $Z(0 \dots k_\mathrm{e}) = (z(0), v(1), \dots z(k_\mathrm{e}))$ with k_e being the time horizon. State transitions are denoted by $(z_i \to z_i')$ and the empty symbol is denoted by ε. $\mathcal{Z}^*$ denotes the kleene closure of the set $\mathcal{Z}$ and $\prod_i^I \mathcal{Z}_i$ denotes the cartesian product.

2. PHYSICALLY COUPLED CONTROLLED SUBSYSTEMS

2.1 Physically coupled plants

Each controlled subsystem $\bar{\Sigma}_i$ consists of a plant $\Sigma_{\mathrm{P}i}$ and a state-feedback controller $\Sigma_{\mathrm{C}i}$ (Fig. 1(b)). The plant is modelled by a deterministic extended Input/Output (I/O)-automaton (Lunze, 2017) $\Sigma_{\mathrm{P}i} = (\mathcal{Z}_i, \mathcal{V}_i, \mathcal{S}_i, \mathcal{R}_i, G_i, F_i, z_{i0})$

$$\Sigma_{\mathrm{P}i} : \begin{cases} z_i(k+1) = G_i(z_i(k), v_i(k), s_i(k)),\ z_i(0) = z_{i0} \\ r_i(k) = F_i(z_i(k), v_i(k), s_i(k)) \end{cases} \quad (1)$$

with the state set $\mathcal{Z}_i$, the input set $\mathcal{V}_i$, the coupling input set $\mathcal{S}_i$ and the coupling output set $\mathcal{R}_i$. The state transition function $G_i : \mathcal{Z}_i \times \mathcal{V}_i \times \mathcal{S}_i \to \mathcal{Z}_i$ maps the current state $z_i(k)$, the current input $v_i(k)$ and the coupling input $s_i(k)$ at time step k to the next state $z_i(k+1)$, whereas the coupling function $F_i : \mathcal{Z}_i \times \mathcal{V}_i \times \mathcal{S}_i \to \mathcal{R}_i$ maps the same arguments to the current value of the coupling output $r_i(k)$. In the following, the current state $z_i(k)$ is assumed to be measurable and the next state $z_i(k+1)$ is denoted by z_i'.

Physically coupled state transitions are represented as *synchronous state transitions* having a coupling input $s_i \neq \varepsilon$, whereas physically uncoupled state transitions are denoted as *autonomous state transitions* having the empty symbol ε as the coupling input $s_i = \varepsilon$. The description of the synchronous state transitions is similar to the notion of γ introduced by Zgorzelski and Lunze (2019, 2018b), however, in this paper, the generalised description with extended I/O-automata is applied.

The couplings between the plants (1) are described by the coupling function $K : \prod_{i=1}^I \mathcal{R}_i \to \prod_{i=1}^I \mathcal{S}_i$

$$\Sigma_\mathrm{K} : (s_1(k), \dots, s_I(k))^\mathrm{T} = K((r_1(k), \dots, r_I(k))^\mathrm{T}), \quad (2)$$

which maps all coupling outputs $r_i(k)$ to the coupling inputs $s_i(k)$, whereas the component $K_i : \prod_{i=1}^I \mathcal{R}_i \to \mathcal{S}_i$ of (2) $\Sigma_{\mathrm{K}i} : s_i(k) = K_i((r_1(k), \dots, r_I(k))^\mathrm{T})$ does the same for the i-th subsystem.

2.2 Local state-feedback controller

The plants $\Sigma_{\mathrm{P}i}$ are connected to a local state-feedback controller

$$\Sigma_{\mathrm{C}i} : v_i(k) = C_i(z_i(k), e_i(k)) \quad (3)$$

with the controller function $C_i : \mathcal{Z}_i \times \mathcal{Z}_i \to \mathcal{V}_i$ that maps the current state $z_i(k)$ and the control input $e_i(k)$ to an input symbol $v_i(k)$ for the plant (1). The controller (3) has the control aim to steer the plant (1) into the desired target state $z_{\mathrm{F}i} \in \mathcal{Z}_i$ given to the control input $e_i(k) = z_{\mathrm{F}i}$.

The controller function C_i is designed by the Algorithm 1 with the active input set $\mathcal{V}_i(z_i, z_i') = \{v_i \mid \exists s_i \in \mathcal{S}_i : z_i' = G_i(z_i, v_i, s_i)\}$.

Algorithm 1 Local state-feedback controller design

Input: Plant $\Sigma_{\mathrm{P}i}$
For $\forall z_i, z_{\mathrm{F}i} \in \mathcal{Z}_i$ **do**
(1) Search shortest state sequence: $Z_i(0 \dots k_\mathrm{e}) = (z_i(0) = z_i, z_i(1) = z_i' \dots, z(k_\mathrm{e}) = z_{\mathrm{F}i})$
(2) Define: $C_i(z_i, z_{\mathrm{F}i}) \in \mathcal{V}_i(z_i, z_i')$
Result: Controller function C_i

The controlled subsystem $\bar{\Sigma}_i = (\mathcal{Z}_i, \mathcal{V}_i, \mathcal{S}_i, \mathcal{R}_i, \bar{G}_i, \bar{F}_i, z_{i0})$

$$\bar{\Sigma}_i : \begin{cases} z_i(k+1) = \bar{G}_i(z_i(k), e_i(k), s_i(k)),\ z_i(0) = z_{i0} \\ r_i(k) = \bar{F}_i(z_i(k), e_i(k), s_i(k)) \end{cases} \quad (4)$$

with $\bar{G}_i(z_i, e_i, s_i) = G_i(z_i, C_i(z_i, e_i), s_i)$ and $\bar{F}_i(z_i, e_i, s_i) = F_i(z_i, C_i(z_i, e_i), s_i)$ is obtained by the connection of the plant (1) with the state-feedback controller (3).

The set of all local state transitions is given by

$$\mathcal{G}_i = \{(z_i \to e_i) \mid \exists s_i \in \mathcal{S}_i : e_i = \bar{G}_i(z_i, e_i, s_i)\}$$

and the set of all coupling output signals r_i of a local state transition $(z_i \to z_i')$ is described by the active coupling output set

$$\mathcal{R}_i(z_i, z_i') = \{r_i \mid \exists s_i \in \mathcal{S}_i : r_i = \bar{F}_i(z_i, z_i', s_i) \wedge z_i' = \bar{G}_i(z_i, z_i', s_i)\}.$$

The overall state transition function $\bar{G}$ and coupling function $\bar{F}$ of all controlled subsystems are denoted by

$$\bar{G}(\boldsymbol{z}, \boldsymbol{e}, \boldsymbol{s}) = \begin{pmatrix} \bar{G}_1(z_1, e_1, s_1) \\ \vdots \\ \bar{G}_I(z_I, e_I, s_I) \end{pmatrix}, \bar{F}(\boldsymbol{z}, \boldsymbol{e}, \boldsymbol{s}) = \begin{pmatrix} \bar{F}_1(z_1, e_1, s_1) \\ \vdots \\ \bar{F}_I(z_I, e_I, s_I) \end{pmatrix}$$

with $\boldsymbol{z} = (z_1, \ldots, z_I)^{\mathrm{T}}$, $\boldsymbol{e} = (e_1, \ldots, e_I)^{\mathrm{T}}$, $\boldsymbol{s} = (s_1, \ldots, s_I)^{\mathrm{T}}$ and the overall state set is given by $\mathcal{Z} = \prod_{i=1}^{I} \mathcal{Z}_i$.

The active coupling input set is defined by

$$\mathcal{S}_i(z_i, z_i') = \{s_i \mid z_i' = \bar{G}_i(z_i, z_i', s_i)\} \quad (5)$$

and the following behaviour is assumed that the plant remain in the current state for the wrong coupling input:

Assumption 1. $\bar{G}_i(z_i, e_i, s_i) = z_i$ if $s_i \notin \mathcal{S}_i(z_i, e_i)$

In the proof of the following Lemma 2, it is shown that the local control objective is met by the controlled subsystem (4), if the coupling input $s_i(k)$ is an element of the active coupling input set (5).

Lemma 2. (Local target state control). In the controlled subsystem (4) with the control input $e_i(k) = z_{\mathrm{F}i}, k \geq 0$, the relation

$$\forall z_i, z_{\mathrm{F}i} \in \mathcal{Z}_i, \exists k_{\mathrm{e}} : z_i(k_{\mathrm{e}}) = z_{\mathrm{F}i}$$
$$\text{holds if } \forall k \leq k_{\mathrm{e}} : s_i(k) \in \mathcal{S}_i(z_i(k), z_i(k+1)). \quad (6)$$

Proof. The proof is given by induction. For all state pairs $z_i, z_{\mathrm{F}i} \in \mathcal{Z}_i$ and its connecting shortest state sequences $Z_i(0 \ldots k_{\mathrm{e}}) = (z_i(0) = z_i, \ldots, z(k_{\mathrm{e}}) = z_{\mathrm{F}i})$ in $\Sigma_{\mathrm{P}i}$, it holds $z_i(k+1) = \bar{G}_i(z_i(k), e_i(k), s_i(k))$ in the controlled subsystem $\bar{\Sigma}_i$ with $e_i(k) = z_{\mathrm{F}i}$, if $s_i(k) \in \mathcal{S}_i(z_i(k), z_i(k+1))$ holds. Base case $k = 0$: $z_i(1) = \bar{G}_i(z_i(0), z_{\mathrm{F}i}, s_i(0))$ if $s_i(0) \in \mathcal{S}_i(z_i(0), z_i(1))$. Induction case $k \to k+1$: From the optimality principle (Bellman, 1957) it follows that each shortest path includes subpaths that are the shortest subpaths. Hence, it holds that $z_i(k+2) = \bar{G}_i(z_i(k+1), z_{\mathrm{F}i}, s_i(k+1))$ if $s_i(k+1) \in \mathcal{S}_i(z_i(k+1), z_i(k+2))$, which concludes the proof. □

2.3 Cooperative control problem

As shown in Lemma 2, the local control aim is satisfied only if the physical coupling input $s_i(k)$ of each local subsystem (4) is an element of the active coupling input set (5). $s_i(k)$ depends on the synchronous state transitions with other subsystems as explained below.

The coupling input $s_i(k)$ of each subsystem depends on the coupling output $r_i(k)$ of their neighbouring subsystems according to the physical coupling network (2). Furthermore, the current value $r_i(k)$ of each subsystem is determined by the coupling output function F_i, which, in turn, depends on the execution of synchronous state transitions with other subsystems. These synchronous state transitions have to be executed by the subsystems to solve physical restrictions before reaching their local target states $z_{\mathrm{F}i}$.

Hence, the controlled subsystems cooperatively have to change their local state trajectories $Z_i(0 \ldots k_{\mathrm{e}}) = (z_i, \ldots, z_{\mathrm{F}i})$ from the current state into their target state to include all necessary synchronous state transitions to reach the required state $z_{\mathrm{F}i}$, respectively.

Problem 3. (Cooperative control problem). In the networked discrete-event system, each controlled subsystem (4) has to reach its individual local target state:

$$\forall \begin{pmatrix} z_{10} \\ \vdots \\ z_{I0} \end{pmatrix}, \begin{pmatrix} z_{\mathrm{F}1} \\ \vdots \\ z_{\mathrm{F}I} \end{pmatrix} \in \mathcal{Z}, \exists \begin{pmatrix} k_{e1} \\ \vdots \\ k_{eI} \end{pmatrix} : \begin{pmatrix} z_1(k_{e1}) \\ \vdots \\ z_I(k_{eI}) \end{pmatrix} = \begin{pmatrix} z_{\mathrm{F}1} \\ \vdots \\ z_{\mathrm{F}I} \end{pmatrix}.$$

To solve the cooperative control problem, this paper applies the network unit Σ_i' introduced by Zgorzelski and Lunze (2019) and extends Σ_i' for its application to an arbitrary number of subsystems. The network units Σ_i' with the control input u_i are connected to each controlled subsystem (4) and they are able to exchange information via the communication network Σ_{N} (Fig. 1(a)). Depending on the current states $z_i(k)$ and the target states $z_{\mathrm{F}i}$ being set to the control inputs $u_i(k) = z_{\mathrm{F}i}$, the network units Σ_i' temporarily modify the local target states $e_i(k) = \tilde{z}_{\mathrm{F}i}$ $(z_{\mathrm{F}i} \neq \tilde{z}_{\mathrm{F}i})$ for their controlled subsystems $\bar{\Sigma}_i$ to cooperatively change the local state trajectories $Z_i(0 \ldots k_{\mathrm{e}})$ such that all necessary synchronous state transitions are executed to satisfy the Problem 3.

3. COOPERATIVE STATE-FEEDBACK CONTROL

3.1 Model abstraction

The challenge to solve Problem 3 is to ensure the execution of the synchronous state transitions by modifying the local target states. The main idea is to describe the cooperative behaviour of the controlled subsystems by a reduced overall model $\tilde{\Sigma}$. The modified target states $\tilde{z}_{\mathrm{F}i}$ that lead to the new state trajectories solving Problem 3 are selected from a state sequence $\tilde{Z}(0 \ldots k_{\mathrm{e}})$ in the abstracted model $\tilde{\Sigma}$. To obtain the simplified overall model, each subsystem is reduced to an abstracted model based on an equivalence relation and all abstracted subsystem models are combined to the abstracted overall model afterwards.

The equivalence relation

$$z_i \sim \hat{z}_i \text{ if } P_i(z) = P_i(\tilde{z}_i) \quad (7)$$

with $P_i(z_i) = \{\tilde{z}_i \mid \exists Z_i(0 \ldots k_{\mathrm{e}}) = (z_i, \ldots, \tilde{z}_i) \in \mathcal{Z}_i^* : z_i(k+1) = \bar{G}_i(z_i(k), \tilde{z}_i, \varepsilon)\}$ defines local states $z_i, \tilde{z}_i \in \mathcal{Z}_i$ to be equivalent, if the states are strongly connected by state sequences containing only autonomous state transitions. The resulting equivalence classes

$$[z_i] = \{\hat{z}_i \mid z_i \sim \hat{z}_i\} \quad (8)$$

partition the local state sets $\mathcal{Z}_i$ into disjoint subsets $\mathcal{Z}_i^m \subseteq \mathcal{Z}_i (m = 1, 2, \ldots, M)$. The states of the abstracted local models are defined to be the abstracted state set $\tilde{\mathcal{Z}} = [\mathcal{Z}_i] = \{[z_i] \mid \forall \hat{z}_i \in \mathcal{Z}_i : \hat{z}_i \in [z_i]\}$ resulting form the equivalence classes (8).

The local abstracted model is defined as an extended I/O-automaton $\tilde{\Sigma}_i = (\tilde{\mathcal{Z}}_i, \tilde{\mathcal{V}}_i, \tilde{\mathcal{S}}_i, \tilde{\mathcal{R}}_i, \tilde{G}_i, \tilde{F}_i, \tilde{z}_{i0})$

$$\tilde{\Sigma}_i : \begin{cases} \tilde{z}_i(k+1) = \tilde{G}_i(\tilde{z}_i(k), \tilde{e}_i(k), \tilde{s}_i(k)), \quad \tilde{z}_i(0) = \tilde{z}_{i0} \\ \tilde{r}_i(k) = \tilde{F}_i(\tilde{z}_i(k), \tilde{e}_i(k), \tilde{s}_i(k)) \end{cases} \tag{9}$$

with $\tilde{\mathcal{Z}}_i = \tilde{\mathcal{V}}_i = [\mathcal{Z}_i]$, $\tilde{\mathcal{S}}_i = \mathcal{S}_i$ and $\tilde{\mathcal{R}}_i = \mathcal{R}_i$. The state transition function and the coupling function are defined by $\tilde{G}_i([z_i],[e_i],s_i) = [\bar{G}_i(z_i,e_i,s_i)]$ and

$$\tilde{F}_i([z_i],[e_i],s_i) = \begin{cases} r_i, & \text{if } \mathcal{R}_i(z_i,e_i) \neq \emptyset \\ \varepsilon, & \text{else,} \end{cases}$$

with $r_i \in \mathcal{R}_i(z_i,e_i)$. $[\bar{G}_i(\ldots)]$ denotes the equivalence class of the codomain of the function $\bar{G}_i$. The set of all local abstracted state transitions is given by

$$\tilde{\mathcal{G}}_i = \{([z_i] \to [e_i]) \mid \exists s_i \in \mathcal{S}_i : e_i = \bar{G}_i(z_i,e_i,s_i)\}$$

and the state transition function and coupling output function of all abstracted models are denoted by

$$\tilde{G}(\tilde{\boldsymbol{z}},\tilde{\boldsymbol{e}},\tilde{\boldsymbol{s}}) = \begin{pmatrix} \tilde{G}_1(\tilde{z}_1,\tilde{e}_1,\tilde{s}_1) \\ \vdots \\ \tilde{G}_I(\tilde{z}_I,\tilde{e}_I,\tilde{s}_I) \end{pmatrix}, \tilde{F}(\tilde{\boldsymbol{z}},\tilde{\boldsymbol{e}},\tilde{\boldsymbol{s}}) = \begin{pmatrix} \tilde{F}_1(\tilde{z}_1,\tilde{e}_1,\tilde{s}_1) \\ \vdots \\ \tilde{F}_I(\tilde{z}_I,\tilde{e}_I,\tilde{s}_I) \end{pmatrix}.$$

with $\tilde{\boldsymbol{z}} = (\tilde{z}_1,\ldots,\tilde{z}_I)^{\mathrm{T}}$, $\tilde{\boldsymbol{e}} = (\tilde{e}_1,\ldots,\tilde{e}_I)^{\mathrm{T}}$ and $\tilde{\boldsymbol{s}} = (\tilde{s}_1,\ldots,\tilde{s}_I)^{\mathrm{T}}$.

3.2 Properties of the abstracted model

This section investigates the properties of the proposed model abstraction. The question to be answered is under what condition is the abstracted model suitable for the selection of the modified target states to include all necessary synchronous state transitions.

The proof of Lemma 4 will show that the composition of the abstracted subsystem models (9) with the coupling network (2) is equal to the abstraction of the composition of all original controlled subsystems (4), if the local coupling signals of all synchronous state transitions connecting the same abstracted states are equal according to condition (11).

Lemma 4. (Compositional abstraction).

$$\forall([\boldsymbol{z}] \to [\boldsymbol{e}]) \in \prod_{i=1}^{I} \tilde{\mathcal{G}}_i \, \forall(\boldsymbol{z} \to \boldsymbol{e}) \in \prod_{i=1}^{I} \mathcal{G}_i \exists! \boldsymbol{r} \in \prod_{i=1}^{I} \mathcal{R}_i :$$
$$\begin{pmatrix} \tilde{G}([\boldsymbol{z}],[\boldsymbol{e}],K(\boldsymbol{r})) \\ \tilde{F}([\boldsymbol{z}],[\boldsymbol{e}],K(\boldsymbol{r})) \end{pmatrix} = \begin{pmatrix} [\bar{G}(z,e,K(\boldsymbol{r}))] \\ \bar{F}(z,e,K(\boldsymbol{r})) \end{pmatrix}, \tag{10}$$

$$\text{if } \bigwedge_{i=1}^{I} (\mathcal{R}_i(z_i,e_i) = \{r_i\} \wedge \mathcal{S}_i(z_i,e_i) = \{K_i(\boldsymbol{r})\}). \tag{11}$$

with $\boldsymbol{r} = (r_1,\ldots,r_I)^{\mathrm{T}}$.

Proof.

$$\forall([\boldsymbol{z}] \to [\boldsymbol{e}]) \in \prod_{i=1}^{I} \tilde{\mathcal{G}}_i \, \forall(\boldsymbol{z} \to \boldsymbol{e}) \in \prod_{i=1}^{I} \mathcal{G}_i \exists! \boldsymbol{r} \in \prod_{i=1}^{I} \mathcal{R}_i :$$
$$\bigwedge_{i=1}^{I} (\mathcal{R}_i(z_i,e_i) = \{r_i\} \wedge \mathcal{S}_i(z_i,e_i) = \{K_i(\boldsymbol{r})\})$$
$$\Rightarrow \bigwedge_{i=1}^{I} \begin{pmatrix} \tilde{G}_i(\tilde{z}_i,\tilde{z}_i,K_i(\boldsymbol{r})) \\ \tilde{F}_i(\tilde{z}_i,\tilde{z}_i,K_i(\boldsymbol{r})) \end{pmatrix} = \begin{pmatrix} [\bar{G}_i(z_i,e_i,K_i(\boldsymbol{r}))] \\ \bar{F}_i(z_i,e_i,K_i(\boldsymbol{r})) \end{pmatrix}$$
$$\Rightarrow \text{eqn. (10).}$$

□

In the following, the state transitions of the composed abstracted subsystem models (9) are described by the set

$$\tilde{\mathcal{G}} = \{(\tilde{\boldsymbol{z}} \to \tilde{\boldsymbol{e}}) \mid \exists \boldsymbol{r} \in \mathcal{R} : \tilde{\boldsymbol{e}} = \tilde{G}(\tilde{\boldsymbol{z}},\tilde{\boldsymbol{e}},K(\boldsymbol{r})) \wedge \boldsymbol{r} = \tilde{F}(\tilde{\boldsymbol{z}},\tilde{\boldsymbol{e}},K(\boldsymbol{r}))\}. \tag{12}$$

In the proof of Lemma 5 it is shown, that under the condition (11), all original state transitions $\prod_{i=1}^{I} \mathcal{G}_i$ that correspond to an abstracted state transition (12) are feasible in the local controlled subsystems (4), which means that the controlled subsystem (4) can follow each abstracted state transition because it leads to state transitions satisfying condition (6) in Lemma 2.

Lemma 5. (Feasibility of abstracted state transitions).

$$\forall([\boldsymbol{z}] \to [\boldsymbol{e}]) \in \prod_{i=1}^{I} \tilde{\mathcal{G}}_i \, \forall(\boldsymbol{z} \to \boldsymbol{e}) \in \prod_{i=1}^{I} \mathcal{G}_i \exists! \boldsymbol{r} \in \prod_{i=1}^{I} \mathcal{R}_i :$$
$$\boldsymbol{e} = \bar{G}(\boldsymbol{z},\boldsymbol{e},K(\boldsymbol{r})) \wedge \boldsymbol{r} = \bar{F}(\boldsymbol{z},\boldsymbol{e},K(\boldsymbol{r})) \tag{13}$$

if condition (11) holds.

Proof.

$$\forall([\boldsymbol{z}] \to [\boldsymbol{e}]) \in \prod_{i=1}^{I} \tilde{\mathcal{G}}_i \, \forall(\boldsymbol{z} \to \boldsymbol{e}) \in \prod_{i=1}^{I} \mathcal{G}_i \exists! \boldsymbol{r} \in \prod_{i=1}^{I} \mathcal{R}_i :$$
$$\bigwedge_{i=1}^{I} (\mathcal{R}_i(\tilde{z}_i,\tilde{e}_i) = \{r_i\} \wedge \mathcal{S}_i(\tilde{z}_i,\tilde{e}_i) = \{K_i(\boldsymbol{r})\})$$
$$\Rightarrow \bigwedge_{i=1}^{I} e_i = \bar{G}_i(z_i,e_i,K_i(\boldsymbol{r})) \wedge r_i = \bar{F}_i(z_i,e_i,K_i(\boldsymbol{r}))$$
$$\Rightarrow \text{eqn. (13).}$$

□

In the next subsection, an abstracted state-feedback controller is designed to determine state sequences in the abstracted model for selecting the temporary modified target states.

3.3 Abstracted state-feedback controller

The abstract model $\tilde{\Sigma}$ is obtained from the composition of the abstracted subsystem models (9) with the coupling network (2). $\tilde{\Sigma} = (\tilde{\mathcal{Z}},\tilde{\mathcal{V}},\tilde{G}_i,\tilde{z}_0)$ is a deterministic I/O-automaton

$$\tilde{\Sigma} : \tilde{z}(k+1) = \tilde{G}(\tilde{z}(k),\tilde{e}(k)), \quad \tilde{z}(0) = \tilde{z}_0 \tag{14}$$

with $\tilde{\mathcal{Z}} = \tilde{\mathcal{V}} = \prod_{i=1}^{I} \tilde{\mathcal{Z}}_i$ and the measurable abstracted current state $\tilde{z}(k)$.

To obtain a state sequence from the current abstracted state $\tilde{z}(k) = [z(k)]$ into the desired abstracted target state $\tilde{z}_{\mathrm{F}} = [z_{\mathrm{F}}]$, a state-feedback controller

$$\tilde{\Sigma}_{\mathrm{C}} : \tilde{e}(k) = \tilde{C}(\tilde{z}(k),\tilde{u}(k)) \tag{15}$$

is designed by applying the method from Zgorzelski and Lunze (2017) with the abstracted controller function $\tilde{C} : \tilde{\mathcal{Z}} \times \tilde{\mathcal{Z}} \to \tilde{\mathcal{Z}}$. The controller function $\tilde{C}$ maps the current abstracted state $\tilde{z}(k) = \tilde{z}$ and the abstracted target state $\tilde{u}(k) = \tilde{z}_{\mathrm{F}}$ to the successor state $\tilde{e}(k) = \tilde{z}'$ of $\tilde{z}$ in the shortest state sequence $\tilde{Z}(0 \ldots k_{\mathrm{e}}) = (\tilde{z},\tilde{z}',\ldots,\tilde{z}_{\mathrm{F}})$ from $\tilde{z}$ towards $\tilde{z}_{\mathrm{F}}$ in the abstracted model (14).

3.4 Network unit

The state-feedback controller (15) is decomposed into components

$$\tilde{\Sigma}_{\mathrm{C}i} : \tilde{e}_i(k) = \tilde{C}_i(\tilde{z}(k), \tilde{u}(k)) \tag{16}$$

with the local controller functions $\tilde{C}_i : \tilde{\mathcal{Z}} \times \tilde{\mathcal{Z}} \to \tilde{\mathcal{Z}}_i$. Each controller (16) is applied by the network unit Σ'_i belonging to the subsystem as shown in Fig. 1(a).

The network units

$$\Sigma'_i : \begin{cases} e_i(k) = u_i(k), & \text{if } [z(k)] = [u(k)] \\ e_i(k) \in \tilde{C}_i(\tilde{z}(k), \tilde{u}(k)), & \text{else} \end{cases} \tag{17}$$

are defined to have an autonomous and a cooperative mode. In the autonomous mode ($e_i(k) = u_i(k)$), any controlled subsystem has to execute synchronous state transitions ($[z(k)] = [u(k)]$) to reach its target state and no cooperation is needed between the subsystems. However, in the cooperative mode ($e_i(k) \in \tilde{C}_i(\tilde{z}(k), \tilde{u}(k))$), the controlled subsystems have to execute synchronous state transitions ($[z(k)] \neq [u(k)]$) and the network unit (17) applies the controller (16) and connects all controlled subsystems (4) to the cooperating subsystems

$$\hat{\Sigma}_i : \begin{cases} z_i(k+1) = \bar{G}_i(z_i(k), e_i(k), s_i(k)),\ z_i(0) = z_{i0} \\ r_i(k) = \bar{F}_i(z_i(k), e_i(k), s_i(k)) \\ e_i(k) \in \tilde{C}_i(\tilde{z}(k), \tilde{u}(k)) \end{cases} \tag{18}$$

by selecting the corresponding local target states $e_i(k) = z_{\mathrm{F}i} \in \tilde{e}_i(k)$ from its output by and by exchanging their abstracted current states $\tilde{z}(k) = [z(k)]$ and their target states $\tilde{u}(k) = [u(k)]$ via the communication network Σ_{N} (Fig. 1(a)).

Following from Lemma 5, Lemma 6 states that in the cooperative subsystems (18), the abstracted target states $\tilde{u}_i(k) = \tilde{z}_{\mathrm{F}i}$ are reached after a finite time step $k_{\mathrm{e}i}$.

Lemma 6. (Abstracted state-feedback control). For the digitally connected cooperating subsystem (18) with the control input $u_i(k) = z_{\mathrm{F}i}$, $k \geq 0$, it holds that

$$\forall \begin{pmatrix} z_{10} \\ \vdots \\ z_{I0} \end{pmatrix}, \begin{pmatrix} z_{\mathrm{F}1} \\ \vdots \\ z_{\mathrm{F}I} \end{pmatrix} \in \mathcal{Z}, \exists \begin{pmatrix} k_{\mathrm{e}1} \\ \vdots \\ k_{\mathrm{e}I} \end{pmatrix} : \begin{pmatrix} [z_1(k_{\mathrm{e}1})] \\ \vdots \\ [z_I(k_{\mathrm{e}I})] \end{pmatrix} = \begin{pmatrix} [z_{\mathrm{F}1}] \\ \vdots \\ [z_{\mathrm{F}I}] \end{pmatrix}$$

if condition (11) holds and the abstracted state set $\tilde{\mathcal{Z}} = [\mathcal{Z}]$ in the abstracted model (14) is strongly connected.

Proof. Similar to the proof of Lemma 2. □

Finally, the Theorem 7 follows from Lemma 2 and Lemma 6 and states the main result:

Theorem 7. (Cooperative target state control). In the networked discrete-event system

$$\hat{\Sigma} = (\bar{\Sigma}_1, \ldots, \bar{\Sigma}_I, \Sigma_{\mathrm{K}}, \Sigma'_1, \ldots, \Sigma'_I, \Sigma_{\mathrm{N}})$$

in Fig. 1(a), the Problem 3 is solved if condition (11) is satisfied and if the abstracted state set $\tilde{\mathcal{Z}} = [\mathcal{Z}]$ in the abstracted model (14) is strongly connected.

4. COOPERATIVE CONTROL OF THE FLEXIBLE MANUFACTURING SYSTEM

4.1 Model of the flexible manufacturing system

In this experimental setup, we consider the flexible manufacturing system (see Section 1.1) with three workpieces highlighted in Fig. 2 in grey, red and green. For simplification we reduce each subsystem to handle only one workpiece at the same time. Each subsystem $\Sigma_{\mathrm{P}i}$ is modelled by three extended I/O-automata $\Sigma_{\mathrm{P}ij} = (\mathcal{Z}_{ij}, \mathcal{V}_{ij}, \mathcal{S}_{ij}, \mathcal{R}_{ij}, G_{ij}, F_{ij}, z_{ij0})$ $(j = 1, 2, 3)$

$$\Sigma_{\mathrm{P}ij} : \begin{cases} z'_{ij} = G_{ij}(z_{ij}, v_{ij}, s^{\mathrm{L}}_{ij}, s_{ij}),\ z_{ij}(0) = z_{ij0} \\ r^{\mathrm{L}}_{ij} = F^{\mathrm{L}}_{ij}(z_{ij}, v_{ij}, s^{\mathrm{L}}_{ij}, s_{ij}) \\ r_{ij} = F_{ij}(z_{ij}, v_{ij}, s^{\mathrm{L}}_{ij}, s_{ij}) \end{cases} \tag{19}$$

that are connected to the subsystems (1) by physical couplings

$$\Sigma^{\mathrm{L}}_{\mathrm{K}i} : (s^{\mathrm{L}}_{i1}, \ldots, s^{\mathrm{L}}_{iJ})^{\mathrm{T}} = K^{\mathrm{L}}_i((r^{\mathrm{L}}_{i1}, \ldots, r^{\mathrm{L}}_{iJ})^{\mathrm{T}}) \tag{20}$$

with the coupling function $K^{\mathrm{L}}_i : \prod_{j=1}^{3} \mathcal{R}_{ij} \to \prod_{j=1}^{3} \mathcal{S}_{ij}$ (Fig. 3). In the following, all sets are modelled by natural

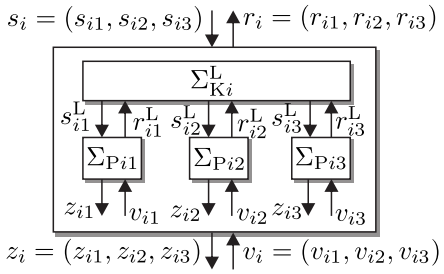


Fig. 3. Block diagram of local subsystem $\Sigma_{\mathrm{P}i}$

numbers $\mathcal{Z}_{ij}, \mathcal{V}_{ij}, \mathcal{R}_{ij}, \mathcal{S}_{ij} \subseteq \mathbb{N}_0$, which are noted as vectors in the composed subsystem (1). Hence, the local coupling functions (20) are modelled by coupling matrices $\boldsymbol{K}^{\mathrm{L}}_i$ and the local coupling relation is written as

$$(s^{\mathrm{L}}_{i1}, \ldots, s^{\mathrm{L}}_{iJ})^{\mathrm{T}} = \boldsymbol{K}^{\mathrm{L}}_i(r^{\mathrm{L}}_{i1}, \ldots, r^{\mathrm{L}}_{iJ})^{\mathrm{T}} \tag{21}$$

for each subsystem.

The area gantry robot $\Sigma_{\mathrm{P}1}$, the line gantry robot $\Sigma_{\mathrm{P}2}$ and the 3-axis robot $\Sigma_{\mathrm{P}3}$ are composed of the extended I/O-automata (19) shown in Figs. 4-6 (abbreviations: gripper i (Gi), handler i (Hi), workpiece i (Wi), magazine (M), conveyor belt (C) and position i (Pi)).

Gripper model $\Sigma_{\mathrm{P}i1}$: Pneumatic gripper for gripping workpieces with

$\mathcal{Z}_{\mathrm{P}11}$={1≙G1 up without W1, 2≙G1 up with W1 gripped, 3≙G1 down without W1, 4≙G1 down with W1 gripped},

$\mathcal{Z}_{\mathrm{P}21}$={1≙G2 up without W2, 2≙G2 up with W2 gripped, 3≙G2 down without W2, 4≙G2 down with W2 gripped},

$\mathcal{Z}_{\mathrm{P}31}$={1≙G3 up without W3, 2≙G3 up with W3 gripped, 3≙G3 down without W3, 4≙G3 down with W3 gripped}.

Handler model $\Sigma_{\mathrm{P}i2}$**:** Handler for the movement of the gripper with

$\mathcal{Z}_{\mathrm{P}12}$={1≙G1 over the M, 2≙G1 above the C, 3≙G1 above storage position 1, 4≙G1 above storage position 2, 5≙G1 above waiting position},

$\mathcal{Z}_{\mathrm{P}22}$={1≙G2 over the waiting position, 2≙G2 at handover position, 3≙G2 above M, 4≙G2 above storage position 1, 5≙G2 above storage position 2},

$\mathcal{Z}_{\mathrm{P}32}$={1≙G3 above the C, 2≙ G3 above P1, 3≙G3 above P2, 4≙G3 above HP}.

Workpiece model $\Sigma_{\mathrm{P}i3}$**:** Position of the workpiece with

$\mathcal{Z}_{\mathrm{P}13}$={1≙No W1 in subsystem, 2≙W1 at C and at G1, 3≙W1 at C, 4≙W1 at storage position 2, 5≙W1 at storage position 1, 6≙W1 at the G1},

$\mathcal{Z}_{\mathrm{P23}}$={1≙W2 at G2, 2≙W2 at handover position, 3≙No W2 in subsystem, 4≙W2 at storage position 1, 5≙W2 at storage position 2},

$\mathcal{Z}_{\mathrm{P33}}$={1≙ W in M, 2≙W3 at G3, 3≙W3 at HP of G2, 4≙W3 at P1, 5≙No W3 in subsystem, 6≙W3 at HP of C, 7≙W3 at P2}.

The conveyor belt is composed of the motor model Σ_{P41}, the sensor model of the belt Σ_{P42} and the workpiece model Σ_{P42} (Fig. 7).

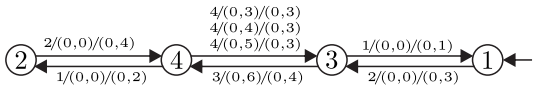
Conveyor belt models: Motor, belt and workpieces with

$\mathcal{Z}_{\mathrm{P41}}$={1≙Motor on, 2≙Motor off},

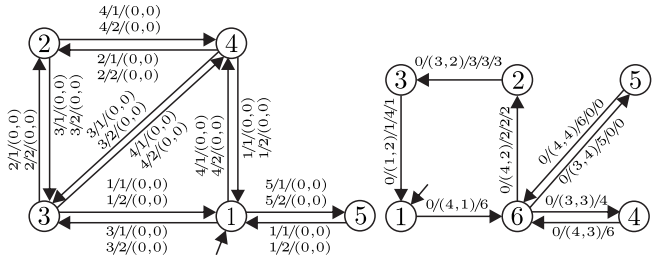
$\mathcal{Z}_{\mathrm{P42}}$={1≙C empty, 2≙W4 at (HP1), 3≙W4 at position 2, 4≙W4 at position 3, 5≙W4 at HP2},

$\mathcal{Z}_{\mathrm{P43}}$={1≙No WP4 in subsystem, 2≙W4 at G1 at HP1, 3≙W4 on C at HP1, 4≙W4 at position 2, 5≙ W4 at position 3, 6≙W4 at HP2}.

Due to the space limitation, the description of the signals cannot be given.

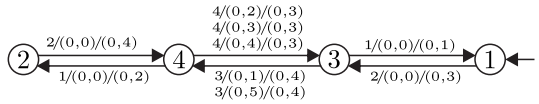


(a) Σ_{P11} (Gripper)

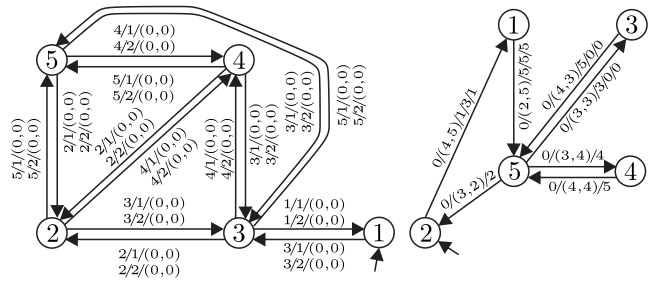


(b) Σ_{P12} (Handler) (c) Σ_{P13} (Workpiece)

Fig. 4. Area gantry robot Σ_{P1}



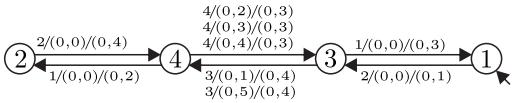
(a) Σ_{P21} (Gripper)



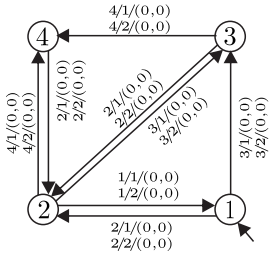
(b) Σ_{P22} (Handler) (c) Σ_{P23} (Worpiece)

Fig. 5. Line gantry robot Σ_{P2}

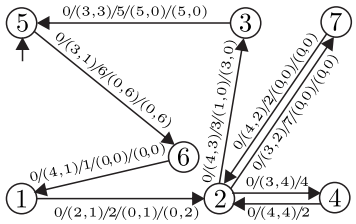
The state transitions of the extended I/O-automata in Figs. 4-7 are represented in the form $v_{ij}/s^{\mathrm{L}}_{ij}/r^{\mathrm{L}}_{ij}/s_i/r_i$ with $v_{ij}/s^{\mathrm{L}}_{ij}/r^{\mathrm{L}}_{ij}$ being a short form for $v_{ij}/s^{\mathrm{L}}_{ij}/r^{\mathrm{L}}_{ij}/\varepsilon/\varepsilon$, respectively, with the entry 0 representing the empty symbol ε. Moreover, each state z_{ij} has a self-loop state transition



(a) Σ_{P31} (Gripper)

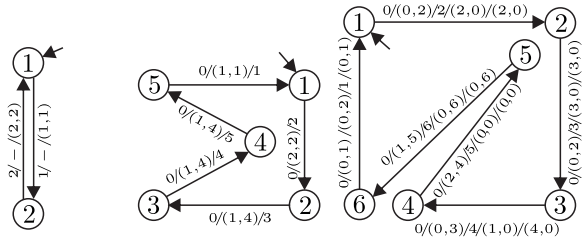


(b) Σ_{P32} (Handler)



(c) Σ_{P33} (Workpiece)

Fig. 6. 3-axis robot Σ_{P3}



(a) Σ_{P41} (Motor) (b) Σ_{P42} (Belt) (c) Σ_{P43} (Workpiece)

Fig. 7. Conveyor belt Σ_{P4}

with the entry $v_{ij}/\varepsilon/z_{ij}/\varepsilon/z_{ij}$. The local coupling input vectors $s^{\mathrm{L}}_i = (s^{\mathrm{L}}_{i1}, s^{\mathrm{L}}_{i2}, s^{\mathrm{L}}_{i3})^{\mathrm{T}}$ and output vectors $r^{\mathrm{L}}_i = (r^{\mathrm{L}}_{i1}, r^{\mathrm{L}}_{i2}, r^{\mathrm{L}}_{i3})^{\mathrm{T}}$ are connected according to eqn. (21) with the local coupling matrices

$$\boldsymbol{K}^{\mathrm{L}}_1 = \begin{pmatrix} 0&0&1&0&0\\ 0&0&0&0&1\\ 1&0&0&0&0\\ 0&1&0&0&0\\ 0&0&0&1&0 \end{pmatrix}, \boldsymbol{K}^{\mathrm{L}}_2 = \begin{pmatrix} 0&0&1&0&0\\ 0&0&0&0&1\\ 1&0&0&0&0\\ 0&1&0&0&0\\ 0&0&0&1&0 \end{pmatrix},$$

$$\boldsymbol{K}^{\mathrm{L}}_3 = \begin{pmatrix} 0&0&1&0&0\\ 0&0&0&0&1\\ 1&0&0&0&0\\ 0&1&0&0&0\\ 0&0&0&1&0 \end{pmatrix} \text{ and } \boldsymbol{K}^{\mathrm{L}}_4 = \begin{pmatrix} 1&0&0&0\\ 0&0&0&1\\ 0&1&0&0\\ 0&0&1&0 \end{pmatrix}.$$

In the following section, the states of each subsystem (1) are encoded from of the local states of the subsystems (19) and decoded with

$$z_i = \sum_{j=1}^{3} z_{ij} \cdot 10^{j-1} \text{ and } z_{ij} = \left\lfloor \frac{z_i \bmod 10^j}{10^{j-1}} \right\rfloor \qquad (22)$$

and $\lfloor x \rfloor := \max\{k \in \mathbb{Z} \mid k \leq x\}$.

The global coupling input vectors $s = (s_1, s_2, s_3, s_4)^{\mathrm{T}}$ and coupling output vectors $r = (r_1, r_2, r_3, r_4)^{\mathrm{T}}$ of the subsystems (1) are connected by the physical coupling relation $s = \boldsymbol{K} r$ with

$$\boldsymbol{K} = \begin{pmatrix} 0&0&0&0&1&0\\ 0&0&1&0&0&0\\ 0&1&0&0&0&0\\ 0&0&0&0&0&1\\ 1&0&0&0&0&0\\ 0&0&0&1&0&0 \end{pmatrix}.$$

4.2 Model abstraction

The composed subsystems (1) (Figs. 4-7) are extended to controlled subsystems (4) by local controllers (3), which are designed by applying Algorithm 1. Finally, the controlled subsystems (4) are abstracted with the equivalence relation (7) to the abstracted subsystems (9), whose extended I/O-automata are shown in Fig. 8 with the following disjoint state sets, which are encoded and decoded by applying eqn. (22):

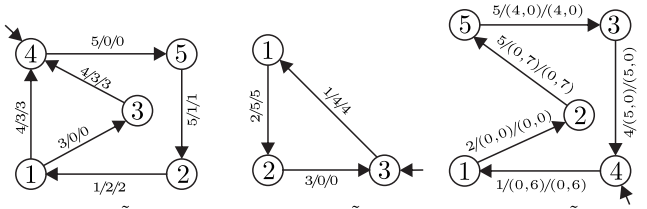
$$\begin{aligned}\mathcal{Z}_1 =&\{323\} \cup \{212, 222, 232, 242, 252, 412, 422, 432,\\ &442, 452\} \cup \{123\} \cup \{111, 121, 131, 141, 151, 311,\\ &321, 331, 341, 351\} \cup \{114, 115, 124, 125, 134, 135,\\ &144, 145, 154, 155, 216, 226, 236, 246, 256, 314, 315,\\ &324, 325, 334, 335, 344, 345, 354, 355, 416, 436, 456\},\end{aligned}$$

$$\begin{aligned}\mathcal{Z}_2 =&\{451\} \cup \{113, 114, 123, 124, 133, 134, 143, 144,\\ &153, 154, 215, 225, 235, 245, 255, 313, 314, 323, 324,\\ &333, 334, 343, 344, 353, 354, 415, 425, 435, 445, 455\}\\ &\cup \{112, 122, 132, 142, 152, 312, 322, 332, 342, 352\},\end{aligned}$$

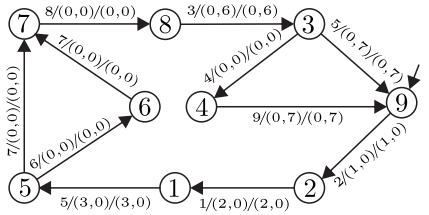
$$\begin{aligned}\mathcal{Z}_3 =&\{316\} \cup \{411\} \cup \{213, 233, 243, 223, 413, 433, 443,\\ &423\} \cup \{115, 135, 145, 125, 315, 335, 345, 325\} \cup\\ &\{114, 117, 134, 137, 144, 147, 124, 127, 212, 232, 242,\\ &222, 314, 317, 334, 337, 344, 347, 324, 327, 412, 442,\\ &422\}\end{aligned}$$

and

$$\begin{aligned}\mathcal{Z}_4 =&\{223\} \cup \{222\} \cup \{156\}, \cup \{256\} \cup \{134\}\\ &\cup \{234\} \cup \{144\} \cup \{245\} \cup \{111, 211\}.\end{aligned}$$



(a) $\tilde{\Sigma}_1$ (Area gantry robot) (b) $\tilde{\Sigma}_2$ (Line gantry robot) (c) $\tilde{\Sigma}_3$ (3-axis robot)



(c) $\tilde{\Sigma}_4$ (Conveyor belt)

Fig. 8. Abstracted controlled subsystems $\tilde{\Sigma}_i$

Note that only the states reachable from the initial states z_{i0} are shown in the disjoint sets above. Moreover, condition (11) is satisfied because only one local subsystem Σ_{Pi3} (Workpiece model) connects the controlled subsystems (4) by physical couplings. Thus, the composed abstracted subsystem models (9) in Fig. 8 satisfy condition (11). In the following experiment, we apply only the strongly connected and reachable abstracted state set $\tilde{\mathcal{Z}}$ of the model (14) satisfying the condition of Theorem 7.

4.3 Experimental results of the cooperative control

This section presents the experimental results of the cooperative control of the flexible manufacturing system HANS (Fig. 2). Each controlled subsystem (4) can perform an autonomous transportation task with a workpiece, however, each subsystem is able to handle only one workpiece at the same time. The area gantry robot Σ_{P1} is able to add new workpieces to the process from a storage and the line gantry robot Σ_{P2} is able to remove workpieces from the process, autonomously.

Moreover, the subsystems can transfer the workpieces from the area gantry robot Σ_{P1} to the conveyor belt Σ_{P4}, from the conveyor belt to the 3-axis robot Σ_{P3} and, finally, from the 3-axis robot to the line gantry robot Σ_{P2} (double arrows in Fig. 2). The cooperative mode of the network unit (17) is necessary whenever any workpiece has to be transferred between two subsystems.

Figure 9(a) presents the graphs of the experiment showing the state output z_i and control input e_i of each controlled subsystem (4) and the control input u_i of the network units (17). Figure 9(b) depicts the active physical couplings between the subsystems Σ_{Pi} as black bars showing when the transfer of the workpices between the subsystems is performed.

In the time interval $t = [0, t_1)$, the network units (17) are in the autonomous mode $e_i = u_i$ (Fig. 9(a)) because each subsystem receives a target state that is reachable without synchronous state transitions and it holds $[z_{i0}] = [u_i]$.

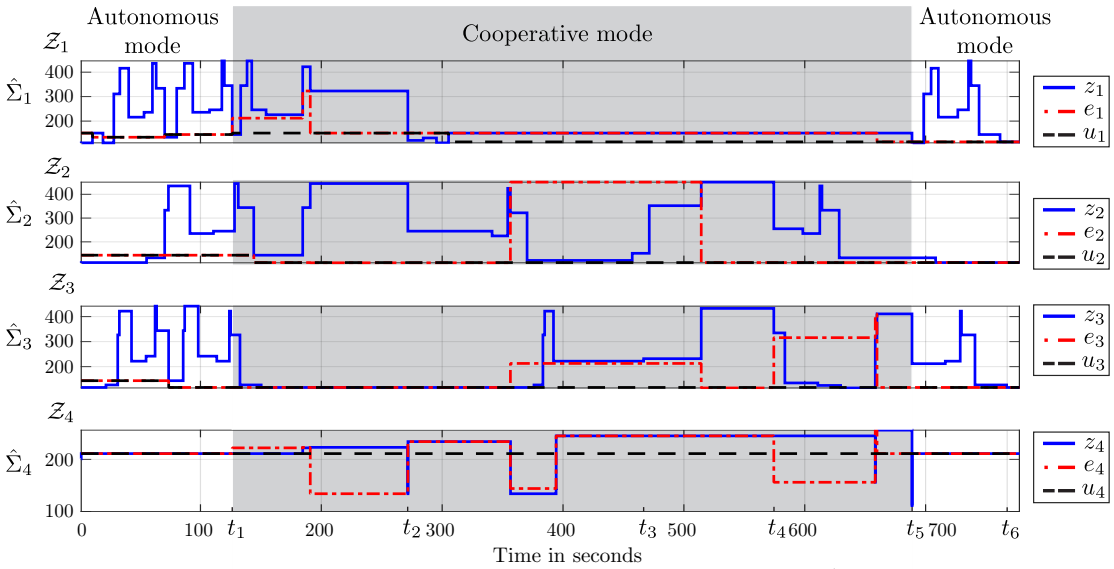
At time $t = t_1$, the area gantry robot receives a new target state to transfer its workpiece to the conveyor belt Σ_{P4}, which necessitates cooperation $[z_1(t_1)] \neq [u_1(t_1)]$ and the network units Σ_i' switch into the cooperative mode according to eqn. (17). As the conveyor belt Σ_{P4} has the local task to remove the workpiece from the belt given by the local target state u_4, it has to transfer the workpiece to the 3-axis robot Σ_{P3}. However, as the 3-axis robot Σ_{P3} handles a workpiece (Fig. 2), Σ_{P3} has to transfer this workpiece to the line gantry robot Σ_{P2}, before Σ_{P3} is ready for the cooperation with Σ_{P4}.

In the physical coupling diagram in Fig. 2(b), it can be seen that the area gantry robot Σ_{P1} transfers the workpiece to the conveyor belt Σ_{P4} in the time interval $t = [t_1, t_2)$. This transfer results from the target states $e_i(t) \neq u_i(t)$ determined by the distributed controller (16) being applied by the network unit (17). Moreover, in the time interval $t = [t_2, t_3)$, the line gantry robot Σ_{P2} brings its workpiece into the storage to be ready for the next cooperation with the 3-axis robot Σ_{P3}, which also results from the dependent target states $e_2(t)$ determined by the abstracted state-feedback controller (Fig. 9(a)). In the time interval $t = [t_3, t_4)$, the application of the abstracted state-feedback controller leads to the transfer of the green workpiece (Fig. 2) from the 3-axis robot Σ_{P3} to the line gantry robot Σ_{P2} (Fig. 9(b)).

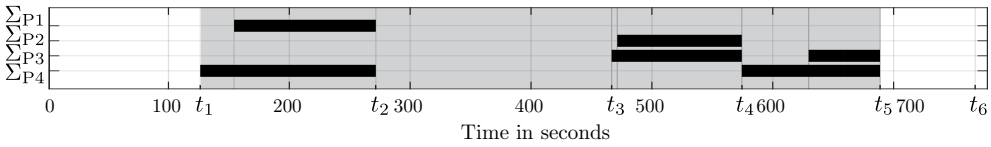
In the time interval $t = [t_4, t_5)$ the conveyor belt Σ_{P4} transfers its workpiece to the 3-axis robot Σ_{P3} (Fig. 9(b)). Finally, at time t_5 all cooperating subsystem switch into the autonomous mode according to eqn. (17) and all subsystems reach their target states $z_i(t_6) = z_i(t_6)$ at time $t = t_6$.

5. CONCLUSION

This paper has presented a cooperative control method for networked discrete-event system. The control aim of each subsystem is to reach a given target state, which necessitates



(a) Input/Output behaviour of the cooperative subsystems $\hat{\Sigma}_i$



(b) Physical interactions between the cooperative subsystems $\hat{\Sigma}_i$

Fig. 9. Experimental results of the cooperative control of the flexible manufacturing systems HANS

cooperation in certain situation due to physical restriction. Each controlled subsystem is extended by a network unit implementing a cooperative state-feedback controller that temporarily organises the necessary cooperation between them with the help of the digital communication network. The proposed method has been applied to a flexible manufacturing system and it has been shown that the cooperative control method for networked discrete-event system is applicable to real-life systems.

REFERENCES

Barrett, G. and Lafortune, S. (2000). Decentralized supervisory control with communicating controllers. *IEEE Transactions on Automatic Control*, 45(9), 1620–1638.

Bellman, R. (1957). *Dynamic Programming*. Princeton University Press, Princeton.

Brave, Y. and Heymann, M. (1989). On stabilization of discrete-event processes. In *Proceedings of the 28th IEEE Conference on Decision and Control,*, 2737–2742 vol.3.

Cai, K. and Wonham, W.M. (2010). Supervisor localization: A top-down approach to distributed control of discrete-event systems. *IEEE Transactions on Automatic Control*, 55(3), 605–618.

Drüppel, S., Lunze, J., and Fritz, M. (2008). Modeling of asynchronous discrete-event systems as networks of input-output automata. *IFAC Proceedings Volumes*, 41(2), 544 – 549. 17th IFAC World Congress.

Lunze, J. (2017). *Ereignisdiskrete Systeme: Modellierung und Analyse dynamischer Systeme mit Automaten, Markovketten und Petrinetzen*. De Gruyter Studium. De Gruyter Oldenbourg, Berlin, 3 edition.

Ramadge, P.J.G. and Wonham, W.M. (1989). The control of discrete event systems. *Proceedings of the IEEE*, 77(1), 81–98.

Rudie, K., Lafortune, S., and Lin, F. (2003). Minimal communication in a distributed discrete-event system. *IEEE Transactions on Automatic Control*, 48(6), 957–975.

Seow, K.T., Pham, M.T., Ma, C., and Yokoo, M. (2009). Coordination planning: Applying control synthesis methods for a class of distributed agents. *IEEE Transactions on Control Systems Technology*, 17(2), 405–415.

Stursberg, O. and Hillmann, C. (2017). Decentralized optimal control of distributed interdependent automata with priority structure. *IEEE Transactions on Automation Science and Engineering*, 14(2), 785–796.

Su, R. and Lennartson, B. (2017). Control protocol synthesis for multi-agent systems with similar actions instantiated from agent and requirement templates. *Automatica*, 79, 244 – 255.

Wang, F., Shu, S., and Lin, F. (2016). Robust networked control of discrete event systems. *IEEE Transactions on Automation Science and Engineering*, 13(4), 1528–1540.

Zgorzelski, M. and Lunze, J. (2017). Feedforward and state-feedback control of deterministic I/O automata. *IFAC-PapersOnLine*, 50(1), 13426 – 13433. 20th IFAC World Congress, Toulouse.

Zgorzelski, M. and Lunze, J. (2018a). Discrete-event demonstrator hans. *IFAC-PapersOnLine*, 51(7), 298 – 303. 14th IFAC Workshop on Discrete Event Systems, Sorrento Coast.

Zgorzelski, M. and Lunze, J. (2018b). A new approach to tracking control of networked discrete-event systems. *IFAC-PapersOnLine*, 51(7), 448 – 455. 14th IFAC Workshop on Discrete Event Systems, Sorrento Coast.

Zgorzelski, M. and Lunze, J. (2019). Cooperative tracking control in networked discrete-event systems. In *6th International Conference on Control, Decision and Information Technologies 2019*, 115–120.