Abstract A distributed discrete-event system consists of an interconnection of two or more subsystems. Control of a distributed system demands a set of controllers each receiving the observation stream of a local subsystem and providing a subset of enabled events to that subsystem. Several control architectures are defined in this chapter: Distributed control, distributed control with communication, coordination control, and hierarchical control. Results are provided for several problems of decentralized and distributed control. Research issues for control of distributed discrete-event systems are described.

1.1 Introduction

The purpose of this chapter is to introduce to the reader the problem of supervisory control of distributed discrete-event systems. This problem area is highly relevant for current engineering and for companies developing high-tech systems. There are no fully satisfactory solutions yet. This chapter has therefore more the character of an introduction to the problem area and a research program, rather than the exposition of a fully investigated research topic. This chapter may also be regarded as an introduction to the Chapters ?? and ??.

Control engineering of high-tech systems nowadays is about systems consisting of very many components or subsystems. Examples of distributed discrete-event systems are large printers, a MRI scanner, a chemical plant, automobiles, aerial vehicles, etc. Each component was separately designed and has a controller developed for that component exclusively. But the large high-system has to meet a specifia-
tion which requires the interaction of all the subsystems. Control engineering and control theory therefore have to address the cooperation or the coordination of these subsystems.

In this chapter distributed discrete-event systems and the related concept of a decentralized discrete-event system are defined. Four control architectures are then defined: (1) distributed control, (2) distributed control with communication, (3) coordination control, and (4) hierarchical control. The control problems and the research approaches are then discussed. Results are presented for several subtopics of decentralized and distributed control. The chapter ends with research issues for control of distributed discrete-event systems and advice of further reading.

1.2 Motivation

*Example 1. The alternating bit protocol.* The protocol or variants of it are used in many communication networks. It is a canonical model of a distributed discrete-event system.

The engineering model of the communication system consists of a sender, a receiver, and a communication channel. The sender gets packets from the host at which it is located. The sender sends the package to the communication channel. The communication channel delivers a packet to the receiver either fully, or in a distorted way, or does not deliver the packet at all.

The alternating bit protocol operates as described next. The sender sends a packet and attaches a bit, either a zero or a one, to the header of the packet. The receiver receives information. If the receiver finds that the header of the packet is undamaged then it sends an acknowledgement to the sender including the protocol bit. The sender, if it does not receive an acknowledgement in a prespecified period, sends the packet again with the same bit. The sender, if it receives an acknowledgement of the receiver with the protocol bit, sends the next message with the next alternate protocol bit.

Note that at the sender there is information about the packets to be sent and that have been sent but not about the packets received by the receiver. Similarly, the receiver knows what it has received but not was sent by the sender. The distributed character of this system is in the different information and the different control actions at the two locations, at the sender and at the receiver. There is also communication from the receiver to the sender. The alternating bit protocol is a form of distributed control with communication. An extension of the above protocol exists in which, instead of only one bit, two or more bits are used so four, eighth, or more messages can be in process simultaneously, see [53].

*Example 2. Underwater vehicles.* At the university of Porto in Portugal there is a laboratory in which control of underwater vehicles is developed. This is a form of control of distributed systems. Part of the control deals with continuous space systems, but part of the control addresses the distributed discrete-event system.
Consider then a group of vehicles consisting of a surface vehicle, which acts as the local command center, and two or more underwater vehicles. The communication between the surface vehicle and the underwater vehicles is sonar communication, by sound waves. This form of communication requires relatively much energy from the underwater vehicle having limited battery power on board.

The operation of the vehicles is a characteristic example of a distributed system. Each vehicle is a subsystem of the group system. Each vehicle has local observations of its position and speed, and of its control actions. An underwater vehicle may communicate its position, speed, and action to the surface vehicle, either regularly, or when requested, or when required by a protocol.

A particular operation is formation flying. The surface vehicle follows a path specified by a command center. It then sends instructions to the underwater vehicles for a position and a speed to be reached at a particular time. The underwater vehicles then carry out the instructions and react only if they cannot meet the requirements.

Characteristic in this example is the local availability of the state information of the subsystem and the communication.

There follows a list of control engineering problems for which models in the form of decentralized/distributed discrete-event systems have been formulated and for which supervisory control has been investigated:

1. The alternating bit protocol described in Example 1. See [53], while discrete-event models are described in [38, 39].
2. Communication networks are a rich source of control of distributed and decentralized systems, see [8, 53].
3. Feature interaction in telephone networks, [47].
5. Control of high-speed printers with many sensors and actuators, [26].
6. A chemical pilot plant for which modular control is applied, [21, Ch. 7].
7. Distributed algorithms are studied by computer scientists, see [25, 31, 45, 50]. Supervisory control is not studied in most of those references.

1.3 Systems

In this section the reader is provided a classification of the system architectures for decentralized and for distributed discrete-event systems. In the literature there is no standardized nomenclature while one is needed. The concepts formulated below should be regarded as preliminary. The terms and notations used below have been introduced in the Chapters ??, ??, and ??.

**Definition 1. Overview of system architectures.** The acronym DES stands for a discrete-event system.

1. *Decentralized discrete-event system* is a global plant modeled as a DES with two or more observed event streams and two or more inputs of enabled events. Each
controller receives an observed event stream defined by either a projection or a mask, and inputs a subset of enabled events.

2. **Distributed discrete-event system** is a DES consisting of the interconnection of two or more subsystems. Each subsystem has an observation stream of local events and an input of enabled events to the local subsystem. **Modular discrete-event system** is the same as a distributed DES, term is still used in the literature. The authors favor the more general term of a distributed DES.

3. An **asynchronous timed distributed discrete-event system**. This is defined as a distributed discrete-event system in which each subsystem has its own clock and the clocks may drift with respect to each other. An example is an audio set consisting of a tuner, a CD player, and an amplifier, see [3]. Asynchronous systems require a timed DES model.

The main difference between a decentralized DES and a distributed DES is the use of the decomposition of the system into an interconnection of subsystems in a distributed DES while in a decentralized DES such a distinction is not made. For future research it is expected that the decomposition of the system into its subsystems can be usefully exploited for control synthesis.

**Definition 2.** A **decentralized discrete-event systems**, see Fig. 1.1, is a tuple denoted by \(G_{\text{dec}} = (Q, E, f, \mathcal{Q}_0, \mathcal{Q}_m, \{E_{i,c}, i \in I\}, \{E_{i,o}, i \in I\})\), where \(I = \{1, 2, \ldots, n\}\) denotes the index set of the observation streams, \(\{E_{i,c}, E_{i,uc} \subseteq E, i \in I\}\) is a partition of \(E\), \(E_{i,c}\) and \(E_{i,uc}\) denote respectively the subset of controllable events and of uncontrollable events of Controller \(i\). \(\{E_{i,o}, E_{i,uo} \subseteq E, i \in I\}\) is a partition of \(E\), and \(E_{i,o}\) and \(E_{i,uo}\) denote respectively the subset of observable events and the subset of unobservable events of Controller \(i\).

**Definition 3.** A **distributed discrete-event system**, see Fig. 1.2, is a tuple denoted by \(G_{\text{dis}} = (Q, E, f, \mathcal{Q}_0, \mathcal{Q}_m, \{E_i, i \in I\}, \{E_{i,c}, i \in I\}, \{E_{i,o}, i \in I\})\), where \(I = \{1, 2, \ldots, n\}\) denotes the index set of the subsystems, \(E_i \subseteq E\) denotes the subset of events of Subsystem \(i\), \(E_{i,c} \subseteq E_i\) denotes the subset of controllable events of Subsystem \(i\), \(E_{i,o} \subseteq E_i\) denotes the subset of observable events of Subsystem \(i\).
Definition 4. Control architectures

1. Decentralized/distributed control see Fig. 1.3. The observation stream of a controller consists of a projection or mask of the strings of the system which for a distributed system are restricted to the local subsystem. Each controller inputs a subset of enabled events to the subsystem. There is no direct communication whatsoever with other controllers though the controllers communicate indirectly with other controllers via the system.

2. Distributed control with communication, see Fig. 1.4. Controllers may send part of their observation stream or of their states to other controllers. Each controller uses besides its observation stream received directly from the plant also the other observations received from other controllers. An example is the class of nearest neighbor controls which use for the supervisory control the state of the local subsystem and the states of the nearest neighbors of the local subsystem.

3. Coordination control of a coordinated DES, see Fig. 1.5. A coordinated DES is defined as a distributed DES distinguished into a coordinator and the remaining subsystems. The coordinator has the control-theoretic task to coordinate the actions of the other subsystems. There is a controller for the coordinator and for each of the subsystems.

4. Hierarchical control There is a controller for each subsystem at each level of the hierarchy.

Guidelines for the appropriate choice of a control architecture are not much discussed in the literature and deserve more attention. The trade-off between more central control or more distributed control has to be made on a case by case basis. The principle is often stated that it is best to have the subsystems operate as independently as possible. But in many examples of distributed control problems a degree of coordination or centralized control is necessary to achieve the control objectives.
1.4 Problem Formulation

There follows a general problem statement which will be refined in the subsequent sections. The reader is assumed to be familiar with supervisory control of a discrete-event system both with complete observations and with partial observations as described the Chapters ?? and ??.
Problem 1. Supervisory control of a distributed DES. Consider a distributed DES with two or more observation streams. Consider a specification language, either a global specification or a set of local specifications. Determine a set of supervisory controls, as many as there are inputs of enabled subsets of event, such that the closed-loop system meets the control objectives of safety, required behavior, non-blockingness, and of fairness. There is a corresponding problem for decentralized discrete-event system.

Definition 5. Control objectives. There are of course the control objectives of supervisory control of discrete-event systems like safety, required behavior, and non-blockingness. In addition, there are control objectives particular for control of distributed DES:

- **Fairness.** Each subsystem is regularly provided access to all shared resources.
- **Non-starvation.** No subsystem is denied access forever to a shared resource.

The main difficulties of distributed control are: (1) control synthesis and design: the construction of a tuple of supervisory controls; more specifically, the fact that the partial observations of the subsystems differ and are not nested, and the fact that there are two or more controllers; and (2) the decidability and the complexity issues of the problem which are enormous.

In the next sections follow approaches to Problem 1.

1.5 Decentralized Control – Existence and Construction of a Tuple of Supervisors

In this section, the solution is presented to the supervisory control problem of a decentralized DES. It is remarkable that a necessary and sufficient condition for the existence of a tuple of supervisory controls can be presented as will be explained later. The main reference of this section is [40] and the results are extendable from two to three or more supervisors.

Definition 6. [40]. Consider a decentralized DES. For the supervisors \((S_1, g_1)\) and \((S_2, g_2)\) over the event sets \(E_1\) and \(E_2\), respectively, where \(P_1 : E_1^* \rightarrow E_1^*\) and \(P_2 : E_2^* \rightarrow E_2^*\) are natural projections, define a supervisor \((S_1 \land S_2, g_1 \ast g_2)\) over the global observable event set \(E_o = E_1^* \cup E_2^*\) with

\[(g_1 \ast g_2)(s) = [(g_1(P_1(s)))^c \cup (g_2(P_2(s)))^c]^c\]

for \(s \in E_o^*\).

Read this as: disable the event if either \(S_1\) or \(S_2\) disables the event. Hence it is called the conjunctive and permissive control-implementation architecture. The alternative is called the disjunctive and antipermissive control-implementation architecture. The supervisors working with partial observations are permissive while the global fusion rule is conjunctive. Recall from Chapter ?? that \(E_{1,cp}\) denotes the set of control patterns of Subsystem 1.
Proposition 1. [40]. Consider a distributed discrete-event system. The closed-loop system has the following properties: $L(S_1 \land S_2 / G) = L(S_1 / G) \cap L(S_2 / G)$ and $L_m(S_1 \land S_2 / G) = L_m(S_1 / G) \cap L_m(S_2 / G)$.

Definition 7. [40]. Global and local supervisors. Let $(S_1, g_1), g_1 : E_{1,o} \rightarrow E_{1,p}$ be a local supervisor. Its extension to the global event set $E$, called the global supervisor, is defined as $(\hat{S}_1, \hat{g}_1)$, where $\hat{g}_1 : E \rightarrow E_{cp}, \hat{g}_1(s) = g_1(s) \forall s \in E_{1,o}$, and it enables all events of $E_o \setminus E_{1,o}$. The corresponding extension is defined for $(\hat{S}_2, g_2)$.

Problem 2. [40]. Decentralized control for a global legal specification. Consider a decentralized DES $G$ with a prefix-closed regular specification language $\emptyset \neq K \subseteq L_m(G)$. Here $E_{ac} = E \setminus (E_{1,e} \cup E_{2,e})$. Construct supervisors ($S_1, g_1$) and ($S_2, g_2$) such that (1) $L_m(\hat{S}_1 \land \hat{S}_2 / G) = K$ and (2) $(\hat{S}_1 \land \hat{S}_2 / G)$ is nonblocking.

The solution to the above formulated problem requires introduction of concepts.

Definition 8. [40]. Consider Problem 2. Define the relations next action $\text{nextact}_K \subseteq E^* \times E \times E^*$, so that $(s_1, e, s_2) \in \text{nextact}_K$ if $s_1 e \in \text{prefix}(K)$, $s_2 \in \text{prefix}(K)$, and $s_2 e \in L(G)$ imply that $s_2 e \in \text{prefix}(K)$; and mark action $\text{markact}_K \subseteq E^* \times E^*$ so that $(s_1, s_2) \in \text{markact}_K$ if $s_1 \in K$ and $s_2 \in \text{prefix}(K) \cap L_m(G)$ imply that $s_2 \in K$.

Definition 9. Coobservability. [40]. Consider Problem 2. The sublanguage $K \subseteq L_m(G)$ is called coobservable with respect to $(G, P_1, P_2)$ if

$$\forall s, s_1, s_2 \in E^* \text{ with } P_1(s) = P_1(s_1), P_2(s) = P_2(s_2),$$
$$\Rightarrow \forall e \in E_{1,e} \cap E_{2,e}, (s, e, s_1) \in \text{nextact}_K \lor (s, e, s_2) \in \text{nextact}_K; \quad (1.1)$$
$$\lor \forall e \in E_{1,e} \setminus E_{2,e}, (s, e, s_1) \in \text{nextact}_K; \quad (1.2)$$
$$\lor \forall e \in E_{2,e} \setminus E_{1,e}, (s, e, s_2) \in \text{nextact}_K; \quad (1.3)$$
$$\lor (s, s_1) \in \text{markact}_K \lor (s, s_2) \in \text{markact}_K. \quad (1.4)$$

The concept of next action describes that the events are related if the corresponding observed strings are indistinguishable (Conjuncts 1.1-1.3). The marking relation describes that marking actions are related (Conjunct 1.4). Coobservability of a decentralized system corresponds to invariance of the closed-loop system with respect to control with distributed partial observations. There exists an algorithm to check whether a language is coobservable, see [37].

Example 3. Coobservability compared with observability. [40]. Consider the plant $G$ of which the diagram is displayed in Fig. 1.6(a) and its specification $K$ of which the diagram is displayed in Fig. 1.6(b). Denote the event sets by $E_{1,e} = E_{1,o} = \{a, b, c_1\}, E_{2,e} = E_{2,o} = \{a, b, c_2\}$. Then $K$ is coobservable; it is neither observable with respect to $(G, P_1)$ nor with respect to $(G, P_2)$. There exists a tuple of supervisory controls such that $L_m(\hat{S}_1 \land \hat{S}_2 / G) = K$, see [40, Ex. 4.2].

Definition 10. C & P Coobservability. Consider Problem 2 and the conjunctive and permissive control implementation architecture, see Def. ???. The sublanguage $K \subseteq L_m(G)$ is called C & P coobservable with respect to $(L(G), \{E_{o,i} | i \in I\})$ if
\[
\forall s \in \text{prefix}(K), \forall e \in E_c \text{ such that } se \in L(G),
\]
\[
\exists i \in I \text{ such that } e \in E_{i,c}, s' \in \text{prefix}(K), P_i(s) = P_i(s'), s'e \in \text{prefix}(K)
\]
\[
\Rightarrow se \in \text{prefix}(K).
\]

The concept of C & P coobservability stated above is based on a corresponding concept defined in [59] and [14]. C & P coobservability can be proven to be equivalent to coobservability. Several researchers prefer its interpretation above that of Def. 9.

**Theorem 1.** See [40, Th. 4.1]. Existence supervisors. There exists a solution to Problem 2 of decentralized control for a global regular legal specification language \( K \subseteq L_m(G) \) if and only if (1) the language \( K \) is controllable with respect to \((G, E_{uc})\) and (2) the language \( K \) is coobservable with respect to \((G, P_1, P_2)\).

An alternative proof is provided in [59]. There is a similar result in case of the D&A control implementation architecture, see [59].

Note that the language \( K \subseteq L_m(G) \) is fixed by the problem statement and is not to be determined in the problem. This is major difference with respect to decentralized control of nonlinear systems and is the main reason why this decentralized control problem admits an explicit solution.

**Algorithm 2** [40, p. 1701]. Construction of a decentralized tuple of supervisors. Data algorithm. Consider Problem 2 with a decentralized DES and with the regular language \( K \subseteq L_m(G) \) having a recognizer \( G_K = (Q_K, E_K, f_K, q_{K,0}, Q_{K,m}) \). Compute

\[
\begin{align*}
S_i, g_i & = (Q_i, E_{i,uc}, f_i, q_{i,0}, Q_{i,m}), i = 1, 2, \\
Q_i & = \text{Pwrset}(Q_K) \setminus \{\emptyset\}, \ Q_{i,m} = \{q_i \in Q_i | \exists q_i \cap Q_{K,m} \neq \emptyset\}, \\
q_{i,0} & = \{f_K(q_{K,0}, s) \in Q_K | s \in E^*, P_i(s) = \varepsilon\}, \text{ the unobservable (wrt. } P_i) \text{ reachset from the initial state,} \\
f_i(q_i, e) & = \begin{cases} 
\{f_K(q, e) \in Q_K | s \in E^*, P_i(s) = e, q \in q_i\}, & \text{if not empty,} \\
\text{undefined,} & \text{else.}
\end{cases} \\
g_i(q_i, e) & = E_{i,uc} \cup \{e \in E_{i,c} | \exists q \in q_i \text{ such that } f_i(q, e) \text{ is defined}\}.
\end{align*}
\]

Fig. 1.6 (a) A two-ring discrete-event system and (b) its specification.
Then \( \{g_1, g_2\} \) is a solution of Problem 2.

**Proposition 2.** [40, Prop. 4.1]. Consider Problem 2.

(a) If the sublanguage \( \emptyset \neq K \subseteq L_m(G) \) is controllable, coobservable, and prefix-closed then the supervisors constructed in the algorithm above satisfy (1) \( L_m(\tilde{S}_1 \land \tilde{S}_2 / G) = K \) and (2) \( (\tilde{S}_1 \land \tilde{S}_2 / G) \) is nonblocking.

(b) The time complexity of Algorithm 2 is exponential in the size of the state set of \( G_K \).

The new condition for the existence of a tuple of decentralized supervisors is the concept of coobservability. A sufficient condition of coobservability is decomposability whose computation has a lower time complexity.

**Definition 11.** [40, p. 1696, p. 1702]. Consider Problem 2

(a) The language \( K \subseteq L_m(G) \) is called decomposable with respect to \( (G, P_1, P_2) \) if \( K = P_1^{-1}(P_1(K)) \cap P_2^{-1}(P_2(K)) \cap L(G) \).

(b) The language \( K \) is called strongly decomposable with respect to \( (G, P_1, P_2) \) if \( K = L(G) \cap [P_1^{-1}(P_1(K)) \cup P_2^{-1}(P_2(K))] \).

**Proposition 3.** [40, Prop. 4.2] Let \( K \subseteq L_m(G) \) be \( L_m(G) \)-closed. If prefix(\( K \)) is strongly decomposable then \( K \) is coobservable.

There exists an example showing that \( K \) coobservable does not imply that prefix(\( K \)) is strongly decomposable, see [40, Ex. 4.1].

**Proposition 4.** [40, Prop. 4.3]. Assume that \( K \subseteq L_m(G) \) is regular, \( L_m(G) \)-closed, and controllable, \( E_{i,c} \subseteq E_{i,o} \), for \( i = 1, 2, E_{1,o} \cap E_{2,c} \subseteq E_{1,c} \) and \( E_{2,o} \cap E_{1,c} \subseteq E_{2,c} \). Then \( K \) is coobservable if and only if \( K \) is decomposable.

The following example shows that there exists discrete-event systems for which decentralized control can never meet the control objective. The control objective can only be met if the two controllers communicate. The focus of research should therefore be extended or redirected to decentralized/distributed control with communication as described in Chapter ??.

**Example 4. Decentralized control requiring communication.** The distributed DES is specified in Fig. 1.7. Note that \( E_{1,c} = E_{2,c} = \{e, f\} \), \( E_{1,o} = \{c_1, e, f\} \), \( E_{2,o} = \{d_2, e, f\} \), \( q_5 \) is the forbidden state, \( q_9 \) is an accepting/marked state. From the diagram it is then clear that neither of the supervisors knows whether in state 4 and in state 8 to enable \( e \), or \( f \), or both because neither of them observes in which order the events \( c_1 \) and \( d_2 \) occur. They must communicate with each other to determine this order.
1.6 Decentralized Control – Undecidability

Next follows a major result on undecidability of a decentralized control problem. In [40] it is proven that the decentralized control problems with the following relations are decidable: (1) \( L_m(\tilde{S}_1 \land \tilde{S}_2 / G) = K \), [40, Th. 4.1]; and (2) \( L(\tilde{S}_1 \land \tilde{S}_2 / G) \subseteq K \), [40, Th. 4.2]. Does decidability still hold true in a further generalization of the problem?

**Problem 3.** [48, Def. 5.1]. Consider the setting of Problem 2 with a regular specification language \( K \subseteq L_m(G) \). Does there exist a tuple of supervisors \((S_1, g_1), (S_2, g_2)\) such that \( L_m(\tilde{S}_1 \land \tilde{S}_2 / G) \subseteq K \) and such that the closed-loop system is nonblocking?

Note the difference between Problem 1 and Problem 3 in the relation of the marked language of the closed-loop system with the language of the specification.

**Theorem 3.** [48, Th. 5.1]. Problem 3 is undecidable.

The proof of the above theorem proceeds by reduction of the problem to an observability problem of [30]. A result corresponding to the above theorem for an \( \omega \)-language setting was proven in [20]. See further [36, 46].

1.7 Decentralized Control – Maximal Languages

In supervisory control of a discrete-event system if the specification is not controllable one considers the supremal controllable sublanguage. In the setting of control of a distributed DES this cannot be done directly because there are always two or more supervisors. Therefore other concepts are needed, a maximal solution and a Nash equilibrium as described below. The theory below was published before Theorem 3 listed above. In this section a slightly different definition of a supervisor is used than in Section 1.5 borrowed from [27]. The reason to do so are only notational, there is no inherent restriction.

**Definition 12.** Consider a discrete-event system denoted by \( G = (Q, E, f, q_0, Q_m) \). Denote a supervisory control based on partial observations of this system by \( g : P(L(G)) \rightarrow \text{Pwrset}(E_c) \), where \( P : E^* \rightarrow E^*_p \) is a natural projection. Unlike the previous sections, \( g(P(s)) \) denotes the subset of disabled controllable events,
Definition 13. Consider two supervisory controls \( g_a, g_b : \mathbb{P}(L(G)) \rightarrow \mathbb{P} \mathbb{W}(E_c) \) implemented in relation 12 Jan Komenda, Tomáš Masopust, and Jan H. van Schuppen event system and two local supervisory controls \( g_a, g_b \). Define the implementation relation, denoted by \( g_a \subseteq g_b \), and say that \( g_a \) implements \( g_b \), if \( g_b(s) \subseteq g_a(s) \), \( \forall s \in \mathbb{P}(L(g_a/G)) \). In words, \( g_a \) disables more than \( g_b \).

Definition 14. Consider a discrete-event system \( G \) and two local supervisory controls, \( g_1 : P_1(L(G)) \rightarrow \mathbb{P} \mathbb{W}(E_{1,c}) \) and \( g_2 : P_2(L(G)) \rightarrow \mathbb{P} \mathbb{W}(E_{2,c}) \), and let \( P : E^* \rightarrow (E_{1,c} \cup E_{2,c})^* = E_c^* \) be a natural projection. Define the composition of these two supervisory controls as the supervisory control, \( g_1 \wedge g_2 : \mathbb{P}(L(G)) \rightarrow \mathbb{P} \mathbb{W}(E_{1,c} \cup E_{2,c}) \), \( (g_1 \wedge g_2)(s) = g_1(P_1(s)) \cup g_2(P_2(s)) \), \( \forall s \in \mathbb{P}(L(G)) \).

Proposition 5. [27, Prop. 2.10]. Consider the setting of Def. 14 with a discrete-event system and two local supervisory controls \( g_1 \) and \( g_2 \). Then \( L(g_1 \wedge g_2/G) = L(g_1/G) \cap L(g_2/G) \).

Problem 4. The decentralized supervision safety problem. Consider a discrete-event system \( G \), a specification language \( K \subseteq L_m(G) \), \( E = E_1 \cup E_2 \), and the centralized optimal supervisory control \( g^i \) such that \( L(g^i/G) = \sup \mathbb{C}_{pc}(K, G) \). Determine a tuple of local supervisory controls, \((g_1, g_2)\), such that \( g_1 \wedge g_2 \subseteq g^i \).

Definition 15. Consider Problem 4. The tuple of local supervisory controls \((g_1^1, g_2^1)\) is called an optimal decentralized solution if (1) \( g_1^1 \wedge g_2^1 \subseteq g^i \); (2) \( \forall (g_1, g_2) : g_1 \wedge g_2 \subseteq g^i \) implies that \( L(g_1 \wedge g_2/G) \subseteq L(g_1^1 \wedge g_2^1/G) \). Thus, the closed-loop language of the optimal decentralized solution is least restrictive.

An optimal decentralized solution may not exist. Therefore attention is restricted to a maximal solution defined next.

Definition 16. Consider Problem 4. The tuple of local supervisory controls \((g_1^\square, g_2^\square)\) is called a maximal solution if (1) \( g_1^\square \wedge g_2^\square \subseteq g^i \); (2) \( \nexists (g_1, g_2) \) such that \( g_1 \wedge g_2 \subseteq g^i \) and \( L(g_1^\square \wedge g_2^\square/G) \subseteq L(g_1 \wedge g_2/G) \). Equivalently, there does not exist another tuple of supervisory controls with a strictly larger closed-loop language.

How to determine a maximal solution? There do not exist general results on how to determine all maximal solutions. A way to proceed is to use the concept of a Nash equilibrium.

Definition 17. The tuple of local supervisory controls \((g_1^\circ, g_2^\circ)\) is called a Nash equilibrium if

1. \( (g_1^\circ \wedge g_2^\circ) \subseteq g^i \);
2. \( g_1 \wedge g_2 \subseteq g^i \Rightarrow L(g_1 \wedge g_2/G) \subseteq L(g_1^\circ \wedge g_2^\circ/G) \), \( \forall g_2 \);
3. \( g_1 \wedge g_2 \subseteq g^i \Rightarrow L(g_1 \wedge g_2/G) \subseteq L(g_1^\circ \wedge g_2^\circ/G) \), \( \forall g_1 \).

The concept of a Nash equilibrium is named after the mathematician/economist John Nash who introduced the concept into game theory. Decentralized control is
a special case of a dynamic game problem, all players have the same cost function though different observations. There exists an example such that a Nash equilibrium is not a maximal solution. Therefore the concept of Nash equilibrium has to be strengthened. After that the equivalence condition of a maximal solution can be stated.

**Definition 18.** The tuple of local supervisory controls \((g_1^o, g_2^o)\) is called a **strong Nash equilibrium** if (1) it is a Nash equilibrium and (2) \(\forall (g_1, g_2) : L(g_1 \land g_2 / G) = L(g_1^o \land g_2^o / G)\) implies that \((g_1, g_2)\) is a Nash equilibrium.

**Theorem 4.** [27, Th. 3.4]. Consider Problem 4. The tuple of local supervisory controls \((g_1^o, g_2^o)\) is a maximal element if and only if it is a strong Nash equilibrium.

**Procedure 5** Construction of a maximal tuple of supervisory controls.

1. **Determine a tuple of supervisory controls which is a strong Nash equilibrium,** see below.
2. **Then conclude with Theorem 4 that the considered tuple of supervisory controls is a maximal element.**

**Example 5. Joint action.** Consider the discrete-event system of which the diagram is displayed in Fig. 1.8. Note that \(E_1 = \{a_1, b_1\}\), \(E_2 = \{a_2, b_2\}\), \(E_1 \cap E_2 = \emptyset\).

![Fig. 1.8 The diagram of a distributed DES for which there exists a strong Nash equilibrium.](image)

\(E_c = \{a_1, a_2\}\). The centralized supervisory control which produces the specification sublanguage is described by the language \(K = \{(a_2 b_1)^*\} \parallel \{(a_1 b_2)^*\}\). Consider the following tuple of supervisory controls \((g_1, g_2)\). Supervisory control \(g_1\) always disables controllable event \(a_1\) and the uncontrollable event \(b_1\) is always enabled. Supervisory control \(g_2\) initially enables \(a_2\), and disables \(a_2\) only after the first occurrence of \(b_2\) and then \(a_2\) remains always disabled. Uncontrollable event \(b_2\) is always enabled.

This tuple of supervisory controls in closed-loop with the plant achieves the specification, is a strong Nash equilibrium, and hence a maximal solution. See [27, Ex. 4.5.].

There exists a procedure for computing Nash equilibria, see [27], but there is an example of a controlled DES for which this procedure does not converge in a finite number of steps while there exists a strong Nash equilibrium, see [27, Ex. 4.6].
1.8 Distributed Control of Distributed DESs

In this section distributed control is investigated. The older term used for this approach is modular control and both terms will be used in this section. By going from a decentralized system, as discussed in the three preceding sections, to a distributed system, the decomposition of the global system into two or in general many subsystems is to be noted.

The motivation for the investigation of distributed/modular discrete-event systems is the complexity of control design. The time complexity of the computation of the supervisor increases polynomially with the size of the state set and the event set. If the computation of the supervisor can be carried out for each subsystem in parallel then the overall time complexity will be much less. This is the main motivation for the investigation of the control synthesis of distributed/modular systems.

Modular control synthesis of DES was first investigated in [57]. But approaches of distributed/modular control have been investigated in control theory and other areas of engineering and the sciences for many centuries.

Specifications of control of distributed systems can be distinguished into a global specification and a set of local specifications, where the latter contain one specification per subsystem.

**Problem 5.** Distributed control of a distributed/modular discrete-event system. Consider a distributed/modular discrete-event system and a specification. Restrict attention to the distributed control architecture defined in Def. 4. Determine a set of supervisors, one for each subsystem, such that the closed-loop system meets the specification.

Define the distributed/modular control synthesis as to synthesize a supervisor for each of the local subsystems. The supervisor of the distributed system is then the set of the local supervisors.

Global control synthesis is defined by the steps: (1) compose all subsystems into one system and then (2) compute the supervisory control for the composed system according to the method described in Chapter ???. The global control synthesis is used only for a theoretical comparison.

Problem 5 leads to the following research issues:

1. Is the closed-loop system nonblocking? There exist examples in which the system is blocking. The quest is therefore to find equivalent or sufficient conditions for nonblockingness of the closed-loop distributed system.
2. Can a distributed control synthesis as described below achieve the same closed-loop language as a global control synthesis?

The research issue of nonblockingness of a composition of two or more subsystems has been discussed in Chapter ???. See also [57]. If the languages of two subsystem satisfy the condition of nonconflicting languages, see Chapter ??, then the product of their languages is nonblocking. But the time complexity of checking nonconflictingness is almost as high as the time complexity of checking nonblockingness of the global systems. Alternative approaches are to use the observer property, see
[28] or abstractions with particular properties [7]. Coordination control, see Chapter ??, is another approach to deal with nonblockingness of the interconnection of a distributed system.

The second research issue is whether the closed-loop systems obtained by global control synthesis and by modular control synthesis are equal. The answer to the question depends on the interaction of the subsystems. It will be stated below that the equality of global control synthesis and of modular control synthesis is equivalent to the concept of modular controllability.

Recall the definition and the notation of a distributed discrete-event system, in this section also referred to as a modular DES,

$$G_{dis} = (Q, E, f, q_0, Q_m, \{E_i, i \in I\}, \{E_i, i \in I\}, \{E_i, i \in I\})$$

where the index set of the subsystems is denoted by $I = \{1, 2, \ldots, n\}$, the subset of the events of Subsystem $i$, $E_i \subseteq E$, the subset of controllable events of Subsystem $i$, $E_{i,c} \subseteq E_i$, and the subset of observable events of Subsystem $i$, $E_{i,o} \subseteq E_i$. Denote further $G_i = (Q_i, E_i, f_i, q_{i,0}, Q_{i,m})$, $L_i = L(G_i)$, $L = \bigcap_{i=1}^n L(G_i)$, $E_c = \bigcup_{i=1}^n E_{i,c}$, and $E_{uc} = \bigcup_{i=1}^n E_{i,uc}$. Then, $P_i : E^* \rightarrow E_i^*$ is a natural projection and $P_i^{-1} : \text{Pwrset}(E_i^*) \rightarrow \text{Pwrset}(E^*)$.

**Definition 19.** Consider a modular DES. The subsystems are said to **agree on the controllability of their common events** if $E_{i,c} \cap E_j = E_{i,c} \cap E_j$ for all $i, j \in I$ with $i \neq j$. The subsystems are said to **agree on the observability of their common events** if $E_{i,o} \cap E_j = E_{j,o} \cap E_i$ for all $i, j \in I$ with $i \neq j$.

The definition that the subsystems agree on the controllability status of their events implies that $E_{i,c} = E_c \cap E_i$ and $E_{uc} = \bigcup_{i \in I \setminus \{i\}} (E_i \setminus E_{i,c})$.

**Definition 20.** **Local specification languages.** Define the local specification languages corresponding to a global specification language $K \subseteq E^*$ as $\{K_i, i \in I\}$, $K_i = K \cap P_i^{-1}(L(G_i)) \subseteq E_i^*$, for all $i \in I$. Then, $K \cap L = \bigcap_{i=1}^n K_i$. One says then that $K_i$ locally overapproximates $K$.

**Definition 21.** **Modular control synthesis and global control synthesis.** Define global control synthesis by the associated closed-loop language of the plant and the supervisory control, $\text{supC}(K \cap L, L_{uc})$. Define modular control synthesis by the associated closed-loop language of the plant and the associated supervisory controls, $\bigcap_{i=1}^n \text{supC}(K_i, L_i, L_{uc})$.

**Problem 6.** Does a modular control synthesis equal a global control synthesis? Which conditions imply that modular control synthesis equals global control synthesis? Equivalently, when does equality in the following expression, $\bigcap_{i=1}^n \text{supC}(K_i, L_i, L_{uc}) = \text{supC}(K \cap L, L_{uc})$ hold true?

**Definition 22.** The modular system is called **modularly controllable** if $L_{E_{uc}} \cap P_i^{-1}(L_i) \subseteq L, \forall i \in I$.

**Theorem 6.** [16, Th. 6.7]. Modular equals global control synthesis in case of locally complete observations. Assume that the subsystems agree on the controllability of their common events.
(a) If the modular system is modularly controllable then \( \bigcap_{i=1}^{n} \sup C(K_i, L_i, E_{uc}) = \sup C(K \cap L, E_{uc}) \).

(b) If the above equality holds for all \( K \subseteq E^* \) then the modular system is modularly controllable.

In the literature there are other sufficient conditions that are much less complex to check, called mutual controllability and global mutual controllability, [16].

**Proposition 6.** [16, Prop. 8.3, Prop. 8.4]. The time complexity of the computation of the supremal controllable sublanguage by modular control synthesis is \( O(n^2(n^*)^2m^2) \) where \( n^* = \max_{i=1}^{n} n_i, m = \max_{i=1}^{n} m_i, n = |Q|, m = |E| \). The time complexity of the computation of the supremal controllable sublanguage by global control synthesis is \( O((n^*)^{2m}n_k) \). The gain in time complexity of modular control synthesis with respect to global control synthesis is then clear.

There exist concepts and theorems for modular control with only local partial observations. The main references for this section are [15, 16, 12, 13]. Other references are [54].

### 1.9 Research Issues

Engineering has a need for further research on control of distributed systems because the existing theory is far from satisfactory.

- **Decentralized control.** The results presented earlier in this chapter require further analysis of control theoretic interpretations of the operation of the supervisors which may then be useful for the development of decentralized control also for other classes of systems. Based on control theory of stochastic systems one expects to note how the structure of the control law deals with common and private information of the subsystems.

- **Distributed control of a distributed DES.** An open problem is to find a condition for nonblockingness of the closed-loop system which is of low time complexity compared with global control synthesis. What is decidable, computationally attractive, and achievable by distributed control of distributed DES?

- **Distributed control with communication.** For particular engineering control problems this form of control is attractive and this motivates investigations of the following research topics: (1) Experience with the forms of this type of control. Examples of control of communication networks will be useful. (2) Synthesis and design of communication laws and control laws. What, when, and to whom to communicate? (3) A theoretical framework for synthesis of distributed control laws with communication has to be developed using concepts of common and of private information, observers, and control. (4) The trade-offs between control and communication on the overall performance have to be investigated.

- **Coordination control.** Research issues needing attention include: Decomposition of large systems into coordinated systems may be useful depending on the ap-
The trade-offs between control and communication also play a role in this setting. The reader is referred to Chapter ?? for further information.

- **Hierarchical control.** The class of hierarchical systems to be considered will be quite general. Research issues include decomposition, abstraction, system reduction, the algebra of hierarchical-distributed systems, the relations between controllers of adjacent hierarchical levels, and the communication and computation aspect of the different hierarchical layers.

### 1.10 Further Reading

The reader is advised to read after the reading of this chapter the following chapters on other aspects of control of distributed systems: Chapter ?? *Distributed Control with Communication*, and Chapter ?? *Coordination Control of Discrete-Event Systems*.

**Books and lecture notes.** The lecture notes of W.M. Wonham, see the Chapters 4, 5, and 6 of [56]. These are available on the web, http://www.control.utoronto.ca/~wonham/

See for a book on supervisory control of decentralized DES [4, Chapter 3].

**Decentralized control.** The problem of decentralized control of discrete-event systems was formulated in [5]. The concept of coobservability and the theorem that it is equivalent to existence of a decentralized supervisor is due to [40]. The algorithm to check coobservability is described in [37]. The space complexity of the algorithm and related matters were investigated in [33, 35, 34]. Early papers on decentralized control include [23, 22, 24, 17]. Other papers on decentralized control include [11, 19, 29, 43, 55, 58].

**Decentralized control for non DES.** To assist the reader, there follow several references on decentralized control of systems which are not discrete-event systems.

Books on decentralized control include [6, 51, 52]. Survey papers are [10, 41]. Papers on complexity of decentralized control problems include [49, 2].

**Hierarchical control of hierarchical discrete-event systems.** The concept of a hierarchical system is rather old. In artificial intelligence the study of these systems was stimulated by Herbert Simon. An early framework for hierarchical DES was proposed in [9] and is called state charts. Other references on hierarchical control synthesis of DES include [60, 44]. The approach of hierarchical with distributed DES is described in [42, 18].

**Asynchronous timed DES.** There are engineering distributed systems were time is best modeled asynchronously. An example is a set of audio equipment consisting of a tuner, a CD player, and an amplifier, see the paper [3]. Each subsystem may have its own clock and the clocks may drift with respect to each other. An approach to diagnosis of distributed asynchronous systems is described in [1]; see also [32].
Acknowledgments

The authors thank the following researchers for their contributions to the topic of this chapter: A. Overkamp, K. Rudie, S. Tripakis, P. Varaiya, K.C. Wong, and W.M. Wonham. The authors also thank the anonymous reviewers.

The authors are very grateful for the financial support in part by the European Community’s Seventh Framework Programme (FP7/2007-2013) via grant INFSO-ICT-224498 (DISC Project), and to the Academy of Sciences of the Czech Republic for the Institutional Research Plan AV0Z10190503 and for the GAČR grants P103/11/0517 and P202/11/P028.

References


## Index

<table>
<thead>
<tr>
<th>Control architectures</th>
<th>agree on controllability</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>control architectures</td>
<td>agree on observability</td>
<td>15</td>
</tr>
<tr>
<td>coordination control</td>
<td></td>
<td></td>
</tr>
<tr>
<td>distributed control</td>
<td></td>
<td></td>
</tr>
<tr>
<td>distributed control</td>
<td></td>
<td></td>
</tr>
<tr>
<td>with communication</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hierarchical control</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control objectives</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fairness</td>
<td></td>
<td></td>
</tr>
<tr>
<td>non starvation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Control synthesis</td>
<td></td>
<td></td>
</tr>
<tr>
<td>global</td>
<td></td>
<td></td>
</tr>
<tr>
<td>modular</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coobservability</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C &amp; P</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>decentralized</td>
<td></td>
<td></td>
</tr>
<tr>
<td>discrete-event system</td>
<td></td>
<td></td>
</tr>
<tr>
<td>distributed</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Event sets</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example</td>
<td></td>
<td></td>
</tr>
<tr>
<td>alternating bit protocol</td>
<td></td>
<td></td>
</tr>
<tr>
<td>underwater vehicles</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Language</td>
<td></td>
<td></td>
</tr>
<tr>
<td>decomposable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>strongly decomposable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modular DES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>modularly controllable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tuple of supervisors</td>
<td></td>
<td></td>
</tr>
<tr>
<td>maximal solution</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nash equilibrium</td>
<td></td>
<td></td>
</tr>
<tr>
<td>strong Nash equilibrium</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>