Ruhr-Universität Bochum
Institute of Automation and Computer Control
Prof. Dr.-Ing. J. Lunze

Universitätsstrasse 150
D-44805 Bochum, Germany

Phone +49 - (0)234 32 - 28071
Fax +49 - (0)234 32 - 14101

Fault-tolerant control of discrete-event systems

Dipl.-Ing. Yannick Nke
nke@atp.rub.de

1 Introduction

The fault-tolerant architecture used in this project is presented in Figure 1. The discrete-event controller generates a control sequence $V_P(0 \cdots k)$ for the plant according to its given input sequence $V_C(0 \cdots k)$ to satisfy the specification $\mathcal{S}$ e.g. to reach a final state z_F. In the nominal case, the plant produces an output sequence $W_P(0 \cdots k)$ which is directed back to the controller and the diagnoser. If a fault has been diagnosed, an alarming symbol f triggers the control reconfiguration unit in order to adapt the control law to the fault.

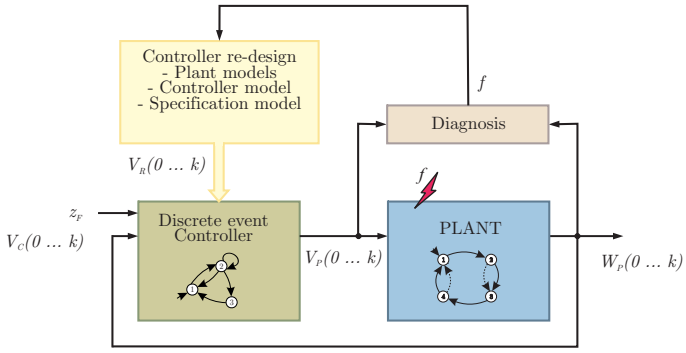

Figure 1: Discrete event fault-tolerant control

2 Project aims

The general objective of the project is to maintain the system at work despite the fault by rationally using physical redundancies. The following theoretical aims are pursued:

1. Suitable formalism for control design, fault modeling and control reconfiguration.
2. Fulfillment of the specification by the plant in the control loop despite the fault.
3. Nonblockingness of the control loop.

3 Modeling

The plant model consists of a nondeterministic I/O automaton $\mathcal{N}_p = (\mathcal{Z}_p, \mathcal{V}_p, \mathcal{W}_p, L_p, \mathcal{Z}_{0p})$. $\mathcal{Z}_p$, $\mathcal{V}_p$, $\mathcal{W}_p$, L_p, and $\mathcal{Z}_{0p}$ respectively represent the set of states, the set of inputs, the set of outputs, the characteristic function and the set of initial states. The characteristic function L_p exhibits the dynamic of the system in the nominal case. For visualization purposes I/O automata graphs are used similarly to Fig. 3(a) but with I/O pairs as transition labels. Alternatively, trellis graphs (Fig. 3(c)) are the unfolded representation of the former and reflect the dynamic of the plant. To reduce complexity issues, a modular modeling framework is used in [1], where the system is considered as a group of interconnected components in contrast to the monolithic approach. The concepts of well-posedness are used as a necessary condition in order to compose the deterministic automata network $\mathcal{DAN}$ into a corresponding automaton $\mathcal{A}$ having the same behavior [1].

However, the given specification to be fulfilled by the plant such as operating and safety constraints also need to be modeled. In this project, the following items may be used to model specifications $\mathcal{S}$:

- a specific final state z_F to reach,
- a state sequence $Z_s(0 \cdots k_e)$ to follow,
- an output sequence $W_s(0 \cdots k_e)$ to generate,
- a handicap specification for a faulty state z^f, a faulty input v^f and a faulty output w^f "imposed" by a fault to the plant.

4 Control design

The basic feasibility of the specification $\mathcal{S}$ was first introduced in [2] but does not guarantee a safe feasibility. That is to ensure that the plant can not deviate from the specified trajectory because of its nondeterministic dynamic.

Basic feasibility of a specification. It is a necessary condition for the existence of a controller to enforce a specification $\mathcal{S}$ in a given system $\mathcal{N}_p$. A specification automaton $\mathcal{N}_s \subseteq \mathcal{N}_p$ contains any transition of $\mathcal{N}_p$ which is line with the specification $\mathcal{S}$. The existence of a homomorphism from $\mathcal{N}_s$ to $\mathcal{N}_p$ has been proved in [2] to be a necessary feasibility condition.
Current investigations are devoted to the **safe** feasibility condition as a sufficient condition, which will ensure the fulfillment of the specification despite the nondeterministic nature of the plant. If the specification is feasible, the controller is obtained by keeping the structure of $\mathcal{N}_s$ while inverting the I/O behavior with

$$L_c(z', w_s, z, v_s) = L_s(z', v_s, z, w_s)$$
$$\forall (z', v_s, z, w_s) \in \mathcal{Z}_s \times \mathcal{V}_s \times \mathcal{Z}_s \times \mathcal{W}_s. \quad (1)$$

Controllability. In order to avoid blocking situations in the control loop it is necessary to enforce a *deterministic* behavior of the control loop through the controller outputs because they are the only signals which can be manipulated by the system designer. A controller having this property is said to be W-deterministic. The latter is used in [3] as necessary and sufficient condition to achieve the controllability of a plant $\mathcal{N}_p$ w.r.t. a specification $\mathcal{S}$.

5 Control reconfiguration

The reconfiguration of a controller requires the knowledge of the effects of the fault on the dynamics of the plant. Hence the modelling of the faulty plant is crucial for reconfiguration.

Fault modeling. When an actuator, a sensor or a system internal fault or failure occurs, the model $\mathcal{N}_p$ is no longer valid to capture the behavior of the faulty plant. Instead, a new model of the faulty plant $\mathcal{N}_p^f$ needs to be built. This problem is handled in [4] by means of suitable error functions modeling faults and failures.

Reconfiguration methods. Offline and online reconfiguration concepts have been proposed in [2] and [5] respectively. The former is based on a global search for redundancies whereas the latter is restricted to a local search. The methods applied in both cases consist of a *trajectory re-planning* and an *Input/Output adaptation*. The special case of cyclic processes is formalized in [6] with a rigorous formalism applied on unfolded I/O automata. The **reconfigurability** of a controller is then formulated as the controllability of a faulty plant w.r.t. the same specification $\mathcal{S}$ as in the nominal case.

6 Example: Manufacturing process subject of faults

The following example demonstrates how fault tolerance can be achieved from a discrete-event system point of view. The specification of the process is to transport workpieces from Belt 1 to Belt 2 with a time constraint (e.g a given deadline) using the robot arm as depicted on Fig. 2. The nominal behavior of the process is first modeled by a standard automaton in Fig. 3(a) representing the nominal behavior. Figure 3(c) shows the corresponding unfolding as a trellis graph. Although I/O automata are used in this project for modeling, standard automata are considered in the sequel to explain fault tolerance because of their simplicity. Thus the nominal behavior is described by the following event sequence: $\sigma_G, \sigma_D, \sigma_A, \sigma_Z, \sigma_G, \cdots$. To fulfill the specification, the state sequence $Z_s(0 \cdots 3) = (1, 2, 3, 4)$ is to perform.
Suppose a fault making the events σ_D and σ_Z unavailable occurs e.g. due to blocking servo motors. The remaining redundant events $\bar{\sigma}_D$ and $\bar{\sigma}_Z$ need to be used in order to fulfill the specification $Z_s(0 \cdots 3)$. Fig. 3(b) illustrates this typical fault-tolerant behavior. The specification might still be fulfilled but degraded in the sense that a time delay have to be taken into account.

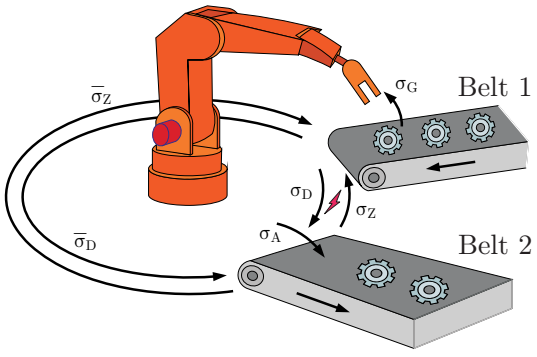

Figure 2: Process under a fault

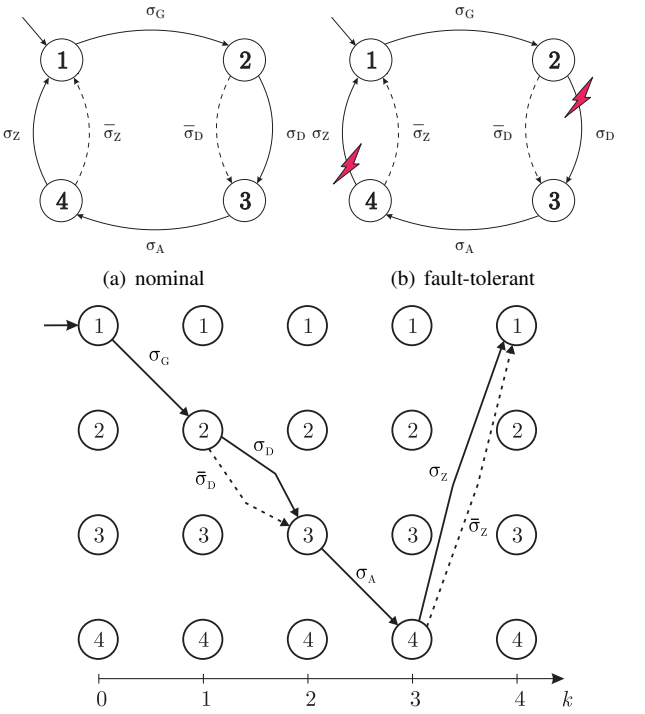

Figure 3: Discrete-event based fault tolerance

References

[1] Y. Nke, S. Drüppel, and J. Lunze. Direct feedback in asynchronous networks of input-output automata. In *Proc. 10th European Control Conference*, Budapest, Hungary, 2009.

[2] Y. Nke and J. Lunze. Fault-tolerant control of nondeterministic input/output automata subject to actuator faults. In *Proceedings of the 10th International Workshop on Discrete Event Systems*, Berlin, Sep 2010.

[3] Y. Nke and J. Lunze. Control design for nondeterministic input/output automata. In *Proceedings of the 18th IFAC Congress*, Milan, 2011.

[4] Y. Nke and J. Lunze. A fault modeling approach for input/output automata. In *Proceedings of the 18th IFAC Congress*, Milan, 2011.

[5] Y. Nke and J. Lunze. Online control reconfiguration for a faulty manufacturing process. In *3rd International Workshop on Dependable Control of Discrete Systems (DCDS)*, Saarbrücken, Germany, 2011.

[6] Y. Nke and J. Lunze. Control reconfiguration based on unfolding of input/output automata. In *Proceedings of the 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*, Mexico City, Mexico, 2012. accepted.