Fault-Tolerant Supervisory Control in Terms of Formal Languages

Thomas Moor
Friedrich-Alexander Universität-Erlangen-Nürnberg, Germany

IFAC WC 2020, Berlin, Germany

Discrete-Event Systems

Supervisory Control

Naive Fault-Tolerant Control

Active Fault-Tolerant Control

Post-Fault Recovery

Fault-Hiding Approach

Formal Languages

- Σ^* set of all finite strings over Σ
- empty string $\epsilon \in \Sigma^*$, $\epsilon \notin \Sigma$
- $*$-language $M \subseteq \Sigma^*$
- prefix operator $\operatorname{pre} s := \{\, t \mid \exists\, r : tr = s \,\}$
- note that $\operatorname{pre} M = \{\, t \mid \exists\, r : tr \in M \,\}$
- M is closed $:\Leftrightarrow \;\; M = \operatorname{pre} M$

A natural domain for the interpretation of liveness properties are ω-languages, i.e., sets of infinite-length strings $w \in \Sigma^\omega$.

If there are no deadlocks, we may use

$$\mathcal{M} := \{w \in \Sigma^\omega \mid \; \operatorname{pre} w \subseteq \operatorname{pre} M \;\}$$

to model the process w.r.t. infinite time.

If, in addition, there are no livelocks, we may consider

$$\mathcal{L} := \{w \in \Sigma^\omega \mid \; \|(\operatorname{pre} w) \cap L\| = \infty\}.$$

to model the process w.r.t. infinite time.

Control patterns

$$\Gamma := \{\, \gamma \subseteq \Sigma \mid \Sigma_{uc} \subseteq \gamma \,\}$$

Projection

- natural projection $p_o : \Sigma^* \to \Sigma_o^*$
  read "removes all symbols not from Σ_o"
- for languages take pint-wise images
- set-valued inverse $p_o^{-1} : \Sigma_o^* \rightsquigarrow \Sigma^*$
  read "arbitrarilly inserts symbols from Σ_{uo}"

Language Quotient

$$K/E := \{\, s \mid \exists\, t \in E \;:\; st \in K \,\}\,.$$

Language Convergence

K *finitely converges* to E if there exists a uniform bound k such that every $s \in K$ can be decomposed

$$s = vw,\; w \in E, \text{ and } |v| \leq k.$$

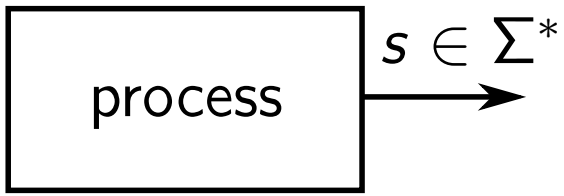
This is written $E \Leftarrow K$.

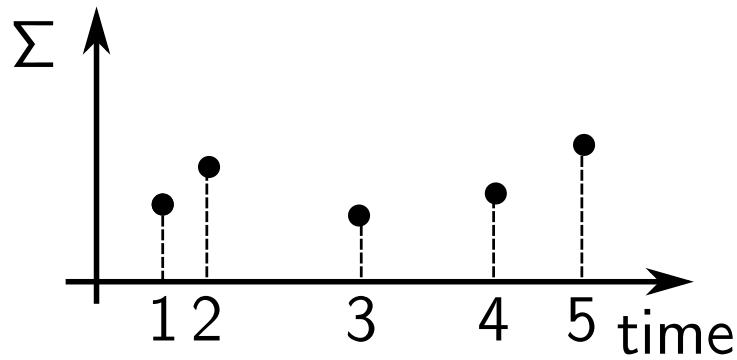
For not-uniformly bounded convergence, one refers to the respective ω-languages and requires

$$\lim K \subseteq \lim(\Sigma^* E)\,.$$

A discrete-event system is a model of a process ... with a particular focus on the ocurrence of events

- finite set Σ of symbols $\sigma \in \Sigma$
- only event ordering is regarded relevant (logic time)
- within finite physical time a finite sequence $s \in \Sigma^*$ is generated
- set $M \subseteq \Sigma^*$ of sequences that can be generated
- write pre M to emphasise that $M = \text{pre}\, M$ (local behaviour)





A closed language pre $M \subseteq \Sigma^*$ is a discrete-event system.

Properties

- safety – bad things never happen

  with $\operatorname{pre} E \subseteq \Sigma^*$, require

  $$\operatorname{pre} M \subseteq \operatorname{pre} E$$

- liveness – good things do happen

  free of deadlocks

  $$(\forall\, s \in \operatorname{pre} M)(\exists\, \sigma \in \Sigma)[\, s\sigma \in \operatorname{pre} M\,]$$

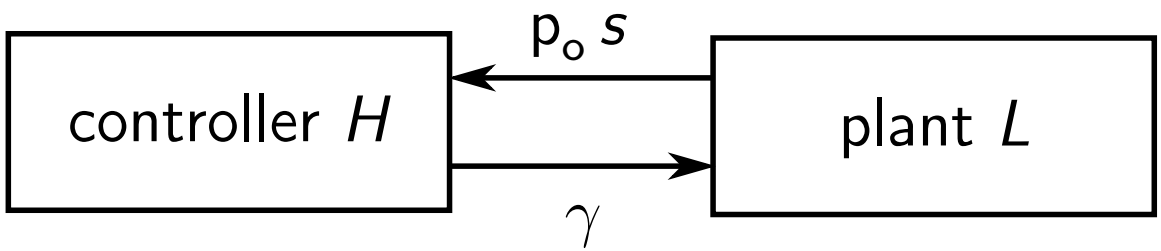
  free of livelocks w.r.t. $L \subseteq \operatorname{pre} M$

  $$(\forall\, s \in \operatorname{pre} M)(\exists\, t \in \Sigma^*)[\, st \in L \cap \operatorname{pre} M\,]$$

For systems with liveness properties:

A language $L \subseteq \Sigma^*$ is a discrete-event system.

W.r.t. the partitioning $\Sigma = \Sigma_c \dot\cup \Sigma_{uc} = \Sigma_o \dot\cup \Sigma_{uo}$ consider a plant $L \subseteq \Sigma^*$ and a controller H in closed-loop configuration:



- at any time, the controller is provided $\mathrm{p}_o\, s \in \Sigma_o^*$ where $s \in \Sigma^*$ is the sequence generated so far;
- in turn, the controller applies a control pattern $\gamma \in \Gamma$ of enabled events, where $\Sigma_{uc} \subseteq \gamma$;
- represent the controller as a discrete-event system

  $$H \subseteq \Sigma^* .$$

  i.e. $\gamma = \{\, \sigma \mid s\sigma \in H \,\}$.

Def. A controller $H \subseteq \Sigma^*$ is *admissible w.r.t. the plant* $L \subseteq \Sigma^*$, if

[H0] $H = \operatorname{pre} H$,

[H1] $(\operatorname{pre} H)\Sigma_{uc} \subseteq \operatorname{pre} H$,

[H2] $\operatorname{pre} H = p_o^{-1}\, p_o \operatorname{pre} H$ (... assuming $\Sigma_c \subseteq \Sigma_o$),

[H3] $(\operatorname{pre} L) \cap (\operatorname{pre} H)$ does not deadlock, and

[H4] $(\operatorname{pre} L) \cap (\operatorname{pre} H) = \operatorname{pre}(L \cap H)$.

Then $K := L \cap H$ represents the cosed-loop behaviour. □

Structural requirement [H4&5]:
liveness properties of the plant shall be retained.

Thm [SCT]: For a plant $L \subseteq \Sigma^*$ and an admissible controller $H \subseteq \Sigma^*$ let $K = L \cap H$. Then

[K0] K is relatively closed w.r.t. L,

[K1] K is controllable w.r.t. L,

[K2] K prefix-normal w.r.t. L, and

[K3] K does not deadlock.

- K is rel. closed w.r.t. L iff
  $K = (\mathrm{pre}\, K) \cap L$
- K is controllable w.r.t. L iff
  $((\mathrm{pre}\, K)\Sigma_{\mathrm{uc}}) \cap (\mathrm{pre}\, L) \subseteq \mathrm{pre}\, K$
- K is prefix normal w.r.t. L iff
  $\mathrm{pre}\, K = (\mathrm{p_o}^{-1}\, \mathrm{p_o}\, \mathrm{pre}\, K) \cap (\mathrm{pre}\, L)$
- K does not deadlock iff
  $\forall\, s \in \mathrm{pre}\, K\, \exists\, \sigma \in \Sigma\colon s\sigma \in \mathrm{pre}\, K$

Vice versa, if K satisfies [K0]-[K3], then there exists an admissible controller H such that $K = L \cap H$. □

Control Problem: given (L, E) with plant $L \subseteq \Sigma^*$ and a specification $E \subseteq \Sigma^*$ construct an admissible controller $H \subseteq \Sigma^*$ such that

$$K := L \cap H \subseteq E\,.$$

Solution: all closed-loop properties are retained under arbitraty union; thus

$$K^{\uparrow} = \sup\{K \subseteq L \cap E \mid K \text{ satisfies [K0]–[K3]}\}$$

itself satisfies [K0]–[K3] and is used to extract a mximally permissive controller.

Note: E can be substituted by a closed language without affecting solutions – it is effectivly a pure safety specification. This becomes a different story when considering ω-languages.

Fault-Tolerant Control

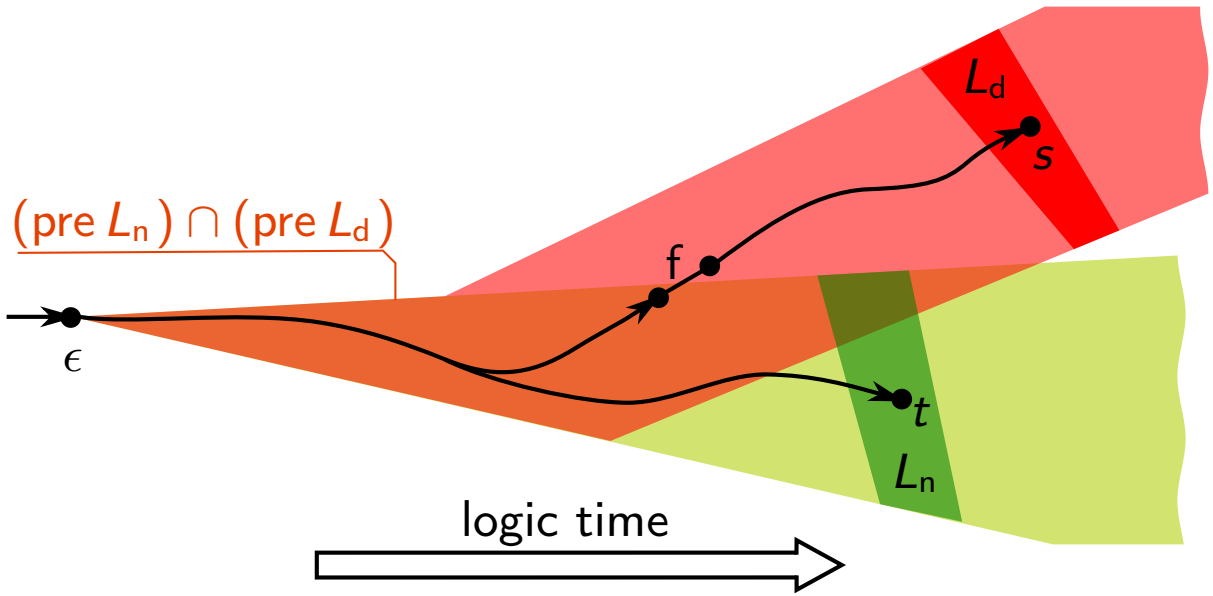
- a fault is a sudden change of behaviour
- passive approach: have a single controller that can handle pre-fault and post-fault behaviour (robust control)
- active approach: detect the fault and switch to another controller (adaptive control)

Core challenge for continuous control systems: switching of plant and controller dynamics and transient behaviour. However, for discrete-event systems:

Sudden change of behaviour and switching in the control scheme are the very nature of discrete-event systems. Hence, fault-tolerant control can be synthesised by the same methods as nominal control [??]

Naive approach to fault-tolerant control:

- nominal plant $L_n \subseteq \Sigma_n^*$
- fault event $f \notin \Sigma_n$, unctrl. and unobs., let $\Sigma_f := \Sigma_n \dot{\cup} \{f\}$
- degraded post-fault behaviour $L_d \subseteq (\mathrm{pre}\, L_n) f \Sigma_f^*$
- *fault-accommodating model* $L_f := L_n \cup L_d$



Algebraic consequence:

Prop. The prerequisite $L_d \subseteq (\text{pre}\, L_n)f\Sigma_f^*$ implies that:

$$L_d \cap \Sigma_n^* = \emptyset\,, \qquad (\text{pre}\, L_d) \cap \Sigma_n^* \subset \text{pre}\, L_n\,,$$
$$L_f \cap \Sigma_n^* = L_n\,, \qquad (\text{pre}\, L_f) \cap \Sigma_n^* = \text{pre}\, L_n\,.$$

From the last line we obtain $(\text{pre}\, L_f) \cap \Sigma_n^* = \text{pre}\, L_n = \text{pre}\,(L_f \cap \Sigma_n^*)$. I.e., the fault-accommodating model and the hypothesis the fault not to occur are non-conflicting. More general we require the fault to never become an inevetible consequence of the past event sequence.

Def. The fault-accommodating model L_f is *well-posed*, if

$$\forall\, s \in \text{pre}\, L_f\, \exists\, \sigma \in \Sigma_n : s\sigma \in \text{pre}\, L_f\,,$$
$$\forall\, s \in \text{pre}\, L_f\, \exists\, t \in \Sigma_n^* : st \in L_f\,.$$

Naive approach to fault-tolerant control (cnt.)

- fault-accommodating specs. $E_\mathrm{f} := E_\mathrm{n} \cup E_\mathrm{d}$, same spirit as L_f
- invoke std. synthesis procedure for $(L_\mathrm{f}, E_\mathrm{f})$
- obtain a controller H_f with sup. closed loop $K_\mathrm{f}^\uparrow = L_\mathrm{f} \cap H_\mathrm{f}$

However: we may encounter

$$\exists\, s \in \operatorname{pre} K_\mathrm{f} \quad \forall\, \sigma \in \Sigma_\mathrm{f} : \quad s\sigma \in \operatorname{pre} K_\mathrm{f} \;\Rightarrow\; \sigma = \mathrm{f}\,,$$
$$\exists\, s \in \operatorname{pre} K_\mathrm{f} \quad \forall\, t \in \Sigma_\mathrm{f}^* : \quad st \in K_\mathrm{f} \;\Rightarrow\; t \notin \Sigma_\mathrm{n}^*\,,$$

This is not desirable – impose additional requirements:

$$\text{[K4]} \quad \forall\, s \in \operatorname{pre} L_\mathrm{f} \quad \exists\, \sigma \in \Sigma_\mathrm{n} \quad : s\sigma \in \operatorname{pre} L_\mathrm{f}\,,$$
$$\text{[K5]} \quad \forall\, s \in \operatorname{pre} L_\mathrm{f} \quad \exists\, t \in \Sigma_\mathrm{n}^* \quad : st \in L_\mathrm{f}\,.$$

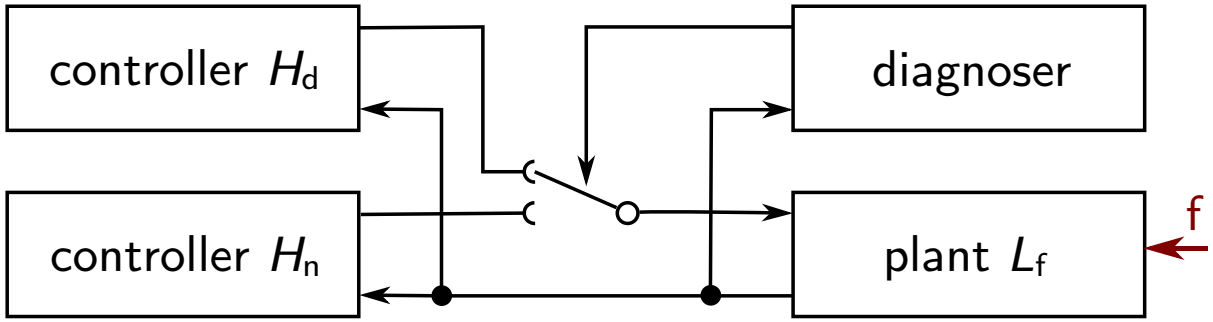
Naive approach to fault-tolerant control (cnt.)

- fault-accommodating specs. $E_\mathrm{f} := E_\mathrm{n} \cup E_\mathrm{d}$, same spirit as L_f
- invoke synthesis procedure for $(L_\mathrm{f}, E_\mathrm{f})$ incl. [K4] and [K5]
- obtain a controller H_f with sup. closed loop $K_\mathrm{f}^\uparrow = L_\mathrm{f} \cap H_\mathrm{f}$

Thm. [N-FTC]: Consider a persistent fault, $L_\mathrm{f} \subseteq \Sigma_\mathrm{n}^* \{\epsilon, \mathbf{f}\} \Sigma_\mathrm{n}^*$. Then there exists a controller H_f with $K_\mathrm{f} = L_\mathrm{f} \cap H_\mathrm{f}$ that is admissible to both L_f and L_n if and only if K_f satiesfies [K0]-[K5].

- diagnosability not required, passive fault-tolerant control
- in general, we have $L_\mathrm{n} \cap H_\mathrm{f} \subseteq K_\mathrm{n}^\uparrow$ — may compute $K_\mathrm{n}^\uparrow$ and test for equality.

Active fault-tolerant control

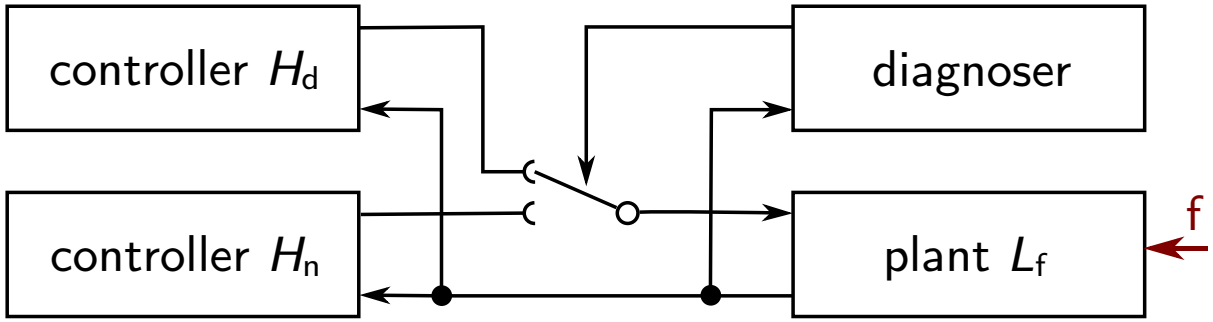


- require the fault to be diagnosable, denote $D \subseteq L_d$ the strings corresponding to f-certain diagnoser states

Diagnosis of DES (Sampath et al 1995)

- *diagnoser*: observer automaton with dedicated state labels
- *f-certain state*: state in which the fault must have occured some time ago
- *diagnosability*: require the plant to after the fault attain an f-certain state after a bounded number of transitions.

Active fault-tolerant control



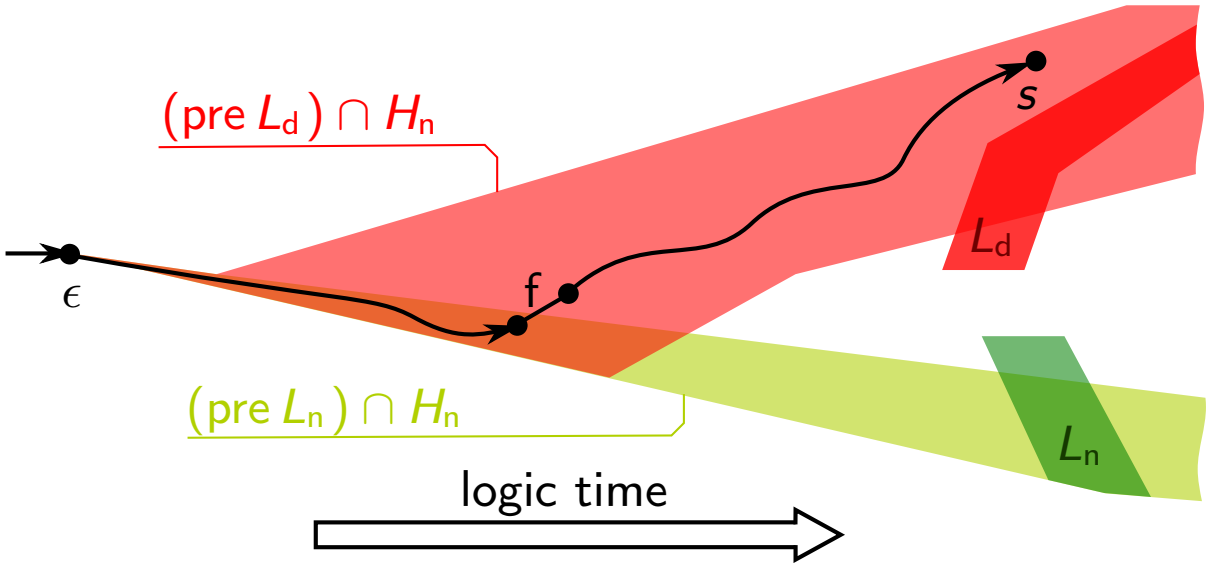
- require the fault to be diagnosable, denote $D \subseteq L_d$ the strings corresponding to f-certain diagnoser states
- require/test that the post-fault-pre-detection behaviour satisfies a safety specification (safe diagnosibility)
- design H_d to take over H_n when the plant first enters D
- note: nominal pre-fault behaviour is guaranteed
- option: synthesise H_d online once the fault has been diagnosed

- associate $H_n \hookleftarrow \mathrm{p}_n^{-1} H_n$ and consider the local-closed loop under nominal control $K_{\mathrm{loc}} := (\mathrm{pre}\, L_f) \cap H_n$
- safe diagnosability condition:

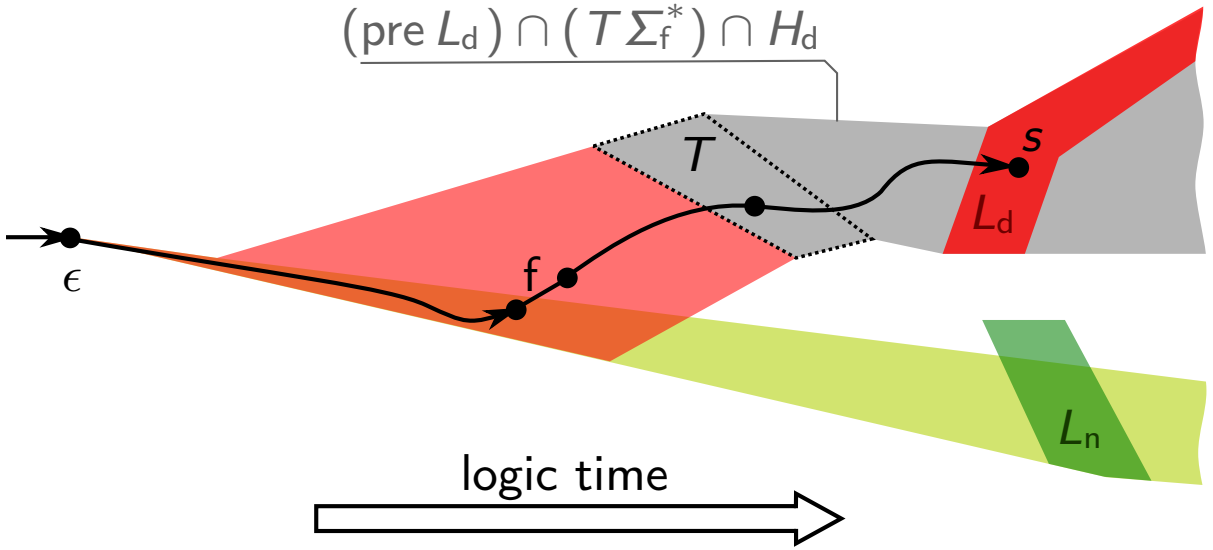
$$D := \{ s \in \Sigma_f^* \mid K_{\mathrm{loc}} \cap (\mathrm{p}_o^{-1} \mathrm{p}_o\, s) \subseteq \Sigma_n^* \mathrm{f} \Sigma_f^* \}\,,$$

$$K_{\mathrm{loc}} \cap \Sigma_n^* \mathrm{f} \Sigma_f^k \subseteq D \text{ for some } k \in \mathbb{N}\,,$$

$$T := \{ s \in K_{\mathrm{loc}} \mid (\mathrm{pre}\, s) \cap D = s \} \subseteq E_{\mathrm{phi}}\,;$$



- local post-fault-detection behaviour $(\mathrm{pre}\, L_\mathrm{d}) \cap (T\Sigma_\mathrm{n}^*)$
- post-fault-detection controller H_d requirements:
  - [A1] admissible w.r.t. $L_\mathrm{d} \cap (T\Sigma_\mathrm{n}^*)$
  - [A2] enforces post fault specs. $L_\mathrm{d} \cap (T\Sigma_\mathrm{n}^*) \cap H_\mathrm{d} \subseteq E_\mathrm{d}$
  - [A3] passive before fault-detection $T \subseteq (\mathrm{pre}\, L_\mathrm{d}) \cap (T\Sigma_\mathrm{n}^*) \cap H_\mathrm{d}$



- local post-fault-detection behaviour $(\mathrm{pre}\, L_\mathrm{d}) \cap (T\Sigma_\mathrm{n}^*)$
- post-fault-detection controller H_d requirements:
  - [A1] admissible w.r.t. $L_\mathrm{d} \cap (T\Sigma_\mathrm{n}^*)$
  - [A2] enforces post fault specs. $L_\mathrm{d} \cap (T\Sigma_\mathrm{n}^*) \cap H_\mathrm{d} \subseteq E_\mathrm{d}$
  - [A3] passive before fault-detection $T \subseteq (\mathrm{pre}\, L_\mathrm{d}) \cap (T\Sigma_\mathrm{n}^*) \cap H_\mathrm{d}$
- require that E_d and E_phi relate by

  $$(\mathrm{pre}\, T) \cap (\Sigma_\mathrm{n}^* \mathrm{f} \Sigma_\mathrm{n}^*) \subseteq E_\mathrm{d} \subseteq E_\mathrm{phi}$$

- post-fault-detection controller H_d, synthesis:
  - for [A1] and [A2] use std. procedure on $(L_\mathrm{d} \cap (T\Sigma_\mathrm{n}^*),\, E_\mathrm{d})$
  - then test for [A3]
  - if test fails, no solution exists

Re-interpret active FTC as naive FTC

- formally construct overall controller H_f:

$$H_\mathrm{f} := \{\epsilon\} \cup \{\, s\sigma \in H_\mathrm{n} \,|\, s \notin C \,\} \cup \{\, s\sigma \in H_\mathrm{d} \,|\, s \in C \,\},$$
$$C := (\, D \cap H_\mathrm{n} \cap H_\mathrm{d} \,) \cap \Sigma_\mathrm{f}^* \,.$$

Thm. [A-FTC]: Given a fault-accomodating model L_f and a nominal controller H_n admissible to $L_\mathrm{n} = L_\mathrm{f} \cap \Sigma_\mathrm{n}^*$, assume that local closed loop $K_\mathrm{loc} := (\mathrm{pre}\, L_\mathrm{f}) \cap H_\mathrm{n}$ is safe diagnosable. If a post-fault-detection controller H_d satisfies conditions [A1]–[A3], then the overall controller H_f defined above is admissible to both L_f and L_n with $L_\mathrm{n} \cap H_\mathrm{f} = L_\mathrm{n} \cap H_\mathrm{n}$.

- by Thm. [N-FTC] the conclusion is equivalent to K_f being a closed-loop behaviour achievable by naive FTC.

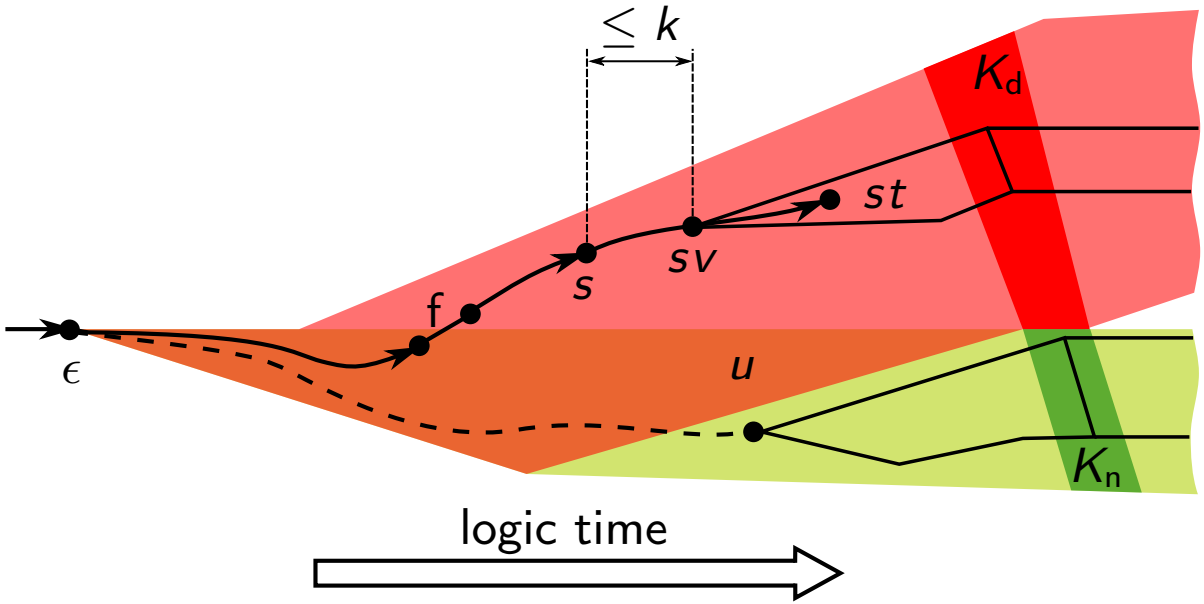
Post-Fault Recovery: safety

Def. [K6]: A closed loop $K_f = K_n \dot{\cup} K_d$ is *weakly recovering* if there exists a uniform bound k such that for all s, t, $|t| \geq k$ with

$$s \in (\operatorname{pre} K_f) \cap (\Sigma_n^* f \Sigma_n^*) \text{ and } st \in \operatorname{pre} K_f$$

there exists $u \in \operatorname{pre} K_n$, $v \in \operatorname{pre} t$, $|v| \leq k$ with

$$K_f/sv \subseteq K_n/u\,.$$



Post-Fault Recovery: safety

Def. [K6]: A closed loop $K_\mathrm{f} = K_\mathrm{n} \dot\cup K_\mathrm{d}$ is *weakly recovering* if there exists a uniform bound k such that for all s, t, $|t| \geq k$ with

$$s \in (\operatorname{pre} K_\mathrm{f}) \cap (\Sigma_\mathrm{n}^* \mathrm{f} \Sigma_\mathrm{n}^*) \text{ and } st \in \operatorname{pre} K_\mathrm{f}$$

there exists $u \in \operatorname{pre} K_\mathrm{n}$, $v \in \operatorname{pre} t$, $|v| \leq k$ with

$$K_\mathrm{f}/sv \subseteq K_\mathrm{n}/u\,.$$

- synthesis problem: given $L_\mathrm{f} = L_\mathrm{n} \cup L_\mathrm{d}$ and E_f, compute an admissible controller H_f such that the closed loop K_f satisfies [K6].
- the property is not retained under union; synthesis procedure exists for $\Sigma_\mathrm{o} = \Sigma$

Post-Fault Recovery: safety

- Weakly recovering [K6] implies conditional finite convergence:

  [K6'] $K_\mathsf{n}/\Sigma^* \Leftarrow K_\mathsf{f}/(\Sigma_\mathsf{n}^*\mathsf{f})$

- formally generalise to:

  [K6''] $E_\mathsf{f} \Leftarrow K_\mathsf{f}/(\Sigma_\mathsf{n}^*\mathsf{f})$

- synthesis problem: given $L_\mathsf{f} = L_\mathsf{n} \cup L_\mathsf{d}$ and E_f, compute an admissible controller H_f such that the closed loop K_f satisfies [K6']/[K6''].

- neither [K6''] nor [K6'] are retained under union; synthesis procedure exists.

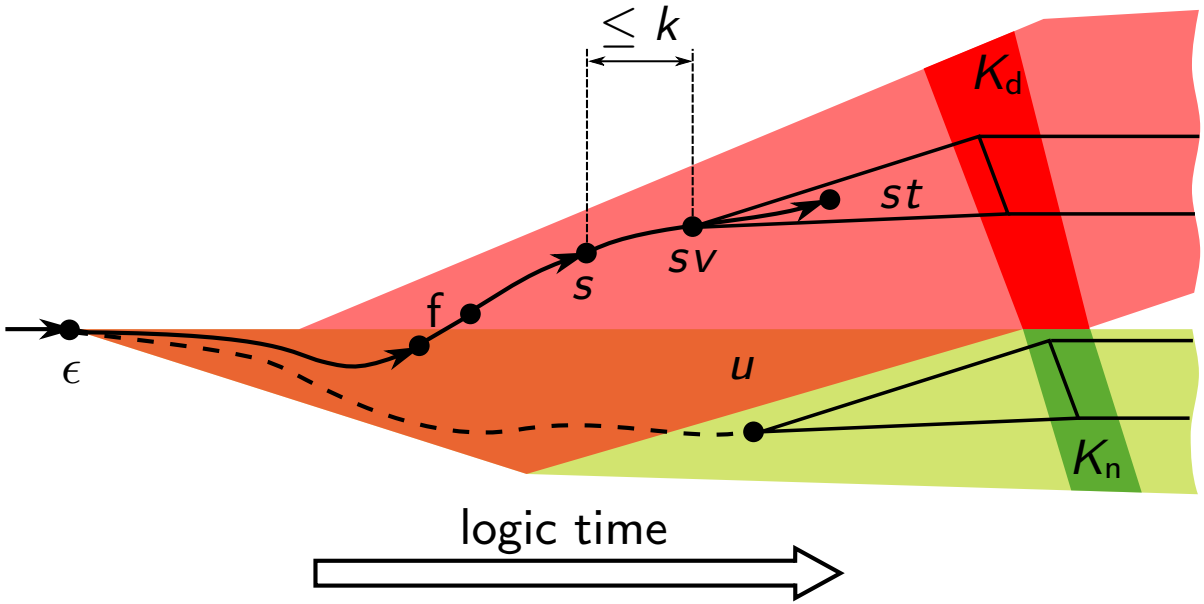
Post-Fault Recovery: liveness

Def. [K7]: A closed loop $K_f = K_n \dot{\cup} K_d$ is *weakly recovering* if there exists a uniform bound k such that for all s, t, $|t| \geq k$ with

$$s \in (\mathrm{pre}\, K_f) \cap (\Sigma_n^* f \Sigma_n^*) \text{ and } st \in \mathrm{pre}\, K_f$$

there exists $u \in \mathrm{pre}\, K_n$, $v \in \mathrm{pre}\, t$, $|v| \leq k$ with

$$K_f/sv = K_n/u\,.$$



Post-Fault Recovery: liveness

Def. [K7]: A closed loop $K_f = K_n \dot{\cup} K_d$ is *weakly recovering* if there exists a uniform bound k such that for all s, t, $|t| \geq k$ with

$$s \in (\operatorname{pre} K_f) \cap (\Sigma_n^* f \Sigma_n^*) \text{ and } st \in \operatorname{pre} K_f$$

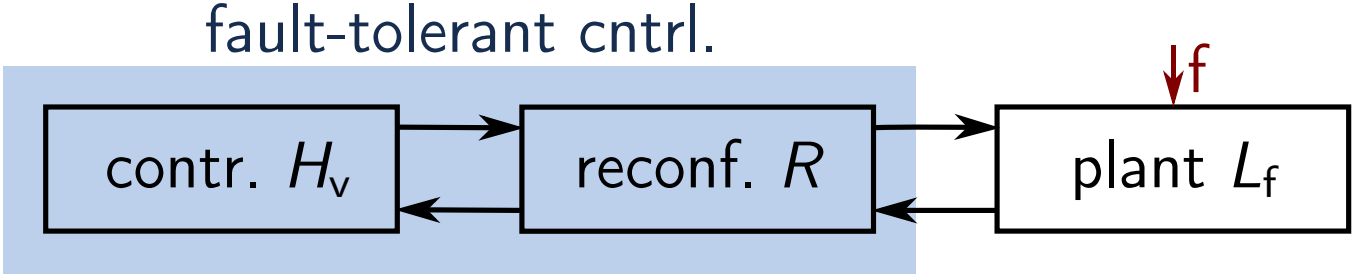
there exists $u \in \operatorname{pre} K_n$, $v \in \operatorname{pre} t$, $|v| \leq k$ with

$$K_f/sv = K_n/u\,.$$

- synthesis problem: given $L_f = L_n \cup L_d$ and E_f, compute an admissible controller H_f such that the closed loop K_f satisfies [K7].
- the property is not retained under union; synthesis procedure exists for $\Sigma_o = \Sigma$

Fault Hiding

Given $L_f = L_n \cup L_d$, $E_f = E_n \cup E_d$, and a solution H_n to (L_n, E_n)



- disconnect nominal controller, i.e., $H_v = h(H_n) \subseteq \Sigma_v^*$ with $\Sigma_v \cap \Sigma_f = \emptyset$, h bijective and applied per event.
- synthesise reconfiguration dynamics $R \subseteq (\Sigma_v \cup \Sigma_o)^*$ to re-connect
- do so by interpreting $H_v \parallel L_f$ as plant and use std. procedures on adapted language inclusion specification, extract R from K
- obtain an overall fault-tolerant controller from H_v and R

- when using a minimal restrictive solution $H_n^{\uparrow}$ resp. $H_v^{\uparrow} \parallel L_f$ for the design, and if the closed loop $K^{\uparrow}$ satisfies [K0]-[K3] and additionally

  [K8] $\quad (\,\forall\, s \in \operatorname{pre} K\,)[\,((\mathrm{p}_v\, s)h(\Sigma_{uc})) \;\cap\; (\operatorname{pre} h(L_n)) \neq \emptyset$
  $\quad\Rightarrow\quad s(\Sigma - h(\Sigma_c))^* h(\Sigma_{uc}) \;\cap\; (\operatorname{pre} K) \neq \emptyset\,]$

  then the corresponding R is admissible to $H_v \parallel L_f$ for any nominal controller H_n that solves (L_n, E_n).

- [K8] is retained under union, synthesis procedures are available.

Note: Nominal controller does not need to be known.

Discrete-Event Systems

Supervisory Control

Naive Fault-Tolerant Control

Active Fault-Tolerant Control

Post-Fault Recovery

Fault-Hiding Approach

Summary

Fault-tolerant supervisory control is addressed by the recent literature in various ways, including passive and active approaches, post-fault recovery and fault-hiding.

Conclusions

- switching is addressed by the common modelling framework — any method for fault-tolerant supervisory control should be interpretable within this framework
- additional features of individual approaches amount to additional closed-loop properties — and novel to synthesis problems
- insisting in uniform bounds for diagnosibility and language convergence may be too strict for particular applications — discussion in terms of ω-languages may turn out beneficial

References

Blanke, M., Kinnaert, M., Lunze, J., Staroswiecki, M., Schröder, J. (2006). Diagnosis and Fault-Tolerant Control. Springer.

Paoli, A. and Lafortune, S. (2005). Safe diagnosability for fault-tolerant supervision of discrete-event systems. Automatica, 41(8), 1335–1347.

Paoli, A., Sartini, M., and Lafortune, S. (2008). A fault tolerant architecture for supervisory control of discrete event systems. Proceedings of the 17th IFAC world congress, 6542–6547.

Paoli, A., Sartini, M., and Lafortune, S. (2011) . Active fault tolerant control of discrete event systems using online diagnostics. Automatica, 47(4), 639–649.

Sülek, A.N. and Schmidt, K.W. (2014). Computation of supervisors for fault-recovery and repair for discrete event systems. WODES, 428–438.

Watanabe, T.Y, Leal, A.B, Cury, J.E.R, Queiroz, M.H.de. (2017) Safe controllability using online prognosis, IFAC WC 2017.

Wen, Q., Kumar, R., and Huang, J. (2014). Framework for optimal fault-tolerant control synthesis: maximize prefault while minimize post-fault behaviorsfor discrete event systems. IEEE Trans. Syst. Man Cybern. Syst., 44, 1056–1066.

Wen, Q., Kumar, R., Huang, J., and Liu, H. (2008). A framework for fault-tolerant control for discrete event systems. IEEE TAC, 53, 1839–1849.

Wittmann, T., Richter, J., and Moor, T. (2013). Fault-hiding control reconfiguration for a class of discrete-event systems. IFAC DCDS.

Wittmann, T., Richter, J., and Moor, T. (2012). Fault-tolerant control of discrete event systems based on fault-accommodating models. SAFEPROCESS, 854–859.

More references given in

Moor, T. (2016). A discussion of fault-tolerant supervisory control in terms of formal languages. Annual Reviews in Control, 159-169.

Schmuck, A.-K., Moor, T., Majumdar, (2020). On the relation between reactive synthesis and supervisory control of non-terminating processes, Discrete Event Dynamic Systems, 30, 81–124.