

Théorie du contrôle par supervision – Introduction

Formation Systèmes à Evénements Discrets

1^{re} édition
Janvier 2024



- 1 Fondements de la théorie du contrôle par supervision
 - Principe
 - Modèles
 - Propriétés utilisées

- 2 Problèmes traités dans l'approche centralisée
 - Problème 1 : existe-t-il un procédé sous contrôle ?
 - Exemple
 - Problème 2 : ce procédé sous contrôle existe-t-il ?

Fondements de la théorie du contrôle par supervision

C'est l'application :

- Sur des Systèmes à Événements Discrets (SED)
Le comportement est décrit par des événements
- Du principe de bouclage
Le modèle du procédé à contrôler est explicite
- Pour la prise de décisions
Quel contrôle choisir ?
- En fonction de l'objectif à atteindre

C'est l'application :

- Sur des Systèmes à Événements Discrets (SED)
Le comportement est décrit par des événements
- Du principe de bouclage
Le modèle du procédé à contrôler est explicite
- Pour la prise de décisions
Quel contrôle choisir ?
- En fonction de l'objectif à atteindre

C'est l'application :

- Sur des Systèmes à Événements Discrets (SED)
Le comportement est décrit par des événements
- Du principe de bouclage
Le modèle du procédé à contrôler est explicite
- Pour la prise de décisions
Quel contrôle choisir ?
- En fonction de l'objectif à atteindre

C'est l'application :

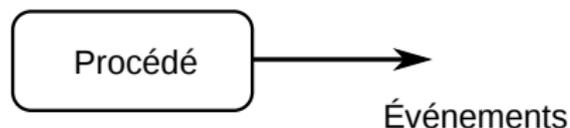
- Sur des Systèmes à Événements Discrets (SED)
Le comportement est décrit par des événements
- Du principe de bouclage
Le modèle du procédé à contrôler est explicite
- Pour la prise de décisions
Quel contrôle choisir ?
- En fonction de l'objectif à atteindre

Objectif à atteindre

L'objectif est de déterminer tous les comportements :

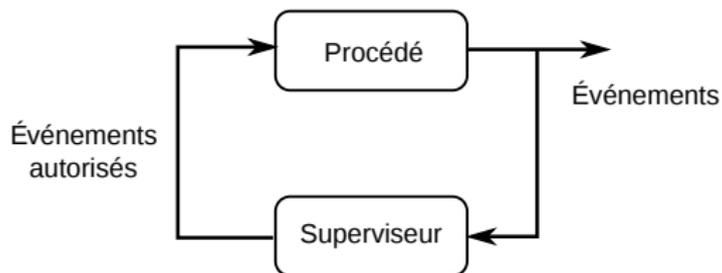
- Contrôlables
- Non bloquants
- Respectant les exigences énoncées

Modèles



- Le procédé est constitué de plusieurs composants
- Chaque composant i est modélisé par un automate à états déterministe G_i
- Le procédé global G est construit par composition des automates des composants G_i

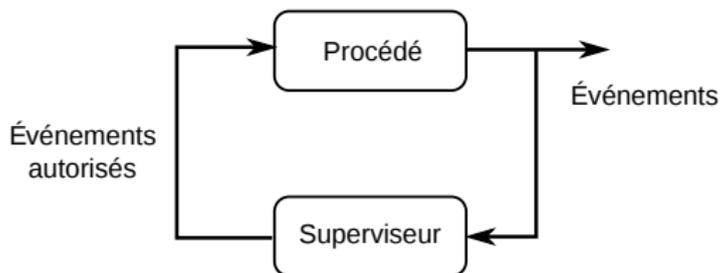
Le procédé sous contrôle : S/G



- Le procédé évolue spontanément
- Le superviseur S restreint les comportements possibles :

$$L(S/G) \subseteq L(G)$$

Comment le superviseur restreint-il le comportement ?



Σ : ensemble des événements générés par le procédé

$$\Sigma = \Sigma_c \cup \Sigma_u \quad \text{et} \quad \Sigma_c \cap \Sigma_u = \emptyset$$

avec :

- Σ_c : ensemble des événements contrôlables c'est-à-dire qui peuvent être interdits par le superviseur
- Σ_u : ensemble des événements incontrôlables

Définition formelle du superviseur

Définition

$S : L(G) \rightarrow 2^\Sigma$ avec 2^Σ ensemble de tous les sous-ensembles de Σ

Remarque : Quel que soit s généré par G , $S(s)$ est l'ensemble des événements autorisés par S . Donc : $\forall s \in L(G) : \Sigma_u \subseteq S(s)$

Langage généré $L(S/G)$

Le langage $L(S/G)$ est défini par :

- 1 $\varepsilon \in L(S/G)$
- 2 $[s \in L(S/G) \text{ et } \sigma \in S(s) \text{ et } s\sigma \in L(G)] \Leftrightarrow s\sigma \in L(S/G)$

Langage marqué $L_m(S/G)$

$$L_m(S/G) = L(S/G) \cap L_m(G)$$

Définition formelle du superviseur

Définition

$S : L(G) \rightarrow 2^\Sigma$ avec 2^Σ ensemble de tous les sous-ensembles de Σ

Remarque : Quel que soit s généré par G , $S(s)$ est l'ensemble des événements autorisés par S . Donc : $\forall s \in L(G) : \Sigma_u \subseteq S(s)$

Langage généré $L(S/G)$

Le langage $L(S/G)$ est défini par :

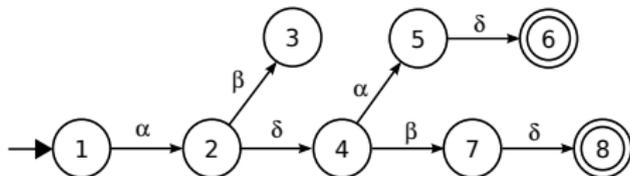
- 1 $\varepsilon \in L(S/G)$
- 2 $[s \in L(S/G) \text{ et } \sigma \in S(s) \text{ et } s\sigma \in L(G)] \Leftrightarrow s\sigma \in L(S/G)$

Langage marqué $L_m(S/G)$

$$L_m(S/G) = L(S/G) \cap L_m(G)$$

Exemple de superviseur

Le procédé G



$$\Sigma = \{\alpha, \beta, \delta\}, \Sigma_c = \{\alpha, \beta\}, \Sigma_u = \{\delta\}$$

Est-il possible d'autoriser tous les comportements mais d'interdire les états 5 et 6 ? oui (ouf!)

Le superviseur

$$S(\varepsilon) = \Sigma \quad S(\alpha) = \Sigma \quad S(\alpha.\delta) = \{\beta, \delta\} \quad S(\alpha.\delta.\beta) = \Sigma$$

Propriétés utilisées

Contrôlabilité des événements

Un événement peut être contrôlable ou pas : $\Sigma = \Sigma_u \cup \Sigma_c$

Contrôlabilité d'un langage

Le langage K est contrôlable par rapport au procédé G et à l'ensemble des événements incontrôlables Σ_u si et seulement si : $\overline{K} \cdot \Sigma_u \cap L(G) \subseteq \overline{K}$

Existence d'un superviseur

Soit $K \subseteq L(G)$ avec $K \neq \emptyset$. Il existe un superviseur S pour G tel que $L(S/G) = K$ si et seulement si K est contrôlable par rapport à G .

Rappel : \overline{K} est le préfixe-clos de K .

Exemple : Si $K = \{abc, de\}$ alors $\overline{K} = \{\varepsilon, a, ab, abc, d, de\}$.

Contrôlabilité des événements

Un événement peut être contrôlable ou pas : $\Sigma = \Sigma_u \cup \Sigma_c$

Contrôlabilité d'un langage

Le langage K est contrôlable par rapport au procédé G et à l'ensemble des événements incontrôlables Σ_u si et seulement si : $\overline{K} \cdot \Sigma_u \cap L(G) \subseteq \overline{K}$

Existence d'un superviseur

Soit $K \subseteq L(G)$ avec $K \neq \emptyset$. Il existe un superviseur S pour G tel que $L(S/G) = K$ si et seulement si K est contrôlable par rapport à G .

Rappel : \overline{K} est le préfixe-clos de K .

Exemple : Si $K = \{abc, de\}$ alors $\overline{K} = \{\varepsilon, a, ab, abc, d, de\}$.

Contrôlabilité des événements

Un événement peut être contrôlable ou pas : $\Sigma = \Sigma_u \cup \Sigma_c$

Contrôlabilité d'un langage

Le langage K est contrôlable par rapport au procédé G et à l'ensemble des événements incontrôlables Σ_u si et seulement si : $\overline{K} \cdot \Sigma_u \cap L(G) \subseteq \overline{K}$

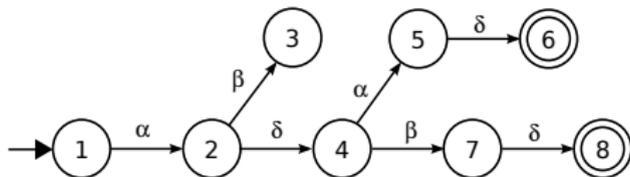
Existence d'un superviseur

Soit $K \subseteq L(G)$ avec $K \neq \emptyset$. Il existe un superviseur S pour G tel que $L(S/G) = K$ si et seulement si K est contrôlable par rapport à G .

Rappel : \overline{K} est le préfixe-clos de K .

Exemple : Si $K = \{abc, de\}$ alors $\overline{K} = \{\varepsilon, a, ab, abc, d, de\}$.

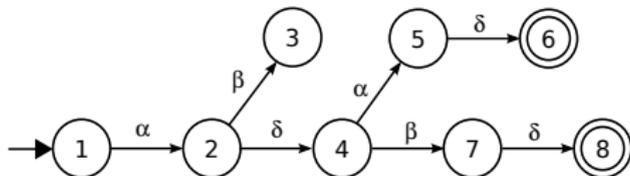
Le procédé G



$$\Sigma = \{\alpha, \beta, \delta\}, \Sigma_c = \{\alpha, \beta\}, \Sigma_u = \{\delta\}$$

Est-il possible d'autoriser tous les comportements mais d'interdire l'état 6 ?

Le procédé G



Procédé bloquant : à partir de l'état 3 impossible d'atteindre 6 ou 8.

Est-il possible d'imposer $K = \{\alpha.\delta.\beta.\delta\}$?

Existence d'un superviseur non bloquant

Soit $K \subseteq L_m(G)$ avec $K \neq \emptyset$. Il existe un superviseur non bloquant S pour G tel que $L_m(S/G) = K$ si et seulement si K est contrôlable par rapport à G .

1 Fondements de la théorie du contrôle par supervision

- Principe
- Modèles
- Propriétés utilisées

2 Problèmes traités dans l'approche centralisée

- Problème 1 : existe-t-il un procédé sous contrôle ?
- Exemple
- Problème 2 : ce procédé sous contrôle existe-t-il ?

Problème 1 : existe-t-il un procédé sous contrôle ?

Problème classique 1

On donne :

- Le procédé non contrôlé
- Des exigences

On cherche :

- À vérifier que les exigences peuvent être respectées
- À déterminer le procédé sous contrôle
- À déterminer le superviseur

Problème 1 : démarche d'étude détaillée (1/2)

On donne :

- Les modèles des composants : automates G_i
- Des exigences : aussi sous forme d'automates E_j

On construit :

- Le modèle G du procédé global par composition des G_i
- Le modèle $G \parallel E$ du procédé respectant les exigences par composition des G_i et E_j

Problème 1 : démarche d'étude détaillée (2/2)

On donne :

- Les modèles des composants : automates G_i
- Des exigences : aussi sous forme d'automates E_j

On cherche : à vérifier que $L_m(G \parallel E)$ est contrôlable par rapport à G et Σ_u :

- Si oui il existe un superviseur
- Sinon il faut trouver le plus grand sous-langage de $L_m(G \parallel E)$ contrôlable (**le suprême contrôlable**). Un algorithme de calcul à partir de $G \parallel E$ permet de trouver le résultat :
 - Si le résultat n'est pas \emptyset , les spécifications peuvent être respectées
 - Sinon... (mais cela n'arrive jamais ! ;-)

Un seul point de vue possible : interdire des événements

A partir de ce point de vue, il est possible de définir **4 types d'exigence** :

- États illégaux : une exigence doit interdire des états particuliers de G
- Partition d'états : une exigence nécessite de mémoriser de quelle façon un état de G a été atteint afin de déterminer quels sont les comportements suivants admissibles
- Alternance d'événements : une exigence nécessite l'alternance des occurrences de 2 événements
- Séquence illégale : une exigence considère comme illégaux tous les mots de $\mathcal{L}(G)$ contenant la séquence $s_f = \sigma_1 \dots \sigma_n \in \Sigma^*$

Un seul point de vue possible : interdire des événements

A partir de ce point de vue, il est possible de définir **4 types d'exigence** :

- États illégaux : une exigence doit interdire des états particuliers de G
- Partition d'états : une exigence nécessite de mémoriser de quelle façon un état de G a été atteint afin de déterminer quels sont les comportements suivants admissibles
- Alternance d'événements : une exigence nécessite l'alternance des occurrences de 2 événements
- Séquence illégale : une exigence considère comme illégaux tous les mots de $\mathcal{L}(G)$ contenant la séquence $s_f = \sigma_1 \dots \sigma_n \in \Sigma^*$

Un seul point de vue possible : interdire des événements

A partir de ce point de vue, il est possible de définir **4 types d'exigence** :

- États illégaux : une exigence doit interdire des états particuliers de G
- Partition d'états : une exigence nécessite de mémoriser de quelle façon un état de G a été atteint afin de déterminer quels sont les comportements suivants admissibles
- Alternance d'événements : une exigence nécessite l'alternance des occurrences de 2 événements
- Séquence illégale : une exigence considère comme illégaux tous les mots de $\mathcal{L}(G)$ contenant la séquence $s_f = \sigma_1 \dots \sigma_n \in \Sigma^*$

Un seul point de vue possible : interdire des événements

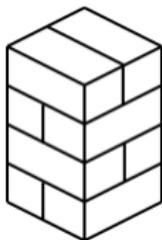
A partir de ce point de vue, il est possible de définir **4 types d'exigence** :

- États illégaux : une exigence doit interdire des états particuliers de G
- Partition d'états : une exigence nécessite de mémoriser de quelle façon un état de G a été atteint afin de déterminer quels sont les comportements suivants admissibles
- Alternance d'événements : une exigence nécessite l'alternance des occurrences de 2 événements
- Séquence illégale : une exigence considère comme illégaux tous les mots de $\mathcal{L}(G)$ contenant la séquence $s_f = \sigma_1 \dots \sigma_n \in \Sigma^*$

Exemple



(a) Machine 1



(b) Stock intermédiaire



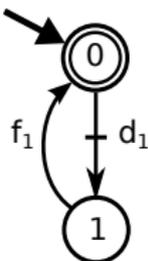
(c) Machine 2

Exigences :

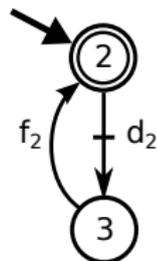
- Le stock intermédiaire ne peut contenir qu'une seule pièce
- La machine 2 ne peut produire que s'il y a une pièce dans le stock

Le procédé non contrôlé

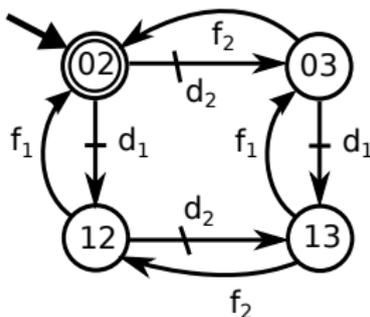
Machine 1 :



Machine 2 :



Le procédé G :

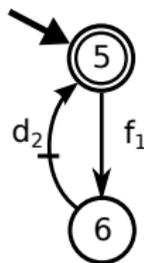


La modélisation des exigences

Exigences :

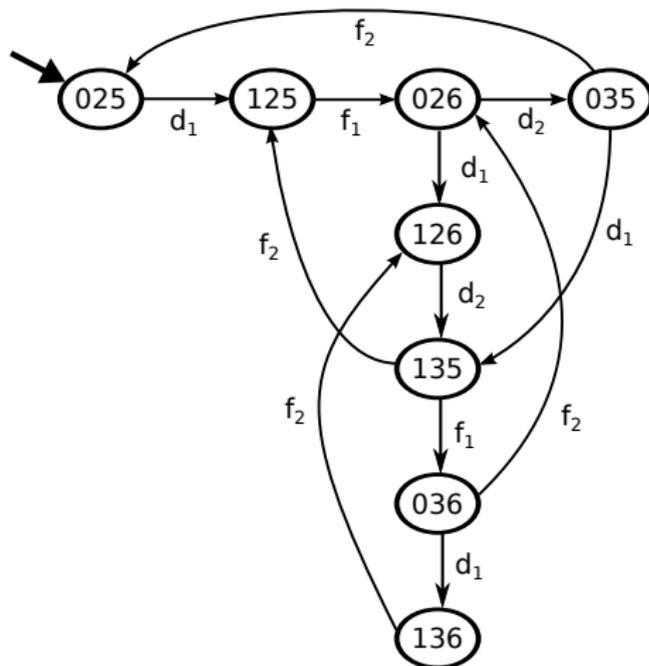
- Le stock intermédiaire ne peut contenir qu'une seule pièce
- La machine 2 ne peut produire que s'il y a une pièce dans le stock

Le modèle E :



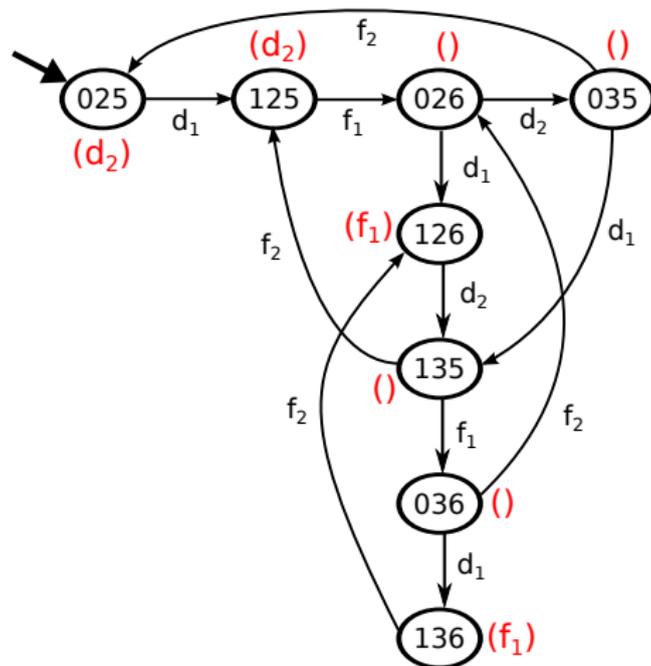
Le procédé respectant les exigences

Le modèle $G \parallel E$:



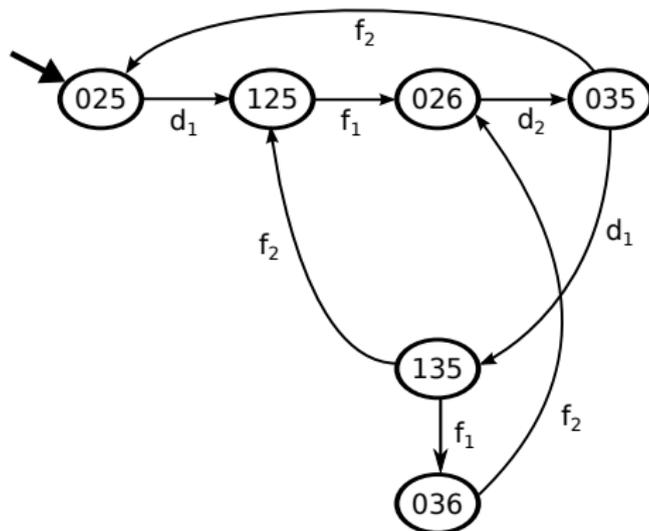
Les événements interdits par les exigences

Le modèle $G \parallel E$ annoté :



Le procédé sous contrôle

Le modèle S/G :



La fonction :

$$S(\varepsilon) = \Sigma - \{d_2\}$$

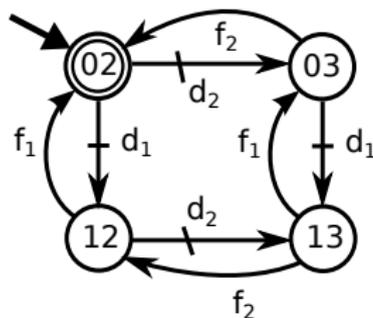
$$S(d_1) = \Sigma - \{d_2\}$$

$$S(d_1.f_1) = \Sigma - \{d_1\}$$

$$S(d_1.f_1.d_2) = \Sigma$$

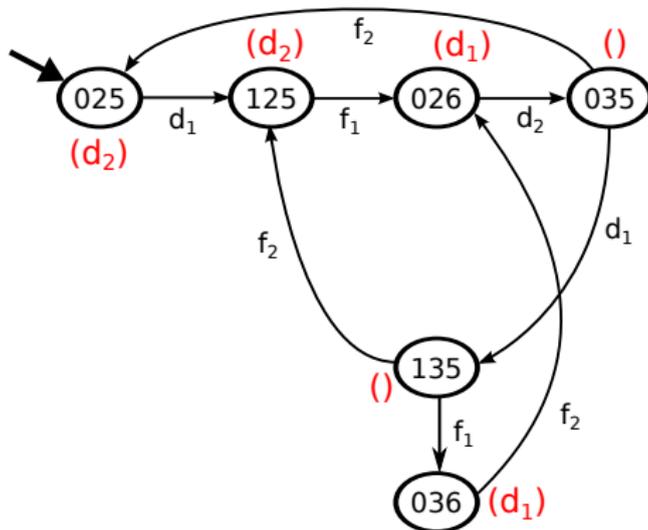
...

Le procédé G :



Une représentation du superviseur

Un automate étiqueté :



1 Fondements de la théorie du contrôle par supervision

2 Problèmes traités dans l'approche centralisée

- Problème 1 : existe-t-il un procédé sous contrôle ?
- Exemple
- Problème 2 : ce procédé sous contrôle existe-t-il ?

On donne :

- Le procédé sous contrôle
- Le procédé non contrôlé

On cherche :

- A vérifier que le procédé sous contrôle est contrôlable
- A déterminer le superviseur

Problème 2 : démarche d'étude

On a :

- Le modèle du procédé G
- Le modèle du procédé sous contrôle H tel que $L(H) \subseteq L(G)$ et $L_m(H) \subseteq L_m(G)$

On cherche : à vérifier que $\mathcal{L}_m(H)$ est contrôlable par rapport à G et Σ_u :

- Si oui il existe un superviseur
- Sinon il faut trouver le plus grand sous-langage de $\mathcal{L}_m(H)$ contrôlable (**le suprême contrôlable**). Un algorithme de calcul à partir de H permet de trouver le résultat :
 - Si le résultat n'est pas \emptyset , les spécifications peuvent être respectées
 - Sinon...

Références bibliographiques

- [1] Benoît CAILLAUD et al., éd. Synthesis and control of discrete event systems. Kluwer Academic Publishers, 2010.
- [2] Christos G CASSANDRAS et Stéphane LAFORTUNE. Introduction to Discrete Event Systems. Springer, 2008.
- [3] W. Murray WONHAM et Kai CAI. « Supervisory control of discrete-event systems ». notes de cours, département of Electrical and Computer Engineering, University of Toronto. 2018. URL : https://www.researchgate.net/publication/319434005_Supervisory_Control_of_Discrete-Event_Systems_v20170901.

Équipe pédagogique

Auteur.rice.s : Pascale Marangé, Alexandre Philippot, Laurent Piétrac

Intervenant : Laurent Piétrac