Abstract

This thesis aims to develop methods for fault-tolerant control of networked systems. Fault tolerance is achieved by redistributing the control tasks of the faulty subsystems to the healthy subsystems. This proposed solution strategy exploits the fact that often networked systems have to accomplish a cooperative task together, where the behavior of a specific subsystem is not relevant. Task redistribution in the presence of a fault does not restore the functionality of faulty subsystems, as it is usually the case, but rather ensures the accomplishment of the cooperative task. This distinguishes the approach of the present thesis from the solutions often described in the literature.

The following example illustrates the approach: In the power supply, many power plants jointly cover the power demand (=cooperative task), where each power plant has to supply a certain amount of energy (=control tasks). If one power plant can no longer deliver the required energy due to a fault, other power plants can increase their output to meet the demand (=task redistribution).

Task distribution in the fault-free case and redistribution in the fault case always require a decomposition of the cooperative task, considering the capabilities of the subsystems. For this purpose, a global coordinator is introduced that performs such a decomposition and sends the computed control tasks to the subsystems. The subsystems are then able to adapt their tasks independently without compromising the integrity of the decomposition. In this way, the autonomy of the subsystems is taken into account.

To trigger the task redistribution in the presence of a fault, the subsystems must detect the fault occurrence, isolate the faulty subsystems and identify the fault effect. Therefore, the subsystems are equipped with diagnostic units that generate local residual signals. Faults are diagnosed locally by local evaluation of these signals. To distinguish faults from influences of neighboring subsystems, the diagnostic units exchange information.

The control task that each subsystem must accomplish requires the subsystem output to be moved along a time-varying trajectory. For this purpose, each subsystem is equipped with a two-degrees-of-freedom controller. To ensure the strict requirement of trajectory tracking, the controllers exchange data with each other, allowing the mutual influences of the subsystems to be taken into account.

As part of this thesis, a new testbed – called COCO – has also been designed, built, and put into operation. COCO is a transport system consisting of 50 linear actuators, which can be used to confirm the practical applicability of the developed methods.

The Problem of Fault-Tolerant Task Assignment 1

This chapter introduces the reader to networked systems and how to prevent failures in such systems. The underlying research questions are presented and the proposed solution of fault-tolerant task assignment is summarized. A classification of the existing literature is used to compare the contributions of this thesis with what is already known.

1.1 Introduction to Fault-Tolerant Control of Networked Systems

1.1.1 General Idea

This thesis deals with networked systems that together have to exhibit a desired common behavior, in other words, they have to satisfy a cooperative task - even when a subsystem is affected by a fault. This requires a coordination of the subsystems, which is achieved by a suitable decomposition of the cooperative task into local control tasks. As the focus is given to the subsystems' common behavior, the decomposition can be changed as long as the cooperative task keeps being accomplished. This introduces redundancy to the overall system, which is important for fault-tolerant control because the cooperative task can be satisfied in multiple ways. If faults are causing subsystems to malfunction, their control tasks can be redistributed among the remaining healthy systems. Hence, faults can be compensated by means of cooperation.

Figure 1.1 shows the networked systems and their interconnection. The subsystems are autonomous units having local controllers and making local decisions. The subsystems can be connected with each other either by a physical or a digital communication network. While the physical network is fixed, the communication network allows information exchange among the subsystems.

What are the main problems to be solved?

For a successful decomposition of the cooperative task, the subsystems' capabilities have to be considered (**decomposition problem**). While there are some limitations, such as output constraints that are known in advance, faults occurring during runtime are a-priori unknown. Therefore, the subsystems have to be able to detect a fault occurrence and to

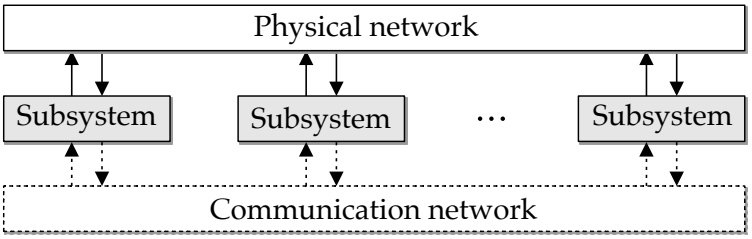

Fig. 1.1: Structure of networked systems. Dashed lines indicate that a signal is transmitted via a digital communication network.

isolate the faulty subsystems (**diagnostic problem**). Since the physical network allows a fault to spread out, it is difficult for a subsystem to distinguish between a local fault and the physical influence of other subsystems. Independently of whether subsystems are faulty or not, each subsystem must be able to satisfy its control tasks. The control tasks require the subsystems to steer their outputs along given reference trajectories, which is difficult to achieve due to the mutual influence caused by the physical network (**trajectory tracking problem**).

What are the essential outcomes of this thesis?

The decomposition problem is solved by the subsystems together with a global coordinator referred to as the task assignment unit (TAU). The unit TAU specifies a valid decomposition of the cooperative task, although the subsystems do not necessarily have to follow this specification at all times. It is shown when the subsystems are allowed to ignore the specifications of the unit TAU and define their own control tasks instead. In order to trigger a redistribution of the control tasks in a fault scenario, the subsystems are equipped with diagnostic units that share information with each other. As a result, a subsystem can immediately identify its own fault behavior. The communication network is also used by the subsystem controllers to steer the subsystem outputs along reference trajectories. Among others, the references need to satisfy certain differentiability conditions that are considered during the decomposition process. This shows that not only the diagnostic problem but also the trajectory tracking problem is closely related to the decomposition problem. To evaluate the individual components and their interconnection in experiments, there has been a completely new testbed – called COCO – built. COCO is used to demonstrate the effectiveness of the methods proposed in this thesis.

The aim of this thesis is to combine three topics of control theory: Networked systems, fault-tolerant systems, and cooperating systems. This combination is referred to as *fault-tolerant task assignment in networked systems*. This chapter is intended to outline the significance and relevance of this combination in which the most important question is:

> How can a flexible assignment of control tasks help to prevent that malfunctions of subsystems adversely affect the accomplishment of a cooperative task?

1.1.2 Illustrative Example

Figure 1.2 shows a transportation system that is used as an illustrative example of fault-tolerant task assignment. Linear actuators are mounted under a stretchable latex foil and have the cooperative task to maneuver the ball along some circular path. Under nominal conditions all actuators can extend and retract their cylinders to enforce some shape on the foil that accelerates the ball along the desired path. When an actuator becomes faulty, which is indicated by the lightning symbol in Fig. 1.2, it cannot accelerate the ball as intended. However, the aim of this thesis is to make the transportation system fault-tolerant with respect to the cooperative task of moving the ball by a modification of the control tasks of neighboring actuators which thereby compensate for the missing contribution of the faulty actuator.

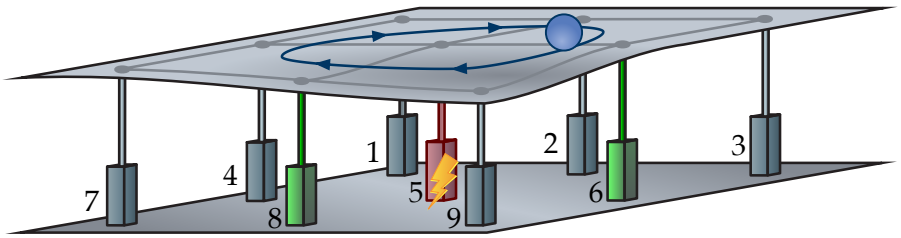

Fig. 1.2: Linear actuators have the cooperative task to steer the ball along some circular path although one actuator is faulty and cannot be moved.

The desired path of the ball also contains the specification of when the ball should be at a certain position. It is therefore a matter of guiding the ball along a trajectory (location and time), which requires the movement of all actuators to be coordinated very precisely with each other. Let denote the position of the ball by $\boldsymbol{s}(t)$ and the cylinder displacement by $\boldsymbol{y}_i(t)$, $(i = 1, 2, \ldots, 9)$, then coordination requires to generate reference trajectories for the displacements $\boldsymbol{y}_i^*(t)$ based on the desired ball position $\boldsymbol{s}^*(t)$. Supposing that this coordination is done properly and each actuator behaves as intended, the ball will be steered along its reference:

$$\boldsymbol{y}_1(t) = \boldsymbol{y}_1^*(t), \quad \boldsymbol{y}_2(t) = \boldsymbol{y}_2^*(t), \quad \cdots \quad , \boldsymbol{y}_9(t) = \boldsymbol{y}_9^*(t) \quad \Rightarrow \quad \boldsymbol{s}(t) = \boldsymbol{s}^*(t)$$

The coordination of the actuators has to consider several aspects. Firstly, the faulty actuator is completely blocked (stuck-at fault) and cannot be moved. Its previous contribution to the ball acceleration must be redistributed to others. Secondly, since the actuators cannot extend their cylinders infinitely quickly, the reference trajectories have to satisfy certain properties regarding differentiability and the rate of change. Thirdly, each actuator is supposed to be an autonomous system with its own control unit that shares information with other actuators only when necessary. And fourthly, the ball is not influenced by

all actuators at the same time but only by those in its direct vicinity, which introduces a switching character to the overall system.

1.1.3 Cooperative Tasks versus Subtasks

Every technical system has a certain task to achieve, in others words, it serves a certain purpose. Putting the task of a system in the focus is particularly interesting in connection with networked systems because they usually have to fulfill a cooperative task. Power plants supply a city with energy, trucks form a convoy to save fuel, tugboats maneuver large ships together in port, and drones fly in a formation. The priority is not on the individual behavior, but on the joint one. The cooperative task is fulfilled when all subsystems accomplish their subtasks. The term *subtask* refers to the control task of a subsystem and should emphasize the connection to the cooperative task. Therefore, the cooperative task needs to be decomposed into appropriate subtasks (see Fig. 1.3), taking into account the capabilities of the subsystems. The decomposition process is referred to as task assignment that can be summarized as follows:

> The task assignment problem requires to assign subtasks to a set of subsystems that having satisfied these tasks, fulfill a given cooperative task.

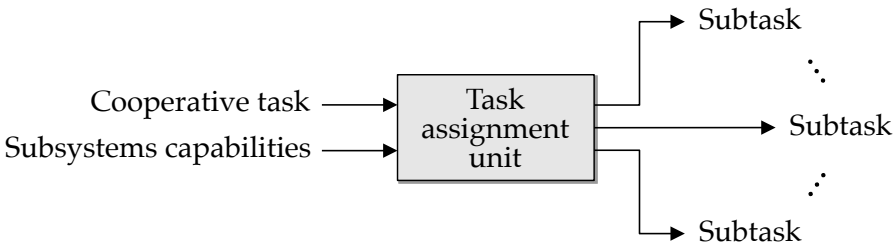

Fig. 1.3: Illustration of the task assignment problem in which the cooperative task is decomposed into suitable subtasks.

For a fault-tolerant task assignment in networked systems it is necessary to understand what characterizes fault-tolerant systems and what are networked systems. Therefore, Subsection 1.1.4 describes the structure of fault-tolerant systems while Subsection 1.1.5 introduces networked systems. Once both systems are understood separately, they are combined to state the aim of this thesis (Section 1.2).

1.1.4 Fault-Tolerant Systems

During their lifetime, technical systems are continuously subject to stresses that can have a negative impact on their components. If the functionality of a component deviates

too much from its intended performance, serious damage can occur: People can be injured, the environment can be damaged, other system components can be destroyed or financial losses can occur due to production downtime. To avoid or at least mitigate these consequences the system has to be made fault-tolerant.

The structure of a fault-tolerant system is shown in Fig. 1.4. There are two layers that can be identified. The execution layer combines the plant P and the controller C. The supervision layer consists of the diagnostic unit D, which detects faults affecting the plant, and the reconfiguration unit R, which adjusts the controller to the fault scenario.

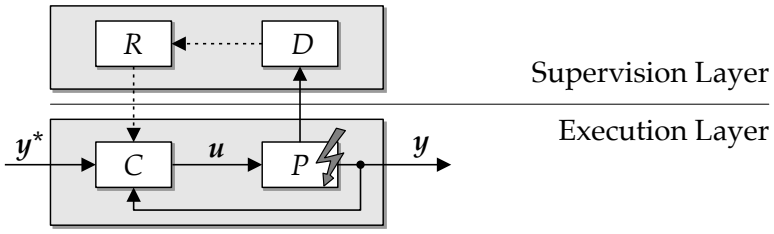

Fig. 1.4: Structure of a fault-tolerant system: The diagnostic unit D detects a fault in the plant P and triggers the reconfiguration unit R to adapt the nominal controller C to the fault scenario.

It is important to distinguish between a fault and a failure. According to [83], a *fault* is "an unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable / usual / standard condition". The loss of a sensor, for example, characterized by a sensor showing the same value although the measured quantity has changed is a fault. The performance degradation of an actuator where the actuator's output is not as high as expected is another. In contrast, a *failure* is "a permanent interruption of a system's ability to perform a required function under specified operating conditions". In other words a failure is what is caused by a fault: its negative consequence.

The aim of any fault-tolerant system is to prevent a fault from causing a failure.

Rather than having a single system P as shown in Fig. 1.4, this thesis investigates the situation of networked systems in which the complexity of fault scenarios increases with the number of subsystems and their interconnection structure. Due to the interconnection any compensation of the fault has to consider that a fault does not only affect the faulty subsystem but also spreads out to others. Thus, although fault-tolerant control of networked systems must consider the interconnection as an additional element, the ability to communicate between subsystems also increases the possible counteractive measures as well.

1.1.5 Network of Dynamical Systems

Whenever the complexity of a system, for example, with respect to its state dimension, geographical expansion or due to privacy policies exceeds a certain level, it is suitable to consider the overall system as a combination of subsystems, which results in a network of dynamical systems or networked systems, respectively.

Figure 1.5 shows the structure of such a network as it is used in this thesis. The overall system P consists of the subsystems P_i, $(i = 1, 2, \ldots, N)$, which are permanently connected to each other via a physical network. In this context, permanent means that a connection between two subsystems can neither be removed nor added. These are typically physical interconnections like a pipe connecting two tanks or a spring connecting two masses. The subsystems P_i are said to be coupled (or interconnected) by a physical network. The overall controller C has a very similar structure to P. Local control units C_i^*, $(i = 1, 2, \ldots, N)$, are assigned to the subsystems and connected to each other. Their interconnections arise from a communication network, which, unlike the physical network, allows variable connections. This means that two controllers can decide both whether and which information they want to exchange. In order to emphasize that some signals are sent via the communication network, all communication links are drawn as dashed lines instead of solid ones (see Fig. 1.5).

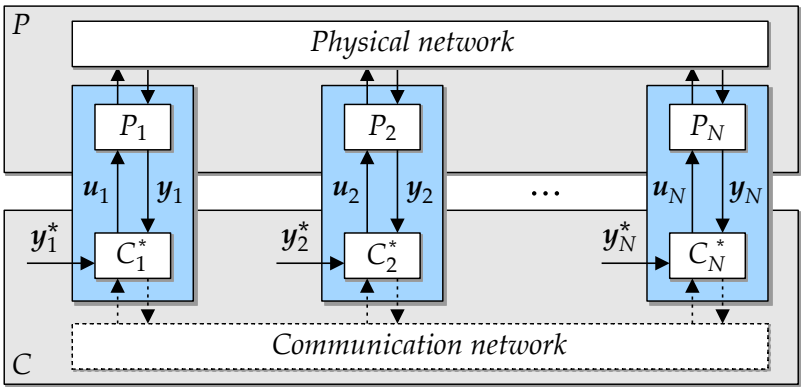

Fig. 1.5: The subsystems $P_1, P_2, \ldots, P_N$ are equipped with local controllers $C_1^*, C_2^*, \ldots, C_N^*$ and form individual control loops. On the plant-side, the subsystems are physically coupled while on the controller-side a communication network allows to share information among the local controllers.

The subtasks generated by the task assignment unit (cf. Fig. 1.3) describe the reference trajectories $\boldsymbol{y}_i^*(t)$ in Fig. 1.5. This thesis proposes two methods for the design of the networked controller C_i^*, $(i = 1, 2, \ldots, N)$. Both methods rely on a combination of feedforward controllers and feedback controllers. When neither model uncertainties nor disturbances are present, the feedforward controllers will guarantee that the subsystem outputs $\boldsymbol{y}_i(t)$ coincide with the reference trajectories. However, to make the tracking

more robust, feedback controllers are used on top. The design of the controllers C_i^* is an interesting topic on its own. Chapters 5 and 6 are dedicated exclusively to this problem.

There are two approaches on how to look at networked systems. In the *top-down* approach, a large problem is broken down into smaller ones to make it solvable at all. For example, in distributed optimization the overall optimization problem is solved indirectly by decomposing it into several smaller optimization problems [123]. The top-down approach is illustrated in Fig. 1.5 by the two gray boxes that show how the plant P and the controller C are separated into smaller units. In contrast, the *bottom-up* approach focuses on the subsystems, shown by the blue boxes in Fig. 1.5. There is a greater emphasis on thinking and reasoning from a local perspective. Typically, the subsystems know of their mutual existence, but not of their exact models or signals. They have only local information available. In this thesis both approaches are used due to the fact that there are some problems, like the decomposition problem that can be solved only from a global perspective (top-down approach) while there are others, like the tracking control problem that benefit from a local perspective (bottom-up approach). However, no matter which approach is used, the following statement is true:

> The communication network is seen as a chance to solve control tasks that would be impossible (or much more difficult) to solve without the network.

Although the availability of a communication network is essential, this thesis is not intended to deal with the problems caused by faulty or restricted communication. For this reason, the following assumption is made:

Assumption 1.1 (Ideal communication)**.** The communication network is ideal. Any data exchanged between two subsystems among the communication network is neither delayed, corrupted, quantized or modified in any way that changes the data.

Remark. Throughout this thesis, the term *networked systems* is used to refer to the overall system that consists of several subsystems. In the literature, which will be examined in Section 1.3, *multi-agent systems*, *networked control systems*, and *large-scale systems* have similar, though not identical, structures to those shown in Fig. 1.5. Each of these topics has a different focus than this thesis does.

1.2 Aim of This Thesis

The aim of this thesis is to develop methods for the fault-tolerant control of networked systems in which fault tolerance is achieved by redistributing subtasks from faulty to healthy subsystems (Fig. 1.6). The cooperative task refers to a global performance output

$$\boldsymbol{p}(t,\sigma) = \boldsymbol{Q}(\sigma) \cdot \begin{bmatrix} \boldsymbol{y}_1(t) & \boldsymbol{y}_2(t) & \cdots & \boldsymbol{y}_N(t) \end{bmatrix}^{\mathrm{T}}$$

that depends on the subsystem outputs $\boldsymbol{y}_i(t)$, $(i = 1, \ldots, N)$. The matrix $\boldsymbol{Q}(\sigma)$ represents the contribution of each subsystem to the cooperative behavior. As the previously given transportation system example has shown, subsystems cannot influence the cooperative task at all time, which is taking into account by using the switching state $\sigma(t)$. The cooperative task of all subsystems is to steer $\boldsymbol{p}(t, \sigma)$ along a prescribed reference trajectory $\boldsymbol{p}^*(t)$:

$$\boldsymbol{p}(t, \sigma) = \boldsymbol{p}^*(t). \tag{1.1}$$

Faults affecting the subsystems, say at time t_f, should be tolerable by the overall system. Recall, a fault tolerant system must prevent a fault from becoming a failure. Specifically, this thesis considers faults of a stuck-at type. A subsystem with a stuck-at fault remains in its current state, no matter what input is given to that subsystem. Hence, the output of a subsystem P_f being in such a stuck-at situation satisfies $\dot{\boldsymbol{y}}_f(t) = \boldsymbol{0}$ for all $t \geq t_\mathrm{f}$. The failure that should be prevented is that the performance output deviates from its reference:

$$\boldsymbol{p}(t, \sigma) \neq \boldsymbol{p}^*(t) \quad \text{for all} \quad t \geq t_\mathrm{f}. \tag{1.2}$$

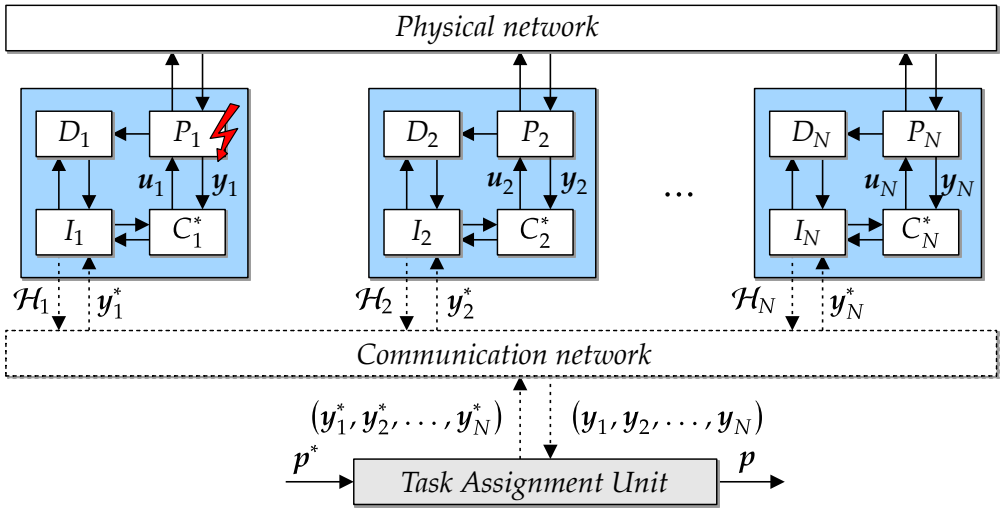

Fig. 1.6: Structure of a fault-tolerant network of dynamical systems.

Control signals $\boldsymbol{u}_i(t)$, $(i = 1, 2, \ldots, N)$ have to be found steering the subsystem outputs $\boldsymbol{y}_i(t)$ in a way such that the cooperative task (1.1) is satisfied and the failure (1.2) is prevented. The proposed control structure is shown in Fig. 1.6 and consists of different components working together:

Task assignment: The reference $\boldsymbol{p}^*(t)$ is decomposed into suitable local references $\boldsymbol{y}_i^*(t)$, $(i = 1, 2, \ldots, N)$, taking into account the switching character of the overall system, the subsystems capabilities and faults that might occur. This decomposition is done by a global task assignment unit (TAU), which is the only unit that has access to the performance output $\boldsymbol{p}(t, \sigma)$.