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

Universitätsstr. 150 Phone +49-(0)234 32-28071
D-44805 Bochum Fax +49-(0)234 32-14101

Fault-Tolerant Control of Discrete Event Systems

Melanie Schuh
melanieschuh@atp.rub.de

1 Introduction

The fault-tolerant control (FTC) architecture for discrete event systems (DES) used in this project is presented in Fig. 1. It is intended to control the plant $\mathcal{P}$ in a way that certain objectives are fulfilled, even when the plant is subject to a fault. For example, the plant should always reach a desired final state z_{F}. To achieve this goal, an occurring fault f has first to be detected and then to be identified, before the nominal controller $\mathcal{C}$ can be reconfigured to restore the functionality of the closed-loop system.

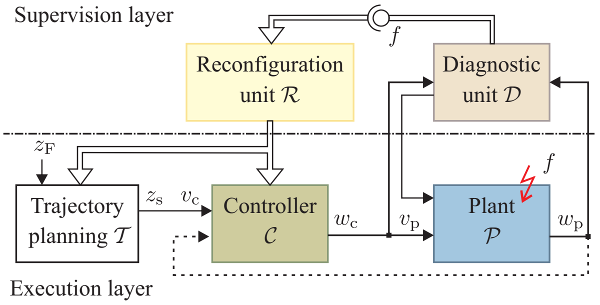

Figure 1: Fault-tolerant control loop

Faults can affect the actuators, the sensors or the internal system behavior of a given plant. In order to recover the plant from a present fault, existent redundancies have to be employed. For example, if a storage position is blocked, an alternative position (which might be harder to access) could be used to keep the process running.

2 Project aim

The main goal of this project is to develop an overall concept for the fault-tolerant control of discrete event systems that takes into account both, the fault diagnosis and the control reconfiguration. To reach this goal, the following question has to be answered:

"How need the elements of the fault-tolerant control loop to be designed such that the plant recovers from a present fault?"

Closely related is the question, under which conditions the aforementioned problem can be solved, that is, under which conditions the plant $\mathcal{P}$ is recoverable from a given fault f.

In [3], an intuitive solution for combining fault diagnosis with reconfiguration to a FTC framework for discrete event systems has been outlined. This project aims at formalizing the results found there, such that the correctness of the method can be formally proved.

In order to answer the main question raised above, the following points need to be considered:

- Conditions for the diagnosability of the plant $\mathcal{P}$
- A method for the efficient fault identification
- A method to decide which information to send from the diagnostic unit $\mathcal{D}$ to the reconfiguration unit $\mathcal{R}$
- Conditions for the reconfigurability of the controller
- A method for the reconfiguration of the relevant part of the controller $\mathcal{C}$.

3 Nominal control loop

The nominal control loop can be seen in the execution layer in Fig. 1. It consists of the plant $\mathcal{P}$, the controller $\mathcal{C}$ and the trajectory planning unit $\mathcal{T}$.

The plant $\mathcal{P}$ is described by a deterministic input/output (I/O) automaton

$$\mathcal{A}_{\mathrm{p}} = (\mathcal{Z}_{\mathrm{p}}, \mathcal{V}_{\mathrm{p}}, \mathcal{W}_{\mathrm{p}}, G_{\mathrm{p}}, H_{\mathrm{p}}, z_{\mathrm{p0}})$$

with state set $\mathcal{Z}_{\mathrm{p}}$, input set $\mathcal{V}_{\mathrm{p}}$, output set $\mathcal{W}_{\mathrm{p}}$, state transition function G_{p}, output function H_{p} and initial state z_{p0}. Based on the model $\mathcal{A}_{\mathrm{p}}$ of the plant, the controller $\mathcal{C}$ is designed, which is defined in form of a deterministic I/O automaton $\mathcal{A}_{\mathrm{c}}$ as well. The trajectory planning unit $\mathcal{T}$ specifies a path for the plant $\mathcal{P}$ from its initial state z_{p0} into the desired final state z_{F}.

Using the controller design method described in [4], it can be shown that the resulting controller automaton $\mathcal{A}_{\mathrm{c}}$ steers the plant automaton $\mathcal{A}_{\mathrm{p}}$ exactly along the trajectory planned by $\mathcal{T}$ into the desired final state z_{F}. As the plant $\mathcal{P}$ is deterministic, the controller $\mathcal{C}$ does not need to have access to the output w_{p} of the plant. Rather, it suffices to control the plant in an open-loop manner.

4 Fault-tolerant control loop

The elements of the fault-tolerant control loop have to be designed such that after the fault diagnosis and the reconfiguration of the controller, the faulty plant again reaches the desired final state z_{F}.

Desired behavior of the FTC loop. In nominal operation, the controller $\mathcal{C}$ steers the plant $\mathcal{P}$ along a trajectory generated by the trajectory planning unit $\mathcal{T}$ into the desired final state $z_{\rm F}$. The diagnostic unit $\mathcal{D}$ monitors the inputs $v_{\rm p}$ and outputs $w_{\rm p}$ of the plant in order to detect whether a fault has occurred. If a fault f is detected, the diagnostic unit aims at identifying it. Afterwards, the reconfiguration unit $\mathcal{R}$ adapts the nominal controller $\mathcal{C}$ and initiates a replanning of the trajectory such that the faulty plant again reaches the desired final state $z_{\rm F}$.

Fault diagnosis. The behavior of the plant under different faults $f \in \mathcal{F}$ is described by a set of deterministic I/O automata $\{\mathcal{A}_f : f \in \mathcal{F}\}$. For the fault diagnosis, whose goal is to identify the fault $f \in \mathcal{F}$ that is present at the plant, two different model-based approaches are possible.

On the one hand, the diagnostic unit $\mathcal{D}$ can perform a classical consistency-based diagnosis. That is, it evaluates the inputs $v_{\rm p}$ to the plant that are generated by the controller $\mathcal{C}$ and the resulting outputs $w_{\rm p}$ of the plant based on the models $\mathcal{A}_f$, $(f \in \mathcal{F})$ of the faulty system.

Contrarily, an active diagnosis method such as the one presented in [1] can be used. In this case, the inputs for the faulty plant are no longer generated by the controller $\mathcal{C}$, but by the diagnostic unit $\mathcal{D}$, which aims at steering the plant such that different faults become distinguishable. It is also possible to perform an active diagnosis, if the actions of the diagnostic unit have to be restricted due to safety constraints (cf. [2]).

Reconfiguration. During the reconfiguration, the controller $\mathcal{C}$ has to be adapted to the present fault $f \in \mathcal{F}$. Furthermore, the trajectory to the desired final state $z_{\rm F}$ has to be replanned based on the current state of the faulty plant and its model $\mathcal{A}_f$, $(f \in \mathcal{F})$.

Usually, the fault $f \in \mathcal{F}$ does not change the entire model $\mathcal{A}_{\rm p}$ of the nominal plant, but only influences it at certain points. Therefore, it is not necessary to design a completely new controller for the faulty plant. Rather, only those transitions in the controller automaton $\mathcal{A}_{\rm c}$ shall be redesigned, which are really affected by the fault.

5 Example

Consider the fault-tolerant control of the manufacturing cell in Fig. 2, through which workpieces are transported by a conveyor belt. The control aim is to twist a screw into the workpieces.

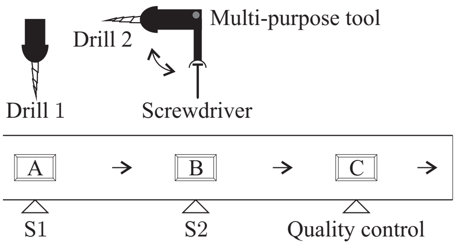

Figure 2: Manufacturing cell

In nominal operation, a hole is drilled into the workpiece by drill 1 at position A and a screw is twisted into the resulting thread using the screwdriver of the multi-purpose tool at position B. Afterwards, the quality of the processed workpiece is checked at position C. If the workpieces are improperly drilled, because drill 1 is faulty, their quality can not be maintained. Therefore, the fault will be detected during the quality control. When drill 1 is identified to be the faulty component, drill 2 of the multi-purpose tool can be used for the drilling task instead.

Figure 3 shows the nominal controller automaton, whose states correspond to the position and status of a workpiece at the conveyor belt. The inputs to the controller automaton are the desired next states for the plant from the trajectory planning unit $\mathcal{T}$, while its outputs are the inputs to the plant (see Fig. 1). There are two possible paths for reaching the desired final state $z_{\rm F} = {\rm C}$ from the initial state $z_{\rm p0} = {\rm A}$. In the upper path, drill 1 is used, while in the lower path the multi-purpose tool is switched to drill 2, which is then employed for the drilling task. In nominal operation the trajectory planning unit specifies the usage of the first path, because it aims at minimizing the necessary number of steps.

If the aforementioned fault occurs, the dashed (red) transition has to be removed from the controller automaton in Fig. 3, because the fault prevents the usage of drill 1. The remainder of the controller may remain unchanged. Now the alternative path has to be used, that is, the multi-purpose tool has to be switched to drill 2, use it for the drilling task and switch back to the screwdriver afterwards. Then the control aim can still be reached, at the expense of a higher production time due to the necessary switching between the different tools of the multi-purpose tool.

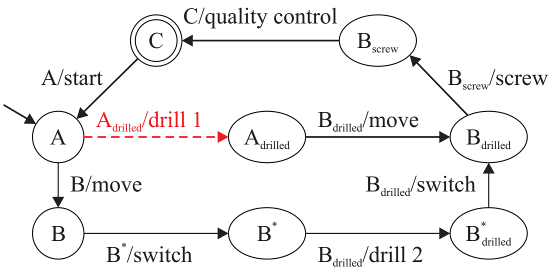

Figure 3: Controller automaton

References

[1] M. Schmidt and J. Lunze. Active diagnosis of deterministic I/O automata. In *4th IFAC Workshop on Dependable Control of Discrete Systems*, York, UK, 2013.

[2] M. Schmidt and J. Lunze. Active fault diagnosis of discrete event systems subject to safety constraints. In *2nd International Conference on Control and Fault-Tolerant Systems*, Nice, France, 2013.

[3] M. Schmidt and J. Lunze. A framework for active fault-tolerant control of deterministic I/O automata. In *12th IFAC - IEEE International Workshop on Discrete Event Systems*, Paris, France, 2014.

[4] M. Schuh and J. Lunze. Feedback control of nondeterministic Input/Output automata. In *53rd IEEE Conference on Decision and Control*, Los Angeles, USA, 2014.