



**HAL**  
open science

# Surveillance des systèmes de production automatisés : détection et aide au diagnostic

Hassan Rayhane

► **To cite this version:**

Hassan Rayhane. Surveillance des systèmes de production automatisés : détection et aide au diagnostic. Automatique / Robotique. Institut National Polytechnique de Grenoble - INPG, 2004. Français. NNT: . tel-00169988

**HAL Id: tel-00169988**

**<https://theses.hal.science/tel-00169988>**

Submitted on 17 Sep 2007

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

N° attribué par la bibliothèque

--	--	--	--	--	--	--	--	--	--

## THESE EN COTUTELLE

Pour obtenir le grade de

DOCTEUR DE L'INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE  
ET DE L'ECOLE MOHAMMADIA D'INGENIEURS DE RABAT

*Spécialité : Automatique, Productique*

Préparée au laboratoire d'automatique de Grenoble dans le cadre de l'École Doctorale  
«**Electronique, Electrotechnique, Automatique, Télécommunications, Signal**»  
et au laboratoire d'automatique et informatique industrielle (L.A.2.I) de l'E.M.I  
dans le cadre de l'U.F.R « **Automatique** ».

Présentée et soutenue publiquement

par

Monsieur **Hassan RAYHANE**

Le 02 juillet 2004

Titre :

**Surveillance des systèmes de production automatisés :  
Détection et Diagnostic.**

---

**Directeurs de thèse : M. T. BENNANI & Mme Z. SIMEU-ABAZI**

---

### JURY

M.	M. AMGHAR	professeur à l'Univ Med V à Rabat	Rapporteur
M.	Y. FREIN	professeur à l'I.N.P.G à Grenoble	Président
Mme	N. SBITI	professeur à l'E.M.I à Rabat	Rapporteur
M	N. ZERHOUNI	professeur à l'E.N.S.M.M à Besançon	Rapporteur
M.	T. BENNANI	professeur à l'E.M.I à Rabat	Directeur de thèse
Mme	Z. SIMEU-ABAZI	professeur à l'U.J.F à Grenoble	Directeur de thèse

# Table des matières

<b>LISTE DES FIGURES</b> .....	4
<b>INTRODUCTION GÉNÉRALE</b> .....	8
<b>CHAPITRE 1. LES APPROCHES DE SURVEILLANCE</b> .....	12
<b>1. SURVEILLANCE DES SYSTÈMES DE PRODUCTION</b> .....	13
1-1. LES MODES DE FONCTIONNEMENT D’UN SYSTÈME.....	13
1-2. LES FONCTIONS D’UN SYSTÈME DE SURVEILLANCE – TERMINOLOGIE .....	14
1-3. LES MÉTHODES DE SURVEILLANCE.....	16
1.3-1. Les méthodes de surveillance par modélisation fonctionnelle et matérielle .....	17
1.3-2. Les méthodes de surveillance par modélisation physique.....	17
1.3-3. Les méthodes de surveillance par analyse des signatures externes.....	17
<b>2. SYNTHÈSE DES APPROCHES DE SURVEILLANCE</b> .....	17
2-1. L’APPROCHE FILTRE .....	18
2.1-1. Approche filtre au CRAN / LACN à Nancy .....	18
2.1-2. Approche filtre au LAIL de Lille .....	19
2-2. L’APPROCHE COMPARATEUR .....	21
2.2-1. Approche de l’université de Carnegie Mellan.....	21
2.2-2. Modèle émulateur de LAI de Lyon.....	22
2-3. APPROCHE HIÉRARCHIQUE PAR MODÈLE DE RÉFÉRENCE.....	23
2-4. SYNTHÈSE- CONCLUSION .....	25
<b>3. PROPOSITION D’UNE DEMARCHE GÉNÉRALE</b> .....	25
3-1. DESCRIPTION DES DIFFÉRENTS MODULES DE LA STRUCTURE DE SURVEILLANCE.....	26
3-2. PRINCIPE DE DÉTECTION .....	28
<b>CHAPITRE 2. MODÉLISATION DU SYSTÈME DE SURVEILLANCE</b> .....	30
<b>1. SYSTÈMES À ÉVÉNEMENTS DISCRETS (S.E.D): RAPPEL</b> .....	31
<b>2. MODÉLISATION PAR RÉSEAUX DE PETRI</b> .....	32
2-1. LES MÉTHODES DE SURVEILLANCE .....	32
2-2. MODÉLISATION D’UN SYSTÈME DE PRODUCTION .....	32
2.2-1. Caractéristiques et modélisation d’une machine.....	33
2.2-2. Procédure de détection des défaillances.....	33

---

2-3. CHOIX DES GRANDEURS À SURVEILLER.....	34
2.3-1. SURVEILLANCE BASÉE SUR LE NIVEAU DES STOCKS.....	34
2.3-2. Surveillance basée sur le temps de service .....	35
2.3-3. Optimisation du temps détection.....	36
2.3-4. Résultats de simulations.....	38
2-4. CONCLUSION .....	40
<b>3. MODÉLISATION PAR AUTOMATES TEMPORISÉS .....</b>	<b>40</b>
3-1. INTRODUCTION.....	40
3-2. CONCEPTS DE BASE .....	41
3-3. DÉFINITIONS.....	42
3-4. MODÈLES DE SURVEILLANCE .....	44
3-5. LES MODES DE FONCTIONNEMENT PRIS EN COMPTE .....	45
3.5-1. Critères d'évaluation .....	45
3.5-2. Construction du modèle dynamique .....	46
3.5-3. Conclusion .....	48
<b>4. MODÉLISATION PAR STATEFLOW.....</b>	<b>49</b>
4-1. GÉNÉRALITÉS.....	49
4.1-1. La terminologie : une conception hiérarchisée .....	49
4.1-2. Présentation d'un exemple "illustratif".....	50
<b>CHAPITRE 3. MODÈLE DE SURVEILLANCE.....</b>	<b>52</b>
<b>1. DÉMARCHE GÉNÉRALE DE MISE EN ŒUVRE D'UN SYSTÈME DE SURVEILLANCE.....</b>	<b>53</b>
1-1. LA DEMARCHE : MISE AU POINT DES MODÈLES .....	53
3-2. PRISE EN COMPTE DES TROIS MODES DE FONCTIONNEMENT .....	54
1-3. MISE EN ŒUVRE DU MODÈLE DE SURVEILLANCE .....	55
<b>2. SURVEILLANCE DU SYSTÈME : REMPLISSAGE DES BACS.....</b>	<b>56</b>
2-1. CONSTRUCTION DU MODÈLE DYNAMIQUE DU SYSTÈME .....	58
2-2. MODÈLE DE SURVEILLANCE .....	59
2-3. CONCLUSION .....	62
<b>3. SURVEILLANCE DU SYSTÈME : ATELIER DE TRAITEMENT DE SURFACE.....</b>	<b>62</b>
3-1. MODÉLISATION DES OPÉRATIONS DE TRANSPORT.....	64
3-2. MODÉLISATION DES OPÉRATIONS DE TRAITEMENT.....	65
3-3. CONSTRUCTION DU MODÈLE DE SURVEILLANCE PAR STATEFLOW .....	66

---

<b>CHAPITRE 4. OPTIMISATION DES TEMPS DE DÉTECTION</b> .....	68
<b>1. DÉMARCHE D’OPTIMISATION DES TEMPS DE DÉTECTION</b> .....	69
1-1. DÉFINITIONS DES INTERVALLES DE FONCTIONNEMENT .....	69
1-2. ALGORITHMES D’OPTIMISATION DES TEMPS DE DÉTECTION .....	70
<b>2. ÉVALUATIONS DES PERFORMANCES</b> .....	73
<b>2. PRÉSENTATION DU LOGICIEL STATEFLOW</b> .....	76
2-1. FONCTIONNEMENT D’UN DIAGRAMME STATEFLOW .....	76
2-2. CONSTRUCTION D’UN DIAGRAMME .....	77
<b>3. SIMULATIONS ET ANALYSES</b> .....	78
3-1. RÉSULTATS DE SIMULATION : APPLICATION DE REMPLISSAGE DES BACS .....	79
3.1-1. Calcul de la valeur du seuil $\Delta_{opt}$ .....	80
3.1-2 Détermination du temps de réponse.....	82
3-2. RÉSULTATS DE SIMULATION : APPLICATION DE TRAITEMENT DE SURFACE .....	83
3.2-1. Calcul de la valeur du seuil $\Delta_{opt}$ .....	83
3.2-2. Détermination du temps de réponse.....	86
<b>CHAPITRE 5. APPLICATION AUX SYSTÈMES DE PRODUCTION</b> .....	87
<b>1. SYSTÈME DE PRODUCTION</b> .....	88
1-1. DÉFINITIONS.....	88
1-2. LES STRUCTURES DES SYSTÈMES DE PRODUCTION .....	89
<b>2. MODÈLE GÉNÉRIQUE : MACHINE-STOCK</b> .....	91
2-1. PRÉSENTATION DU MODÈLE DE SURVEILLANCE .....	92
2-2. SIMULATION : RÉSULTATS ET ANALYSES.....	93
2.2-1. Détermination de seuil de tolérance.....	93
2.2-3. Evaluation des performances de la détection.....	95
<b>CONCLUSION GÉNÉRALE</b> .....	97
<b>RÉFÉRENCES BIBLIOGRAPHIQUES</b> .....	100

# *Liste des figures*

Figure 1.	Structure globale d'un système automatisé	12
Figure 2.	Classification des modes de fonctionnement	13
Figure 3.	Principe de l'approche filtre	17
Figure 4.	Modèle de surveillance :Approche filtre au CRAN	18
Figure 5.	Modèle de surveillance :Approche filtre du LAIL	19
Figure 6.	Principe de détection dans l'approche comparateur	20
Figure 7.	Approche comparateur de l'université Carnegie Mellan	21
Figure 8.	Modèle de surveillance orienté sécurité opérationnelle du LAI	22
Figure 9.	Modèle de surveillance-commande avec modèle de référence au LAAS	23
Figure 10.	La structure globale proposée	25
Figure 11.	Identification des entrées/sorties du module de surveillance	26
Figure12.	Identification des entrées/sorties du module de maintenance	26
Figure 13.	Identification des entrées/sorties du module de supervision	27
Figure 14.	Principe de la détection	27
Figure 15.	Evolution de l'état de la machine : "distributeur de boisson"	30
Figure 16.	Exemple d'un système de production	32
Figure 17-a.	Modélisation du fonctionnement d'une machine	33
Figure 17-b.	Modélisation du stock	33
Figure 18.	Modèle générique de la machine	33
Figure19.	Principe de surveillance basée sur le niveau de stock	35
Figure 20.	Modèle de l'approche de surveillance basée sur le temps de service	36
Figure 21.	Modèle réseau de Petri d'une ligne série	36
Figure 22.	Modélisation du système étudié	37
Figure 23.	Modèle de simulation pour l'optimisation du temps de détection	37
Figure 24.	Quelques résultats de simulation	39
Figure 25.	Automate temporisé qui modélise le distributeur de boisson	41
Figure 26.	Structure proposée du module de surveillance	44
Figure 27.	Durées d'activations et modes de fonctionnement	46
Figure 28.	Les modes de fonctionnement utilisés	47

---

Figure 29.	Comparaison indicative entre les structures : automate et statecharts	48
Figure 30.a.	Diagramme "Machine" sous stateflow du chauffe-eau	51
Figure 30.b.	Automate temporisé : modèle de fonctionnement du chauffe-eau	51
Figure 31.	Les étapes de la démarche de construction du modèle de surveillance	53
Figure 32.	Passage entre les trois modes de fonctionnement	55
Figure 33.	Modèle global de surveillance	56
Figure 34.	Exemple d'application : Remplissage des bacs	56
Figure 35.	Graficet de commande du système de remplissage des bacs	57
Figure 36.	Graphe des situations	57
Figure 37.	Modèle dynamique du système de remplissage des bacs	59
Figure 38.	Le modèle de surveillance de la station de remplissage des bacs	60
Figure 39.	Atelier de traitement de surface à surveiller	62
Figure 40.	Diagramme de GANNT : ordonnancement optimal à surveiller	63
Figure 41.	Modèle automate temporisé pour les opérations de transport	64
Figure 42.	Modèle automate temporisé pour les opérations de traitement	65
Figure 43.	Modèle global de surveillance de l'atelier de traitement de surface	66
Figure 44.	Classification de la surveillance selon la valeur du taux de couverture	73
Figure 45.	Algorithme d'optimisation des temps de détection des défaillances	74
Figure 46.	Exemple de diagramme 'statecharts' représentant un modèle de surveillance	77
Figure 47.	Modèle de surveillance de la station de remplissage des bacs	78
Figure 48.	Résultats de simulation : nombre d'alarmes en fonction du retard $\Delta$	79
Figure 49.	Détermination du seuil de tolérance : remplissage bac 2	80
Figure 50.	Classification de la surveillance en fonction du seuil de tolérance	80
Figure 51.	Détermination des temps de réponse du système de surveillance	81
Figure 52.	Modèle de surveillance de l'atelier de traitement de surface	82
Figure 53.	Résultats de simulation : nombre d'alarmes en fonction du retard $\Delta$	83
Figure 54.	Détermination du seuil de tolérance : déplacement du robot	84
Figure 55.	Classification de la surveillance en fonction du seuil de tolérance	84
Figure 56.	Détermination des temps de réponse du système de surveillance	85
Figure 57.	Système de production de type série	89
Figure 58.	Système de production de type assemblage	89

---

Figure 59.	Système de production de type désassemblage	90
Figure 60.	Structure d'analyse d'un système de production de type série	90
Figure 61.	Modèle de surveillance générique pour la durée d'une tâche	92
Figure 62.	Résultats de simulation : cas d'une ligne de production série	93
Figure 63.	Représentation graphique : nombre d'erreurs en fonction du seuil de tolérance	94
Figure 64.	Résultats de simulation : cas du seuil fixe et temps de service variable	95
Figure 65.	Relation entre taux de couve et seuil de tolérance	95



# INTRODUCTION GÉNÉRALE

Pendant les dernières décennies, grâce aux progrès technologiques liés principalement à la rapidité de traitement des données et les grandes capacités de stockage de l'information, les industries, toutes catégories confondues, ont considérablement évolué grâce aux moyens de haute technologie appliquée aux domaines de la gestion de production, de supervision et bien d'autres. L'automatisation à outrance et l'informatisation ont profondément modifié le cadre de la production manufacturière qui tend à remplacer l'homme dans toutes ses tâches, d'une part, et d'autre part, la forte compétitivité et la globalisation du marché « mondialisation » ont conduit à la réalisation de systèmes de production plus hétérogènes et plus complexes qu'auparavant. La conséquence la plus visible est l'intérêt de plus en plus important que portent ces entreprises aux problèmes d'amélioration de performances en termes de productivité mais aussi de fiabilité, disponibilité et sécurité.

Pour garantir ces performances, il devient nécessaire d'adjoindre à chaque système, des outils de surveillance plus adaptés. La conception et l'exploitation des systèmes à fonctionnement sûr, reposent sur l'utilisation de moyens en méthodes et outils permettant de faire face aux défaillances, dès leurs apparitions et avant d'entraîner des dégâts ou dommages, voire des accidents graves. Cependant, même si les systèmes de surveillance déterministes implantés dans ce type d'industrie ont été longtemps considérés comme solution idéale aux problèmes posés, il n'en demeure pas moins que les industriels sont incapables de réagir lorsqu'une situation inconnue se présente à eux. Dans la pratique, la maîtrise des risques dépend essentiellement de la connaissance empirique ou expérimentale acquise sur ces systèmes. Elle repose sur une analyse a posteriori des événements anormaux ou accidentels.

Tout système de production est composé de la partie opérative dans laquelle l'ensemble des opérations sont réalisées, la partie commande qui donne les ordres, conformément au cahier des charges représentant le bon fonctionnement du système, et l'opérateur qui effectue les opérations manuelles. Les évolutions parfois catastrophiques de certains systèmes automatisés trouvent leurs origines dans les trois classes suivantes de dysfonctionnement [SAH-95, VAL-99] :

- Classe 1 : Défaillance d'un composant du procédé commandé.
- Classe 2 : Défaillance du système de commande.
- Classe 3 : Défaillance due à une faute d'un opérateur humain.

A titre d'exemple, l'explosion, en 1986 de la navette spatiale 'challenger' fut une **défaillance de classe 1**. Cette défaillance est due à une fuite sur l'un des moteurs d'appoint utilisés pendant les deux premières minutes du vol. A posteriori, le dépouillement des données recueillies par télémétrie a mis en évidence que les automatismes de régulation embarqués ont très tôt détecté une forte baisse de poussée du moteur incriminé et l'ont aussitôt compensé en manœuvrant les différents gouvernes de l'engin et en augmentant la poussée des moteurs principaux. Les systèmes d'alarme et de sécurité n'ont pas fonctionné ou plutôt n'étaient pas prévus pour faire face à ce type de problème dont l'occurrence n'avait pas été prévue.

Un exemple de **défaillance de classe 2** concerne le terminal central de la SNCF. En effet, dans les premiers mois qui ont suivi la mise en place par la SNCF du système 'SOCRATE', tous les points de vente étaient paralysés pendant quelques minutes, voire quelques heures, ce problème est dû au dysfonctionnement de l'ordinateur du système central et à la saturation des réseaux de communication avec les terminaux de vente. Pourtant dans bien des cas, par exemple pour la délivrance d'un billet sur une ligne régionale, l'opération pouvait être effectuée sans l'ordinateur central. A l'évidence, ce défaut était lié à l'absence totale d'autonomie des terminaux de vente.

Concernant les **défaillances de classe 3** les exemples sont nombreux. L'explosion de la centrale nucléaire de Tchernobyl est la conséquence d'une mauvaise manœuvre des opérateurs humains pendant une phase très délicate de recuit des barres de contrôle du réacteur. L'interprétation erronée de mesures de température aurait conduit à l'emballement du réacteur provoquant une surchauffe suivie de l'explosion du cœur de réacteur.

Le besoin d'automatiser tout en assurant le bon fonctionnement, jusqu'alors domaine réservé à l'homme, s'est fait ressentir et devient primordial. Généralement, afin d'assurer un fonctionnement sûr, les actions de sécurité se traduisent par des interdictions ou des règles de conduite stricte, ce qui engendre une limitation de la productivité. Ainsi, la sécurité est bien souvent opposée à la productivité. La fonction de surveillance que nous développons, s'inscrit dans un processus global de supervision d'un système en cours de fonctionnement. Elle vise à partir des informations disponibles sur ses modes de fonctionnement, à détecter, à localiser et à diagnostiquer les défaillances qui peuvent affecter ses performances et sa sûreté de fonctionnement. La fonction surveillance intervient particulièrement dans l'amélioration du facteur disponibilité du système. En effet, la détection et la localisation rapide des pannes minimisent le temps moyen d'indisponibilité du système [ZWI-95], [DUB-90]. Plusieurs approches de surveillance ont été développées dans différents laboratoires qui travaillent dans le domaine de la productique.

La plupart des approches existantes placent des filtres, entre la partie commande et la partie opérative ; ceci pour vérifier la cohérence de l'action avec l'état réel du procédé et/ou des filtres pour comparer les états réels capteurs par rapport à ceux donnés par le modèle de référence du procédé. Toutes ces approches traitent 'la surveillance' en fonction de 5 éléments majeurs : *la détection* des anomalies, *le diagnostic* du symptôme pour en trouver les causes, *la décision* pour l'élaboration d'une solution qui permet le retour au fonctionnement normal, *la reprise* pour appliquer cette solution et enfin *l'urgence* qui est dédiée à l'application des traitements 'brutaux', rapides et prédéfinis sur le procédé [NOU-97].

L'objectif principal de nos travaux de recherche consiste à concevoir des systèmes de surveillance ; capables de détecter, localiser et diagnostiquer toute défaillance, susceptible d'affecter la production et ceci le plutôt possible. Une marge de tolérance possible dans la durée d'exécution d'une tâche sera calculée par rapport à un intervalle dit de bon fonctionnement du système de production commandé, afin de contribuer au fonctionnement sûr des systèmes automatisés de production.

Le besoin de surveiller s'impose dans le contexte actuel des ateliers de production devenus de plus en plus complexes. L'ensemble de nos travaux concerne la surveillance ; à savoir les fonctions : Détection, Localisation et Diagnostic. Dans le premier chapitre de ce mémoire, nous trouvons un bref rappel sur la terminologie, suivi d'une présentation des différentes approches de surveillance existantes. Une étude est faite pour montrer les limites de chacune des méthodes de surveillance appliquées aux systèmes de production. Cette partie se termine par la proposition d'une structure globale du système de surveillance, supervision et maintenance. Cette structure est composée de différents modules, communiquant entre eux, à intégrer dans un **système de production**.

Le deuxième chapitre commence par un bref rappel sur les systèmes à événements discrets pour ensuite présenter les outils de modélisation utilisés : l'outil réseaux de Petri, l'outil automate temporisé et l'outil statecharts. Sachant que nous avons choisi de surveiller les durées d'exécution des tâches, ces outils permettent une représentation de la dynamique du procédé pour détecter tout retard dans l'exécution ou la non-réalisation d'une tâche. Dans l'objectif de surveiller le système de production, il est nécessaire de définir des critères de choix des grandeurs à surveiller. Nous présentons aussi dans cette partie les conditions d'évaluation et la prise en compte des trois modes de fonctionnement : mode **normal**, mode **dégradé** et le mode **de défaillance** critique, nécessaire pour la mise en œuvre du système de surveillance.

Dans le troisième chapitre, nous présentons en détail la démarche à suivre pour la mise en oeuvre d'un modèle de surveillance. A partir du modèle de commande que nous supposons existant et sans faute (généralement sous la forme d'un grafcet ou d'un ordonnancement pour chaque activité qui composent le système de production à surveiller), nous construisons d'abord les modèles dynamiques correspondant à chaque activité, en tenant compte des trois modes de fonctionnement possibles et des conditions de passage d'un mode à l'autre. L'outil utilisé pour les modèles dynamiques est les automates temporisés, nous verrons aussi que cet outil présente l'inconvénient de l'explosion du nombre d'états et d'arcs dans le modèle de surveillance. D'où l'utilisation de l'outil stateflow, qui est mieux adapté pour les objectifs de surveillance par sa grande lisibilité lors de l'évolution du système. De plus, la hiérarchisation structurelle permet de réduire le nombre d'arcs et d'états. Le principe étant le suivi de la bonne exécution des différentes tâches du système en surveillant les durées d'exécution des tâches. Cette démarche sera appliquée à un premier exemple, celui d'une station de remplissage des bacs. Une deuxième application, qui illustre notre démarche, est celle correspondant à la surveillance d'un atelier de traitement de surface. Elle a la particularité d'avoir plusieurs activités qui s'exécutent en parallèle.

L'objectif étant de proposer une démarche de surveillance ; permettant d'augmenter la disponibilité par une détection rapide des défaillances. Le paramètre qui peut contribuer à diminuer l'indisponibilité d'un équipement est le temps de détection. Le quatrième chapitre présente la démarche d'optimisation de ce temps de détection. Nous présentons l'algorithme d'optimisation illustré sur les applications citées. Une présentation du logiciel sera faite, dans l'objectif de faciliter la lecture et l'exploitation des diagrammes de simulation. Après simulation, nous déduisons les conditions de choix de l'intervalle dit de tolérance. Ensuite une synthèse sera faite sur les résultats de simulation obtenus.

Enfin, dans le cinquième chapitre, nous appliquons dans un premier temps l'approche de surveillance mise en oeuvre dans le cadre de notre travail, et ensuite l'algorithme d'optimisation des temps de détection des défaillances à un système de production manufacturier de type série. Enfin une évaluation des performances en introduisant le taux de couverture d'un seuil de tolérance. Des résultats de simulations obtenus confirment son intérêt.

Nous terminons ce mémoire par une conclusion, dans laquelle nous montrons l'apport de ce travail dans le domaine de la surveillance des systèmes de production. Quelques perspectives seront présentées comme continuation à ce travail pour compléter le modèle global de surveillance et de supervision.

## CHAPITRE 1. LES APPROCHES DE SURVEILLANCE

*Même si les systèmes de surveillance considérés comme solution idéale, sont bien implantés dans les industries à risques, leurs champs d'action restent limités aux défaillances prévisibles. Le caractère déterministe des méthodes appliquées ne permet pas de prendre en compte tous les cas de dysfonctionnement qui peuvent se présenter. En effet, les systèmes de surveillance déterministes sont incapables de réagir face à une situation inconnue.*

*Dans ce chapitre, après avoir défini les besoins de surveillance pour les systèmes de production, nous présenterons un tour d'horizon sur les approches de surveillance existantes. Nous terminerons ce chapitre par la proposition d'une nouvelle structure de surveillance, supervision et maintenance capable de réagir à différentes situations sans la connaissance préalable des modes de défaillance du système.*

## 1. SURVEILLANCE DES SYSTEMES DE PRODUCTION

Tout système de production est composé d'une partie opérative et d'une partie commande. Pour une séquence de tâches à réaliser, la partie commande donne l'ordre d'exécution à la partie opérative. Dès que la tâche est effectuée, celle-ci informe la partie commande de la réalisation de chaque tâche. Ces échanges représentés sur la figure 1, sont mis en évidence par les ordres (actions) et les comptes-rendus (capteurs).

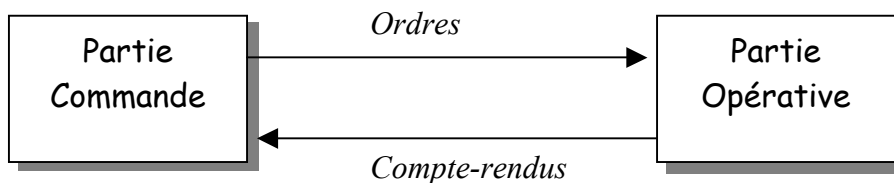


Fig. 1. Structure globale d'un système automatisé

Pour un séquençage de tâches données, l'exécution correcte de toutes les opérations permet d'identifier le comportement attendu du système de production. La surveillance de ce type de système a pour but de traiter tous les comportements qui s'écartent du comportement attendu. Le mécanisme de base utilisé pour la détection consiste à comparer en temps réel les évolutions du système observé avec celle du modèle préalablement défini [SAH-95, VAL-99].

### 1-1. Les modes de fonctionnement d'un système

Un système est par définition un ensemble susceptible d'évoluer dans le temps. Il est composé d'un ensemble d'éléments interagissant entre eux et avec l'extérieur, réunis pour assurer une mission [DUB- 90]. Suivant le mode de fonctionnement sous lequel le système évolue, la mission pour laquelle le système a été conçu peut être totalement remplie, partiellement remplie ou non remplie. Les modes de fonctionnement sont de différents types comme le montre la figure 2 :

- Les modes de fonctionnement « **normaux** » : Ils regroupent le mode de fonctionnement nominal qui doit être, en principe, en parfaite adéquation avec la qualité de la mission.
- Les modes de fonctionnement « **anormaux** » : Lorsque le système est dans ce mode, la mission peut être partiellement remplie ou non remplie ; en tout état de cause, le service n'est pas satisfaisant. Ces modes peuvent être décomposés en :

- Modes interdits : ce sont des modes sous lesquels le système ne doit absolument pas fonctionner pour des raisons de sécurité.
  - Modes défaillants : ils correspondent à de mauvais fonctionnement du système. Un sous-ensemble de ces modes est constitué des modes de pannes où un sous-ensemble du système voire le système complet ne fonctionne plus.
  - Modes dégradés : ils correspondent à l'accomplissement de la mission soit de façon partielle, soit avec des performances moindres.
  - Modes critiques : ce sont des modes pour lesquels le système présente des caractéristiques de fonctionnement très particulières et souvent non souhaitées.
- Les modes de fonctionnement « **d'exception** » : ce sont des modes qui peuvent être normaux ou anormaux mais qui ont la caractéristique d'être peu tolérés ou peu fréquents.

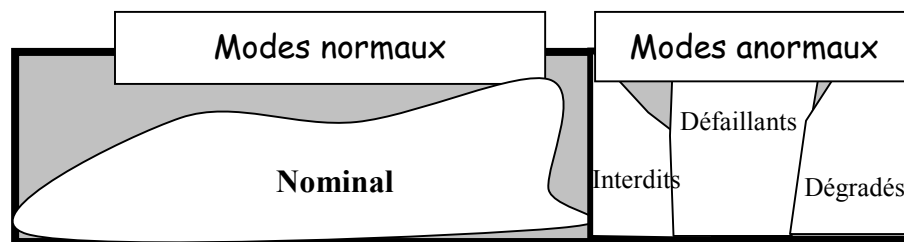


Fig. 2. *Classification des modes de fonctionnement.*

Bien évidemment tous les systèmes ne présentent pas tous ces modes de fonctionnement. Cette classification permet de couvrir pratiquement tous les modes susceptibles de se produire et de s'adapter à la variété des systèmes rencontrés.

Dans la structure du système de surveillance proposé, on s'intéresse plus particulièrement au fonctionnement nominal. Pour les modes de fonctionnement anormaux on étudiera de près le fonctionnement en mode dégradé et en mode défaillant.

## 1-2. Les fonctions d'un système de surveillance – Terminologie

Le rôle d'un système de surveillance est de connaître à chaque instant l'état du procédé, fournir des données validées au système de conduite, améliorer la disponibilité et la sécurité du procédé. Différentes définitions ont été données concernant la surveillance on peut citer la définition suivante [ZWI-95]:

*Définition.1* : La **surveillance** représente l'ensemble des moyens mis en œuvre (opérations manuelles ou automatiques, étapes, fonctions et mécanismes) destinées à observer l'état d'une entité (en ligne, en temps réel) dans le but de faire face aux aléas d'un système au cours de la phase d'exploitation. □

Le suivi en temps réel des signaux permet de reconstituer l'état réel du système commandé en tenant compte de ces différents modes de fonctionnement et de faire toutes les inférences nécessaires pour produire des données utilisées.

L'exploitation de ces données permet par la suite de :

- Dresser des historiques de fonctionnement.
- Le cas échéant, mettre en œuvre un processus de traitement de défaillance.

Les performances d'un système de surveillance peuvent être améliorées grâce aux actions sur les différentes fonctions du système de surveillance à savoir la fonction détection et la fonction diagnostic qui consiste à la localisation, et l'identification de la défaillance.

*Définition.2* : La **détection** consiste à comparer la signature courante à la signature de référence associée aux modes de fonctionnement identifiés et ensuite à prendre une décision en fonction du résultat de la comparaison. □

Dans les procédures de détections, les signatures utilisées sont des grandeurs scalaires, des courbes ou des images. Sachant que le signal d'écart possède un comportement aléatoire, la prise de décision nécessite la définition de seuils aux maxima et aux minima au-delà desquels on déclarera un dysfonctionnement. Les signatures étant toujours bruitées, il devient nécessaire de faire appels aux tests statistiques de décision pour évaluer la prise de risque. En effet, si les valeurs des seuils sont trop basses on observera souvent des fausses alarmes. Par contre si les seuils sont élevés, on évite des fausses alarmes, mais les risques de non-détection d'un dysfonctionnement sont importants.

Cette fonction permet alors de caractériser le fonctionnement du système de normal ou d'anormal. On peut distinguer deux grandes classes d'anomalies :

- La première regroupe les situations pour lesquelles le comportement du système devient anormal par rapport à ses caractéristiques intrinsèques.
- La seconde regroupe les situations dans lesquelles le comportement est anormal par rapport à la loi de commande appliquée.

Les procédures de localisation et d'identification correspondent au diagnostic. Elles nécessitent une connaissance des pannes pouvant apparaître sur le procédé : le type de pannes et leurs symptômes.



*Définition.3* : La **localisation** consiste à déterminer l'endroit du procédé où s'est produite la défaillance et la nature de la défaillance.



Le **Diagnostic** consiste à localiser les éléments défaillants et à identifier les causes à l'origine du problème; ceci en établissant un lien causal entre les symptômes et les éléments fautifs à remplacer. La phase qui suit correspond à la décision. Elle a pour rôle de déterminer et d'engager les actions permettant de ramener au mieux le système dans un état normal. Ces actions peuvent être des ordres d'arrêt d'urgence ou des lancements de réparations ou d'opérations préventives. Dans le cas où on voudrait éviter une perte de production, cette décision peut être une reconfiguration du fonctionnement du procédé.

Différents outils, comme par exemple, le grafcet ou les automates sont utilisés pour la commande des systèmes à événements discrets. Les questions que l'on se pose sont les suivantes:

- Peut-on exploiter ces mêmes modèles pour la surveillance des systèmes?
- Est-il nécessaire d'élaborer de nouveaux modèles pour la détection et la localisation des défaillances ?

A travers l'abondante littérature sur la surveillance des systèmes, différents modèles de surveillance ont été développés. Parmi les solutions proposées, certaines exploitent directement le modèle de commande associée à l'application. Après une brève description du principe de détection une synthèse de ces travaux est présentée.

### **1-3. Les méthodes de surveillance**

Les méthodes de surveillance des défaillances utilisées dans les différents secteurs industriels sont très variées. Leur principe général consiste à confronter les données relevées au cours du fonctionnement réel du système avec la connaissance que l'on a de son fonctionnement nominal ou de ses fonctionnements défaillants. La forme sous laquelle se présente la connaissance sur le système conditionne, les différentes méthodes utilisées en surveillance.

Selon l'ouvrage [ZWI-95], ces dernières peuvent être classées en trois grandes familles:

- Les Méthodes de surveillance par modélisation fonctionnelle et matérielle
- Les Méthodes de surveillance par modélisation physique
- Les Méthodes de surveillance par analyse des signatures externes

### **1.3-1. Les méthodes de surveillance par modélisation fonctionnelle et matérielle**

Ces méthodes sont basées sur la modélisation du comportement des fonctions et matériels du système. Le principe de ces méthodes est d'établir des liens entre les causes initiales des défaillances et leurs effets. Parmi ces méthodes on cite l'arbre de défaillances. Elle permet d'identifier toutes les combinaisons qui mènent à l'apparition d'un événement unique indésirable. Les événements sont représentés sous formes d'arbre utilisant des fonctions logiques.

### **1.3-2. Les méthodes de surveillance par modélisation physique**

Ces méthodes sont fondées sur la connaissance d'un modèle mathématique représentant le comportement dynamique d'un système continu. On peut citer la méthode du modèle. Cette méthode consiste à comparer les grandeurs déduites d'un modèle représentatif du fonctionnement des différentes entités du processus avec les mesures directement observées sur le processus industriel. La présence d'un écart fournira l'indication de la présence d'une défaillance. La profondeur du diagnostic fourni par cette méthode dépendra du degré de la finesse de la modélisation retenue.

### **1.3-3. Les méthodes de surveillance par analyse des signatures externes**

Ces méthodes s'appliquent dans les situations où la modélisation des mécanismes reliant les causes des défaillances et des dégradations n'est pas techniquement modélisables. Elles sont basées sur le retour d'expérience. Parmi ces méthodes, nous citons la reconnaissance des formes. Elle consiste à déterminer le degré de similarité entre des formes (ensemble de symptômes mis sous la forme d'un vecteur) observées sur le système et des formes obtenues préalablement par apprentissage.

## **2. SYNTHÈSE DES APPROCHES DE SURVEILLANCE**

Il existe trois approches pour implémenter un système de surveillance. Elles diffèrent par la place qu'elles donnent au module de surveillance par rapport au système de commande :

- Une surveillance intégrée à la commande.
- Une surveillance séparée de la commande.
- Une approche mixte, c'est à dire ; combinaison des deux approches précédentes.

Dans la première approche, le système de surveillance est intégré à la commande. Elle considère que les fonctionnements anormaux doivent être connus à l'avance et introduites dans le système de commande. Cela suppose une connaissance absolue de toutes les évolutions possibles du système. De plus pour le diagnostic, le système doit être capable d'associer à n'importe quelle défaillance les causes probables.

Dans la seconde approche, le système de commande et celui de la surveillance sont séparés. Toutes les fonctions de la surveillance seront séparées de la commande ; cette séparation a l'avantage de soulager la commande et de pouvoir utiliser des nouvelles techniques pour la surveillance (Intelligence Artificielle). Mais elle présente l'inconvénient de générer des conflits entre la surveillance et la commande qui agissent toutes les deux sur le procédé. Ces conflits proviennent de la séparation entre les situations normales et anormales. En effet, ce qui est normal pour la surveillance ne l'est peut être pas pour la commande.

L'approche mixte est un compromis entre les deux précédentes; les fonctions de diagnostic et de décision sont séparées alors que les fonctions de détection et de reprise sont intégrées à la commande. Dans ce cas le système de commande définit le comportement dit normal, ainsi toute évolution non prévue dans ce modèle de commande sera alors considérée comme anormale. L'avantage de cette approche réside dans le fait que la limite entre normal et anormal est établie dès que l'on spécifie le modèle de la commande [COM-98].

Nous présentons, dans ce qui suit, les approches développées dans différents laboratoires pour la surveillance des systèmes de production.

## 2-1. L'approche filtre

Dans cette approche, il s'agit d'insérer un ou plusieurs filtres, entre la commande et le procédé à surveiller ; Comme le montre la figure 3. Le principe consiste à n'exécuter de commande que lorsque l'état réel du système est cohérent avec cette dernière. L'état réel du système est obtenu à partir de l'ensemble des états capteurs à chaque instant. L'objectif de cette approche est de proposer la meilleure structure de filtre i.e. une structure qui ne réduit pas les performances du système de commande.

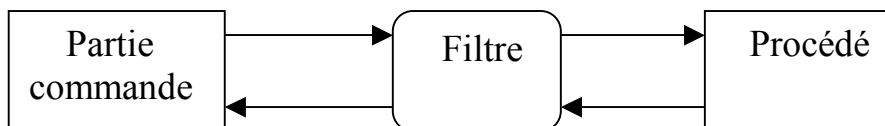


Fig. 3. :Principe de l'approche filtre

### 2.1-1. Approche filtre au CRAN à Nancy

L'approche du CRAN rejette le schéma bloc fonctionnel caractérisé par une vision atomique de l'interface Partie Opérative / Partie Commande (PO/PC), et propose d'appréhender cette interface avec une vision moléculaire[ALN 86]. Elle est constituée autour de la notion d'éléments de la partie Opérative (E.P.O). Cette dernière est vue alors comme un ensemble d'E.P.O (Eléments de la Partie Opérative).

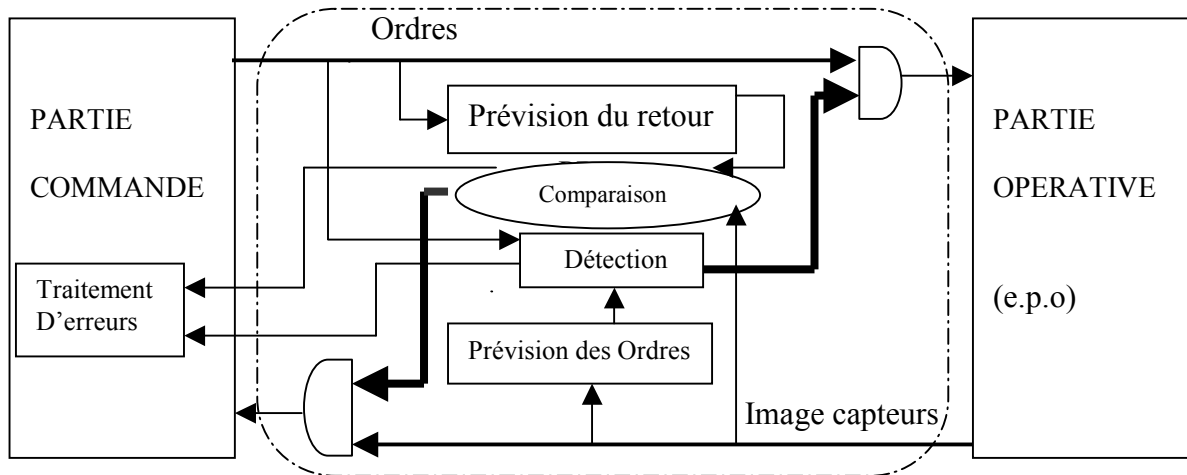


Fig. 4. : *Modèle de surveillance : approche filtre au CRAN.*

Le comportement d'un E.P.O est caractérisé par des états en nombre fini : des états physiques stables et des états transitoires. La modélisation consiste à identifier ces états, et à les organiser afin de caractériser l'évolution des caractéristiques observables par rapport aux ordres de la commande. Le modèle comportemental est un automate à états finis ; cette modélisation concerne le fonctionnement nominal de la partie opérative. Le modèle comportemental peut être utilisé de deux manières différentes : en émulateur ou en filtre [NOU-97].

Cette approche est basée sur l'utilisation d'un modèle de comportement normal de chaque E.P.O. Chacun de ces modèles est utilisé en tant que filtre de commande, leurs rôles consistent à vérifier d'une part que les requêtes issues de la commande sont cohérentes vis à vis de l'état courant modélisant les E.P.O et d'autre part que les comptes rendus reçus correspondent bien à ceux générés par le module de prévision.

Ce modèle, utilisé en émulateur ou en filtre, est développé essentiellement à des fins d'aide à la conduite et de détection des défauts de fonctionnement de la partie opérative.

### 2.1-2. Approche filtre au LAIL de Lille

Dans cette approche, développée au Laboratoire d'Automatique et Informatique Industrielle de Lille (L.A.I.L), la détection est basée sur un modèle temporel du procédé, elle porte sur l'existence et non sur la nature des défaillances, en utilisant un modèle de comportement normal pour chaque élément de la partie opérative.

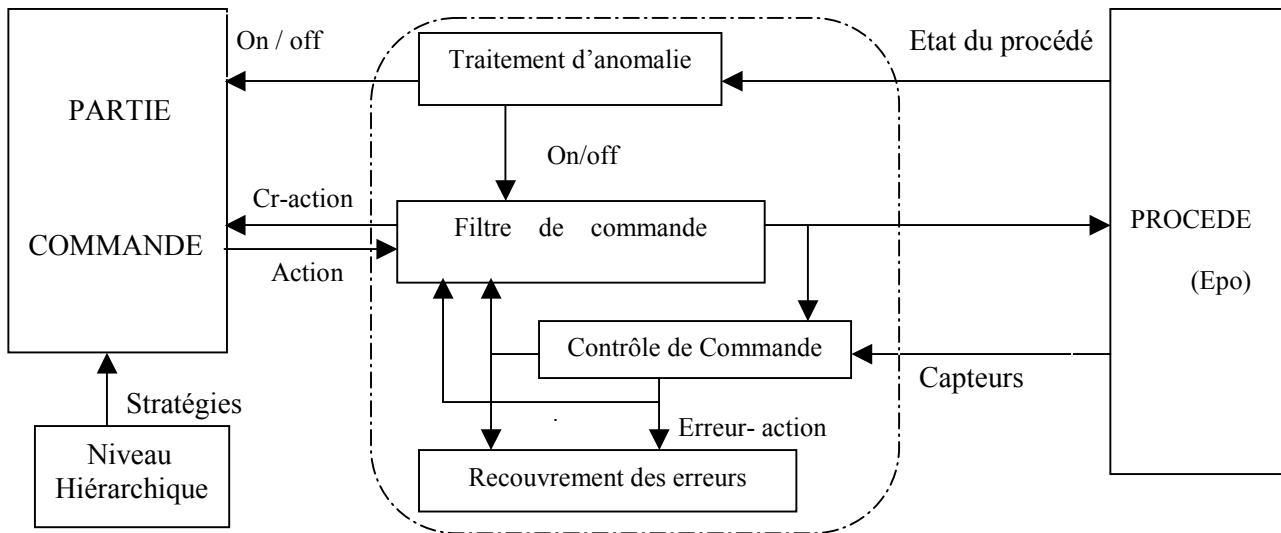


Fig. 5. : *Modèle de surveillance : approche filtre du LAIL.*

Les comptes rendus filtrés sont transmis vers la partie commande, les symptômes sont transmis vers un module de diagnostic, basé sur une modélisation du procédé reflétant sa structure interne. Cette modélisation repose d'une part sur le graphe fonctionnel qui définit les fonctions des différents éléments du procédé et leurs interactions et d'autre part sur les signatures temporelles causales qui sont un modèle du comportement anormal d'une partie du procédé, sachant qu'une signature temporelle causales établit une correspondance entre une séquence d'événements significatifs et une ou plusieurs causes de défaillance [NOU-97].

Le filtre de commande : analyse et transforme des commandes aveugles en commande valides en accord avec les contraintes fonctionnelles liant les différents éléments du procédé.

Le filtre de valeurs capteurs : compare les signaux émis par le procédé avec ceux prévisibles correspondant au fonctionnement normal (Une remise à jour est assurée pour modèle interne).

Une limitation réside dans la difficulté à adapter ce modèle à des ressources complexes disposant de plusieurs emplacements de stockage (convoyeurs par exemple) ou des procédés utilisant des produits à transformation non maîtrisable et les produits à changement d'états. (Gaz, liquide, ...) Cette restriction est due à la non-représentation de la dynamique par le modèle.

Le principe de diagnostic consiste à localiser, à l'aide des signatures temporelles causales, la zone défaillante, puis identifier la fonction susceptible d'être à l'origine de la défaillance dans cette zone, ceci est obtenu en exploitant le graphe fonctionnel. Les défaillances identifiées sont ensuite transmises à un module de recouvrement, qui met en œuvre un certain nombre d'actions pour agir sur la partie commande et sur la partie opérative pour assurer la sécurité du système de production.

Cette démarche de détermination des filtres de commande s'apparente à l'approche RW en ce qui concerne la séparation entre le fonctionnement physiquement possible et les contraintes de fonctionnement (sous formes d'états interdits). Cependant, les deux approches demeurent fondamentalement différentes : des problèmes liés à la théorie RW tels que la contrôlabilité et la synthèse de la supervision, ne sont pas abordés dans la détermination des filtres de commande [NOU-97].

## 2-2. L'approche Comparateur

Pour ce qui concerne cette approche, il s'agit de comparer en permanence l'état réel du système, donné par l'ensemble des états capteurs, avec celui donné par le modèle de comportement du système, comme le montre la figure 6. La détection d'une défaillance s'annonce à chaque fois que l'état réel du système s'écarte par rapport à celui donné par le modèle.

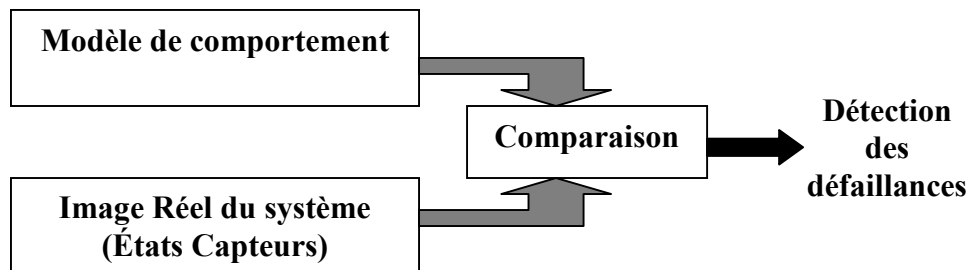


Fig. 6. *Principe de détection dans l'approche comparateur.*

### 2.2-1. Approche de l'université de Carnegie Mellon

Les travaux de L.Holloway et B.Krogh [HOL-90] [HOL-91], à l'université Carnegie Mellon, placent le modèle du procédé en tant qu'émulateur des évolutions normales de la partie opérative. Son rôle est en effet de calculer les fenêtres temporelles d'occurrence des comptes rendus émis par le procédé quand celui-ci est soumis à une commande particulière. Pour une consigne donnée, un bloc de comparaison permet de vérifier si un compte rendu émis par le procédé arrive bien à la date prévue par le modèle. Ce bloc assure également la mise à jour du modèle de comportement (corrections) permettant ainsi de suivre les éventuelles dérives des caractéristiques du procédé (en particulier le vieillissement du matériel) [ZAM-97].

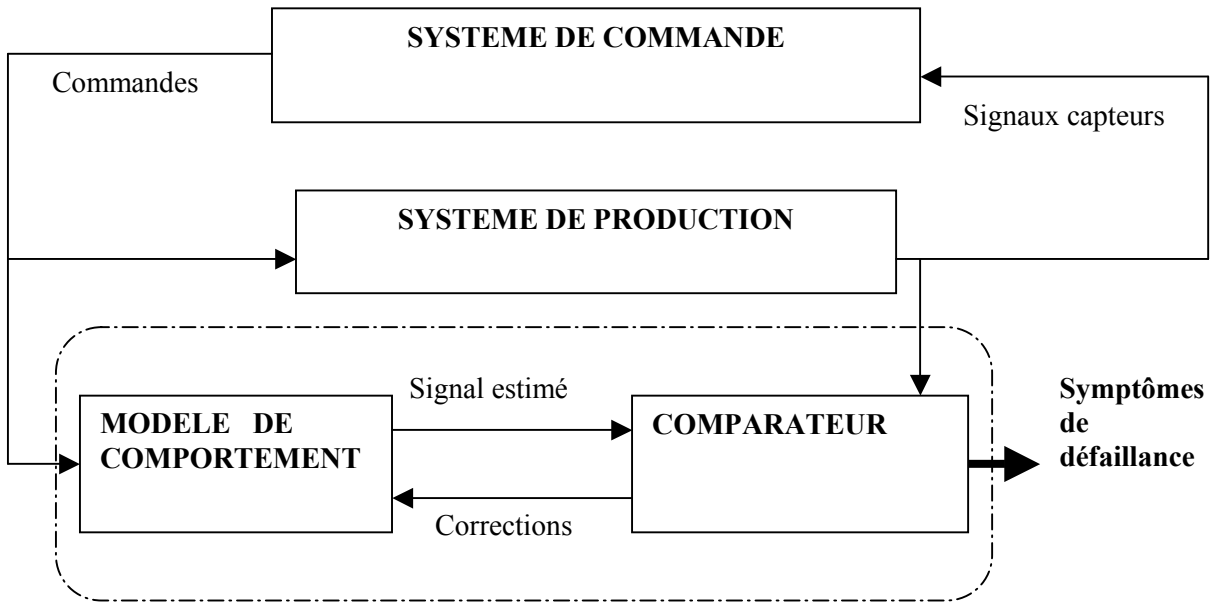


Fig. 7. *Approche comparateur de l'université Carnegie Mellon.*

Notons que lorsqu'une défaillance est détectée, le système de comparaison effectue un pré diagnostic en situant précisément le contexte de la détection : compte rendu concerné, commande en cours et type de symptôme. Contrairement aux autres approches, seules les défaillances du procédé sont ici détectées. De ce fait, si les consignes envoyées au procédé et à l'émulateur ne sont pas cohérentes vis à vis de l'état courant du procédé, rien ne garantit que l'émulateur aura le même comportement que le procédé réel. Si ce n'est pas le cas, une erreur de commande sera interprétée comme une défaillance du procédé. Le modèle du procédé est alors désynchronisé de la partie opérative. Néanmoins, limitée à des problèmes d'états défendus, l'approche de Krogh a pour objectif la commande et non la surveillance [NOU-97].

### 2.2-2. Modèle émulateur de LAI de Lyon

Dans le cadre de la surveillance, cette approche intègre au système de commande, par retour d'état, un système de détection et de diagnostic. Ce système requiert un modèle du procédé ; appelé modèle de référence, qui représente le comportement du procédé sans faute [NID-96].

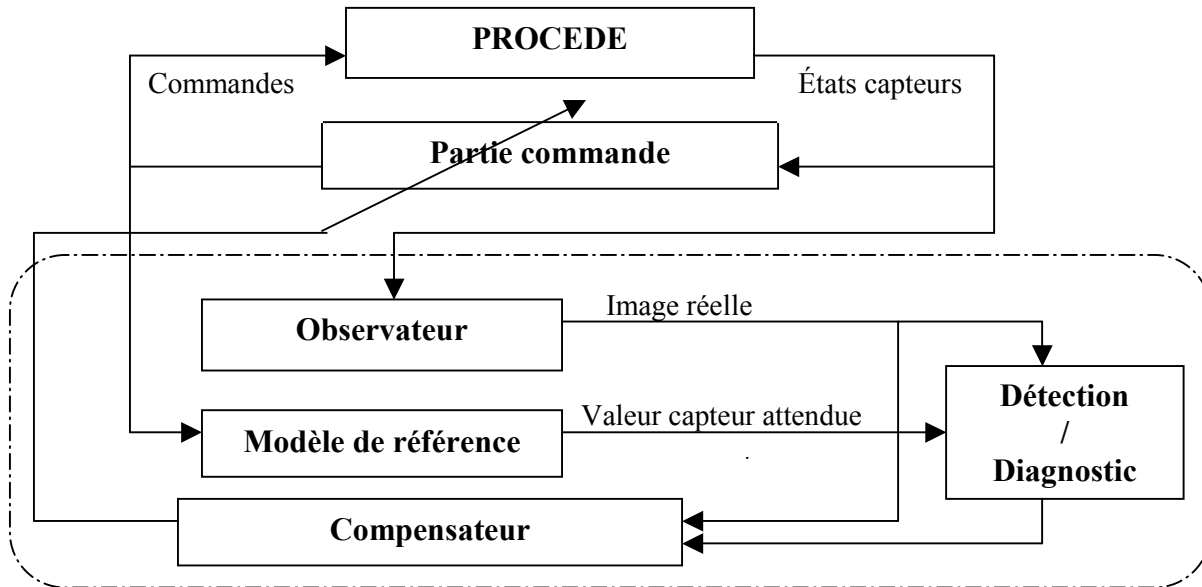


Fig. 8. : *Modèle de surveillance orienté sécurité opérationnelle du LAI*

Ce modèle transforme alors une commande en un vecteur de marquage. La détection est fondée sur la comparaison du vecteur de marquage construit à partir des comptes rendus émis par le procédé et de celui émis par le modèle du procédé. Le système de diagnostic se charge ensuite de déterminer et de localiser la faute selon le processus de danger.

C'est une approche comparateur orientée sécurité opérationnelle : Son but est le fonctionnement sûr, sans danger, ni risque d'accidents ; il autorise un fonctionnement sous fautes avec des risques contrôlés.

Cette approche détecte et pré-diagnostique une défaillance, ainsi à la détection d'une dérive de comportement, des actions de compensations sont engagées dans le but de contrôler l'évolution vers certains états interdits, malgré la faute. Le compensateur se charge de l'exécution du service engagé en évitant de passer par les états interdits [NID-96].

### 2-3. Approche hiérarchique par modèle de référence

L'approche développée au L.A.A.S est basée sur deux concepts : la hiérarchie et le modèle de référence, ce dernier est utilisé par la commande comme modèle du procédé et contient un modèle des comportements normaux de la partie opérative [COM-98]



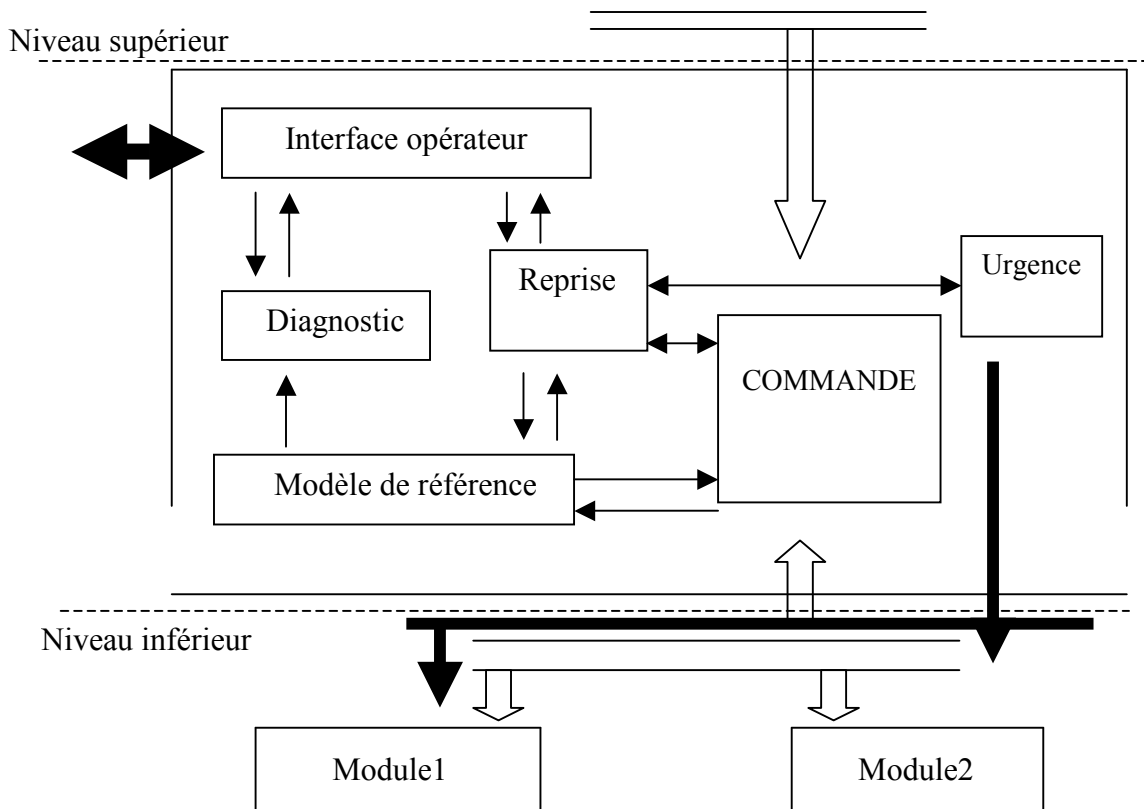


Fig. 9. *Modèle de surveillance-commande avec modèle de référence au LAAS*

Dans cette approche, la fonction de détection est intégrée à la commande et les fonctions de diagnostic, de reprise et d'urgence sont séparées de la commande. Si un module n'arrive pas à diagnostiquer ou à résoudre la défaillance pendant la fenêtre temporelle qui lui est alloué par l'ordonnancement temps réel, il doit passer un message au module responsable dans le niveau supérieur [ZAM-97].

Lors d'une situation de défaillance, le modèle ne peut plus représenter l'état réel du procédé et dans ce cas le système de surveillance ne dispose plus d'aucune image correcte du procédé, nécessaire à tout diagnostic et à toute reprise efficace.

Lorsqu'une défaillance se propage d'une activité à une autre, la défaillance n'est pas détectée pendant l'activité en cours et se répercute sur une des activités suivantes. La détection a donc lieu dans un module différent de celui responsable de cette défaillance. Le système de diagnostic, compte tenu des informations locales dont il dispose, il ne peut conclure quant à l'activité responsable de la défaillance.

## 2-4. Synthèse- Conclusion

Dans toutes les approches présentées, les mécanismes de détection des défaillances sont basés toutes sur le modèle de comportement normal. Ce dernier est caractérisé par un nombre d'états fini. La nécessité de définir à priori la liste exhaustive de tous les états normaux ou anormaux constitue là l'inconvénient majeur de ces approches. De plus, la non prise en compte de la dynamique du système et bien sûr, le suivi temporel dans l'exécution des tâches est une faiblesse incontestable de ces différentes approches.

De plus, les situations d'incohérences, entre les commandes et l'état du système à surveiller sont très probables d'où les possibilités fréquentes d'avoir des fausses alarmes. Ces fausses alarmes risquent de mener à des situations qui imposent l'arrêt de production, et donc affectent considérablement la disponibilité des équipements.

Dans l'objectif d'améliorer la qualité de production, nous nous proposons de développer une nouvelle structure de surveillance, supervision et maintenance capable de réagir à différentes situations sans la connaissance préalable des modes de défaillance du système ; c'est à dire sans avoir une liste exhaustive de toutes les défaillances possibles du système, c'est le point fort de notre démarche. Notre solution, au niveau de la détection des défaillances, est basée sur la prise en compte du caractère temporel du système à surveiller comme nous allons le voir avec plus de détails dans le chapitre 3.

## 3. PROPOSITION D'UNE DEMARCHE GÉNÉRALE

Le problème qui se pose est la détermination des informations nécessaires pour établir une procédure de surveillance ainsi que l'interprétation de la différence entre les grandeurs mesurées et les grandeurs nominales. Certaines approches de surveillance que l'on trouve dans la littérature, sont basées sur les relations entre la commande et le procédé alors que d'autres sont davantage liées à l'évolution dynamique de la partie opérative. Dans le cas des systèmes manufacturiers, la dynamique d'un atelier de production est très importante, elle est liée d'une part, à **l'évolution du flux de pièces** finies ou semi-finies et d'autre part à la **vitesse d'usinage**. Une défaillance peut donc entraîner soit une variation de ce flux au niveau de chaque machine soit un retard d'usinage. Ainsi, l'évolution dynamique dans un système de production peut être caractérisée soit par le niveau des stocks soit par le temps de service.

Nous nous sommes fixé comme objectif, l'intégration des outils de surveillance, supervision et de maintenance dans un système de production déjà commandé. Ce qui justifie le choix de notre approche de surveillance globale. Le module de surveillance reçoit alors les informations correspondant à l'ensemble des états capteurs, c'est l'élément de base utilisé pour la détection des défaillances et par la suite la localisation (diagnostic) de la ou des sources de pannes (le mécanisme sera détaillé dans les parties suivantes).

Une fois la défaillance identifiée, les informations la concernant seront dirigées d'une part, vers le module de supervision qui en fonction du type et de la gravité de la panne générera des séquences de reprise, de recouvrement ou de traitement d'urgence, et d'autre part vers le module de maintenance qui collecte ces données afin de savoir à quel moment il faut engager les opérations de maintenance nécessaires. Ceci est illustré dans la figure 10.

## 1-2. Description des différents modules de la structure de surveillance

L'idée est de chercher à mettre en œuvre un modèle de surveillance, dont l'évolution est basée seulement sur l'ensemble, en temps réel, des états capteurs du procédé, représente alors l'originalité de notre travail. Notre démarche est basée sur le caractère temporel du modèle de surveillance. Seul le modèle dynamique du système ; obtenu à partir du modèle de commande, sera utilisé pour la détection et le diagnostic des défaillances.

Dans le module de surveillance, on retrouve les fonctions clés : détection et diagnostic. Les fonctions de reprise, recouvrement et traitement d'urgence sont comprises dans le module de supervision, qui pour agir sur le procédé doivent passer par la commande, en générant des séquences spécifiques. Alors que le module de maintenance collecte les informations du module de surveillance (vieillesse des installations ou usure des outils par exemple) et du module de supervision (statistiques des pannes par exemple), et le traitement de ces informations permet d'établir les programme d'interventions sur les installations [SIM-03].

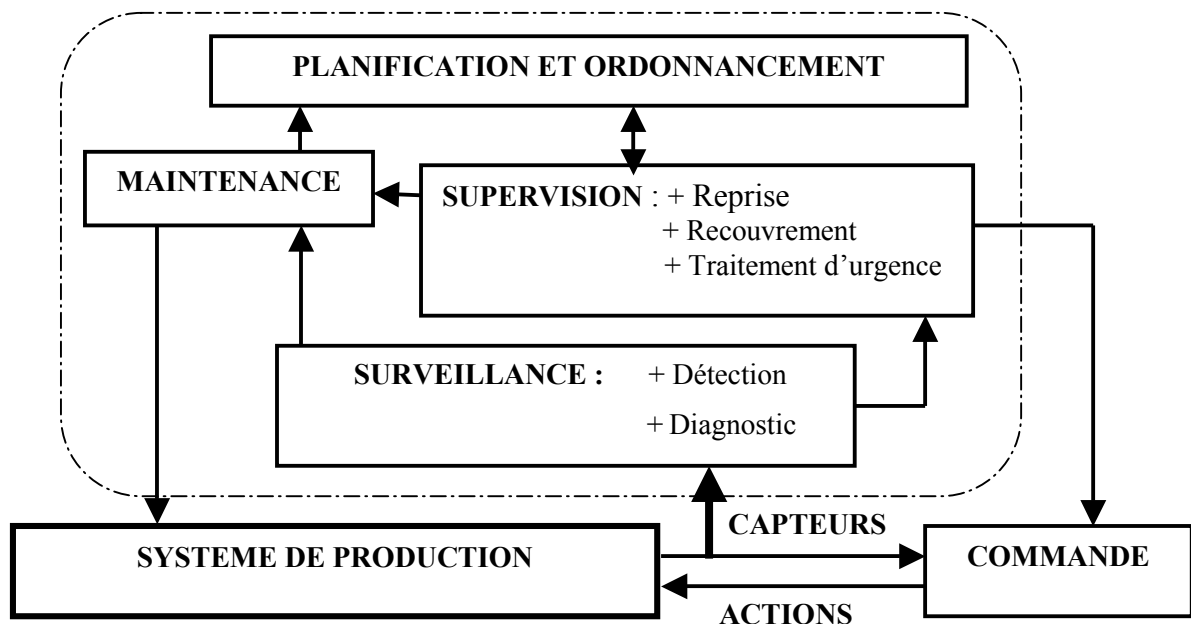


Fig. 10. *La structure globale proposée.*

Les seules informations en entrée du module de **surveillance**, sont les états capteurs en provenance du système de production comme le montre la figure 11. Ce module renseigne les modules de maintenance et de supervision. Les informations destinées à la maintenance telles que l'usure des outils ou des machines de production sont nécessaires pour établir le planning des interventions. Quant aux informations destinées à la supervision, telles que l'état défaillant du système et les causes de la défaillance, permettent de prévoir la reprise, le recouvrement ou le traitement en agissant sur la commande du système.

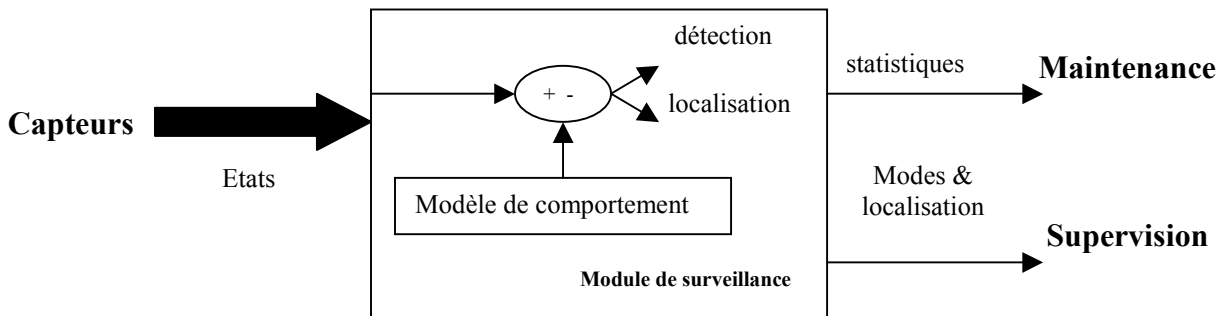


Fig. 11. Identification des entrées/sorties du module de surveillance.

Dans le module de **maintenance**, comme le montre la figure 12, les informations dirigées vers le procédé correspondent aux interventions des équipes de maintenance, qui agissent directement sur le système de production. Alors que celles dirigées vers le module de planification et d'ordonnancement correspondent aux comptes rendus sur les états des machines ou le mode de fonctionnement du système. Sachant que ce module assure la mise à jours de la base de données (interventions, statistiques, ...), en exploitant les informations venant, du module de surveillance et/ou du module de supervision.

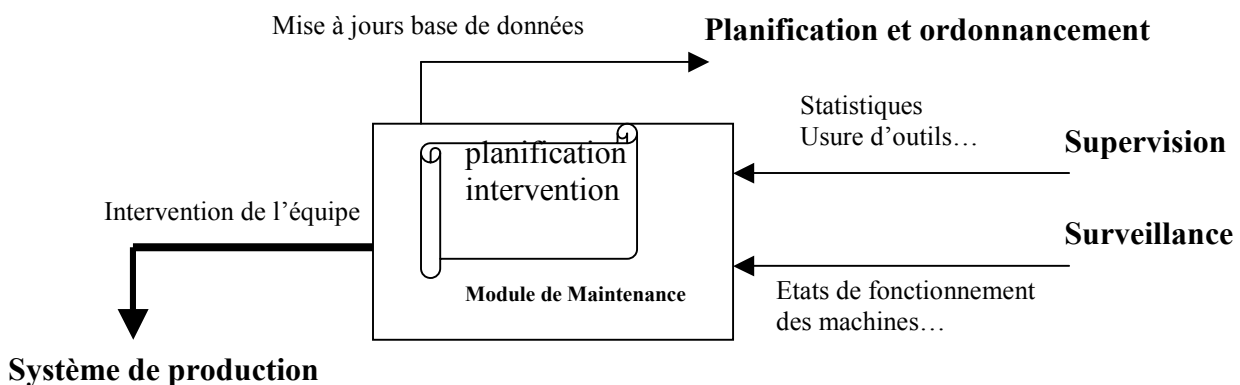


Fig. 12. Identification des entrées/sorties du module de Maintenance.

Le module de **supervision**, représenté sur la figure 13, exploite les informations sur l'état de fonctionnement du système (venant du module de surveillance). Il doit générer les séquences de reprise, de recouvrement ou d'urgence, adéquate pour agir sur le système afin de le ramener dans une situation normale de fonctionnement. Ceci est réalisé par action sur le système de commande qui agit seul et directement sur le procédé. Ce même module transmet des informations comme les statistiques et les fréquences des pannes, au module de maintenance et au module de planification et ordonnancement. Ce dernier peut à son tour, dans le cadre de la gestion de production, donner les informations sur le type de commande à mettre en service, dans le cas des systèmes flexibles de production.

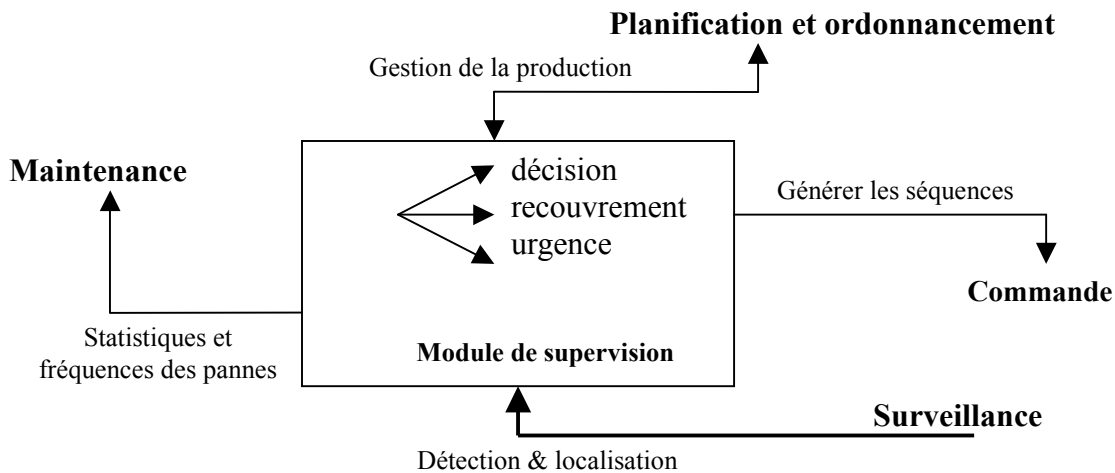


Fig. 13. Identification des entrées/sorties du module de supervision.

### 3-2. Principe de détection

Pour les systèmes de production étudiés, les échanges entre la partie opérative et la partie commandes permettent d'identifier tous les états possibles du système. La construction du modèle de comportement est basée sur cet ensemble d'états. Ainsi, pour la détection d'éventuelles défaillances, ce modèle est comparé de façon synchrone avec l'évolution du système observé comme le montre la figure 14.

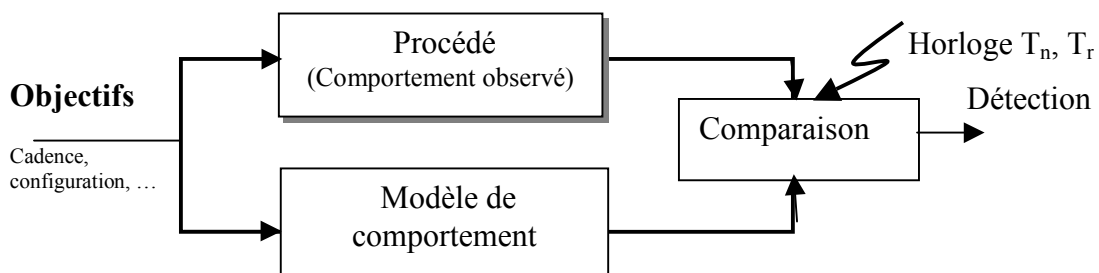


Fig. 14. Principe de la détection

Pour une commande donnée, appliquées simultanément au procédé et au modèle de comportement (ce dernier représente le fonctionnement normal du procédé), correspond un événement qui signifie la fin de l'exécution de la tâche associée à cette commande. Sachant que cet événement devrait se présenter au bout d'un temps  $T_n$  (temps prévu dans le modèle de comportement normal) à partir de l'instant de l'envoi de la commande, alors la **détection** se fait par la comparaison de ce temps  $T_n$  et le temps écoulé réellement par le procédé que l'on note  $T_r$ . Nous verrons dans le chapitre 3 qu'une différence sera tolérée et correspondra au fonctionnement dégradé.

## MODÉLISATION DU SYSTÈME DE SURVEILLANCE

*La surveillance d'un système consiste à recueillir, d'une manière passive, en permanence tous les signaux en provenance du procédé. Sachant que le bon fonctionnement du système correspond à l'exécution de toutes les tâches du procédé pendant des intervalles de temps bien définis ; le système de surveillance est alors basé sur le contrôle de la durée d'exécution des tâches au cours du fonctionnement du système. C'est l'idée de base de notre système de détection des défaillances et de diagnostic ; donc de surveillance. L'outil qui nous paraît le plus simple à utiliser et puissant par sa nature c'est le modèle automate temporisé.*

*Dans ce chapitre, après un bref rappel sur les systèmes à événements discrets, nous présentons les outils de modélisation auxquels nous nous sommes intéressés dans nos travaux de recherche : les automates temporisés et statecharts. Ces modèles héritent la puissance de modélisation des réseaux de Petri autonome. De plus ils permettent de représenter la plupart des contraintes temporelles qui interviennent dans le fonctionnement d'un système à événements discrets.*

## 1. SYSTÈMES À ÉVÉNEMENTS DISCRETS (S.E.D): RAPPEL

On parle de systèmes à événements discrets lorsque l'on ne s'intéresse à l'état de l'atelier qu'à des instants particuliers (événements correspondant par exemple aux débuts et aux fins d'opérations). C'est par exemple, les ateliers manufacturiers (usinage, assemblage etc.) [VAL-00], [COM-91], [TOG-92]. Nous nous intéressons également aux systèmes de fabrication par lots (systèmes discontinus) couramment rencontrés dans l'agroalimentaire et la chimie fine. Pour ces systèmes, en plus de l'état de l'atelier lors des débuts et fins d'opérations, il faut veiller continuellement à ce que certains équipement (cuves de stockage par exemple) ne soient jamais vides ou ne débordent pas.

La surveillance de tels systèmes consiste à mettre en œuvre un plan de production et à contrôler son exécution pas à pas. Le contrôle de cette exécution consiste à vérifier que les opérations sont bien exécutées dans l'ordre prévu et aux dates prévues, que les ressources (machines ou opérateurs) ne sont pas défaillantes que la sécurité des personnes et des équipements est assurée. Cela implique donc de pouvoir réagir en temps réel à toutes les défaillances possibles et plus généralement à toutes les anomalies [VAL-00].

Un système à événements discrets est un système dynamique dans lequel l'espace des états est discret. Ses trajectoires d'états sont constantes par morceaux. Un tel système évolue conformément à l'occurrence des événements physiques à des intervalles de temps généralement irréguliers ou inconnus.

Considérons, par exemple, une machine "distributeur de boisson" qui peut être dans trois états : arrêt, marche et panne. On suppose qu'il peut y avoir l'occurrence de cinq événements : pièce introduite, choix de la boisson, traitement achevé (boisson prête), panne et réparation. Ces événements sont étiquetés par les symboles **p**, **c**, **f**, **d** et **r** respectivement. Une évolution possible de ce système est présentée dans la figure suivante.

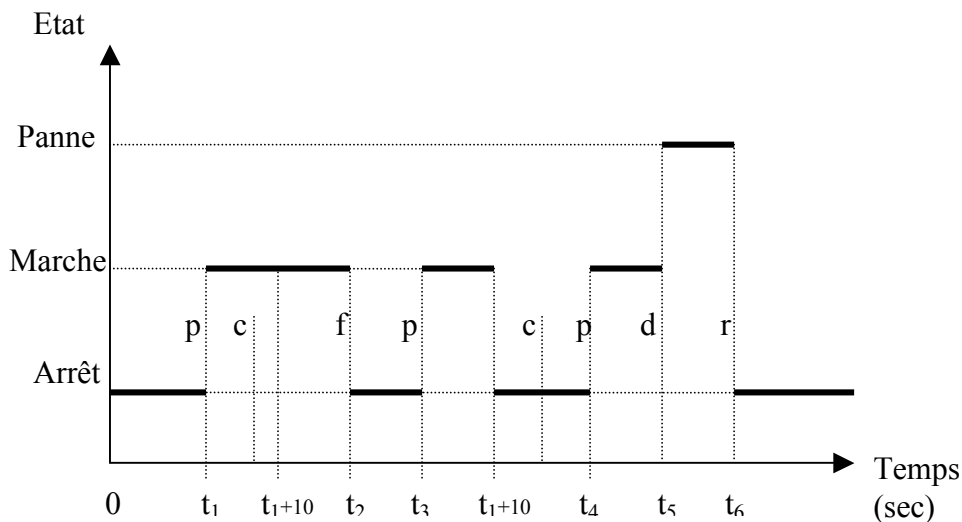


Fig. 15. *Evolution de l'état de la machine "distributeur de boisson".*



Initialement la machine est en état d'arrêt. Suite à l'arrivée de l'événement **p** à l'instant  $t_1$ , la machine passe dans l'état de marche. De façon similaire, le système évolue aux instants  $t_2$ ,  $t_3$ ,  $t_4$  et  $t_5$  sur l'occurrence des événements **p**, **f**, **c**, **d** et **r**.

L'évolution d'un S.E.D est caractérisée par l'occurrence des événements. Selon la manière de modéliser l'arrivée des événements, on peut classer les modèles des S.E.D en trois catégories : modèles logiques (TOR), modèles temporisés (déterministes) et modèles stochastiques.

Dans certaines applications, l'information temporelle est essentielle et doit être prise en compte explicitement par le modèle. Les modèles qui ont cette caractéristique sont appelés temporisés. Il s'agit des modèles où le temps est déterministe (réseaux de Petri temporels, automates temporisés) et des modèles où le temps est aléatoire (chaîne de Markov, réseaux de files d'attente).

Les systèmes de production sont généralement caractérisés par un aspect structurel statique, auxquels ils sont associés des grandeurs évolutives constituant la partie dynamique. Il est donc nécessaire de disposer d'un outil pour aider le concepteur à garder une démarche cohérente pour la modélisation de systèmes de production. Nous présentons deux parmi les possibilités de modélisation, simulation et analyse des systèmes de production automatisés à l'aide des outils : réseaux de Petri et automates temporisés.

## 2. MODÉLISATION PAR RESEAUX DE PETRI [ALA-03]

Les réseaux de Petri, initialement développés par Carl Adam PETRI, présentent deux caractéristiques intéressantes. D'une part, pour l'abondance des résultats théoriques basés sur l'utilisation de cet outil. D'autre part, ils permettent de modéliser des systèmes mettant en jeu des mécanismes de parallélisme, de synchronisation et de partage des ressources.

### 2-1. Modélisation d'un système de production

Considérons le cas d'un système de production de type série, formé par deux machines et trois stocks. Nous nous proposons de surveiller cet atelier.

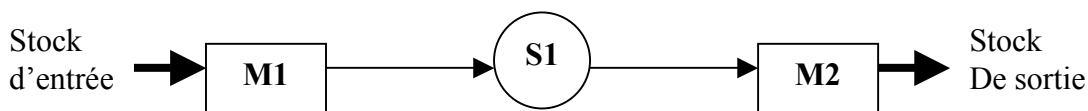


Fig. 16. *Exemple de système de production type : "série".*

**2.1-1. Caractéristiques et modélisation d'une machine**

Une machine  $i$  est généralement caractérisée par un taux moyen de chargement noté :  $\epsilon_i$  (ou aussi vitesse moyenne de mise en route) et un taux moyen de service (sans défaillance) noté :  $\delta_i$  c'est la cadence de sortie d'une pièce usinée.  $\epsilon_i$  et  $\delta_i$  sont connus et peuvent varier légèrement, ici nous les supposons constants.

Malheureusement une machine est toujours sujet à des défaillances, à cause de son vieillissement, dérèglement ou autres. Après défaillance, la machine se trouve dans un état de panne. La fréquence d'occurrence de ces pannes est caractérisée par un **taux moyen de défaillance** noté :  $\lambda_i$  il caractérise le nombre total de défaillance pendant la durée de service.

En fin, un **taux moyen de réparation**, noté :  $\mu_i$  est introduit, il caractérise le nombre total de réparations effectuées par rapport au temps total des réparations.

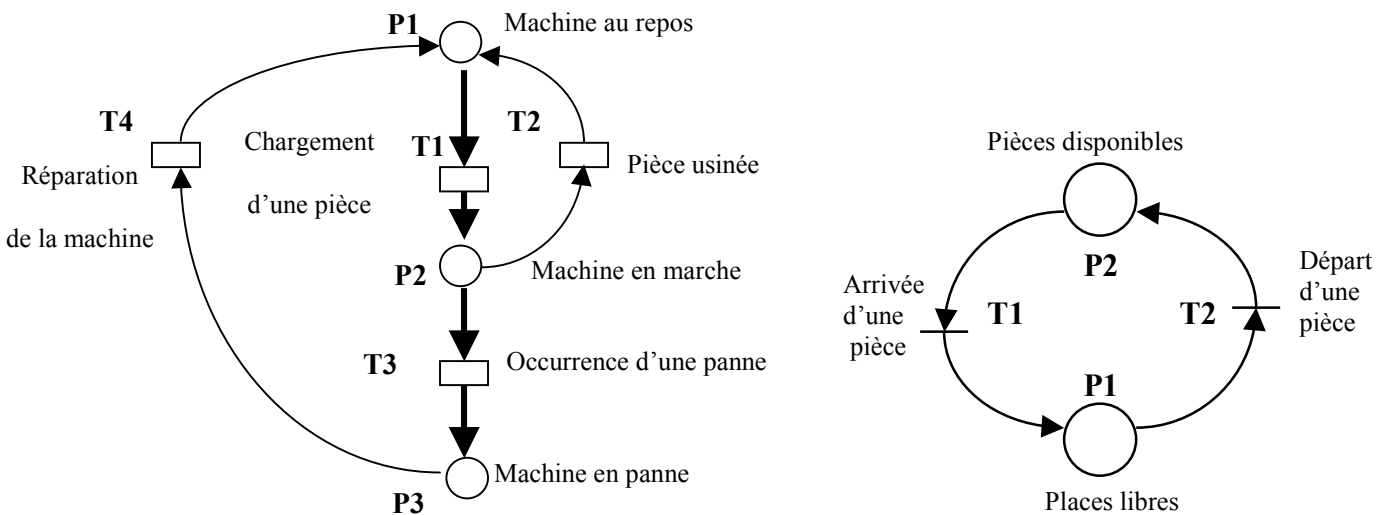


Fig. 17.	a) <u>Modélisation du fonctionnement d'une machine</u>	b) <u>Modélisation du stock</u>
----------	--	---------------------------------

**2.1-2. Procédure de détection des défaillances**

La détection des défaillances est basée sur la comparaison des informations venant du procédé réel à celles données par le modèle de comportement normal, comme le montre la figure 14. Ici la démarche repose d'une part sur la modélisation pour la simulation du modèle de référence. D'autre part sur l'ensemble des paramètres à surveiller du système.

Pour le modèle de référence nous utilisons, le modèle générique de la machine, représenté par la figure 18, en tenant compte des caractéristiques précédemment définies.

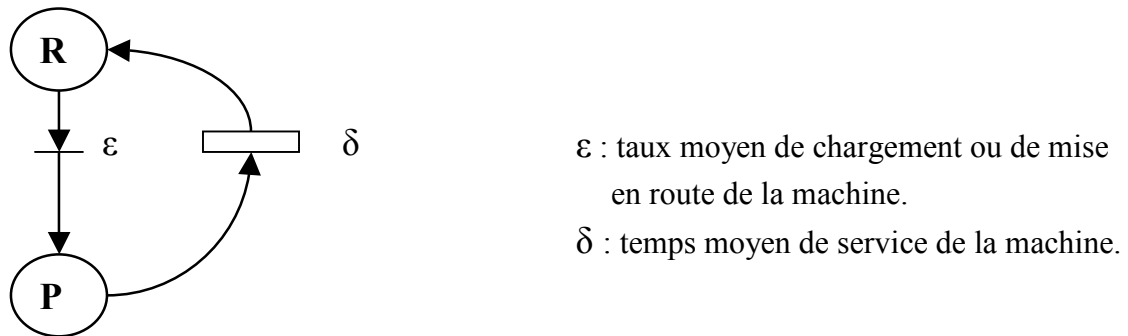


Fig. 18. Modèle générique de la machine.

## 2-2. Choix des grandeurs à surveiller

Dans un atelier de production manufacturier, les grandeurs accessibles pouvant donner l'information sur l'existence d'une déviation par rapport au bon fonctionnement de la machine sont généralement : le temps de service (usinage, traitement, déplacement...) et le niveau de stock au niveau de chacune des machines de l'atelier.

### 2.2-1. Surveillance basée sur le niveau des stocks

Les effets des pannes affectant les machines se répercutent sur le remplissage des stocks. En effet, le passage d'une machine à un état de panne conduira à une diminution du nombre des pièces dans ses stocks avals et une augmentation de ce nombre dans les stocks amonts. Le contenu d'un stock est donc une grandeur qui peut être utile pour la détection et la localisation d'une panne.

Vu que le système étudié est modélisé par des paramètres stochastiques, le nombre de pièces dans un stock à un instant donné au cours du fonctionnement nominal est aléatoire. De ce fait la comparaison de ce nombre avec celui mesuré sur le système en fonctionnement réel n'apporte aucun renseignement sur son état. Il est alors nécessaire de choisir une grandeur plus stable. L'échantillonnage de la durée de fonctionnement permet de déterminer sur chaque période le nombre moyen de pièces dans le stock. Cette grandeur a aussi un caractère aléatoire, cependant on peut faire l'hypothèse qu'elle est généralement comprise dans une marge qui varie d'une période à une autre au cours du régime transitoire et tend à se stabiliser en régime permanent. La détermination des bornes de ces marges peut s'effectuer par simulation du modèle de référence. La surveillance du système se fait alors par comparaisons des valeurs moyennes mesurées au cours du fonctionnement réel avec les marges fournies par le modèle de référence.

A chaque période d'échantillonnage on compare le nombre moyen de pièces dans le stock avec la marge correspondante à cette période ; comme illustré dans la figure 19. Cette comparaison mène aux déductions suivantes:

- Si le nombre moyen dépasse la borne supérieure de la marge, on peut conclure à la panne de la (ou des) machine (s) situées en aval du stock.
- Si le nombre moyen est au-dessous de la borne inférieure, on peut conclure à la panne de la (ou des) machine (s) situées en amont du stock.
- Si le nombre moyen appartient à la marge, aucune panne n'est détectée.

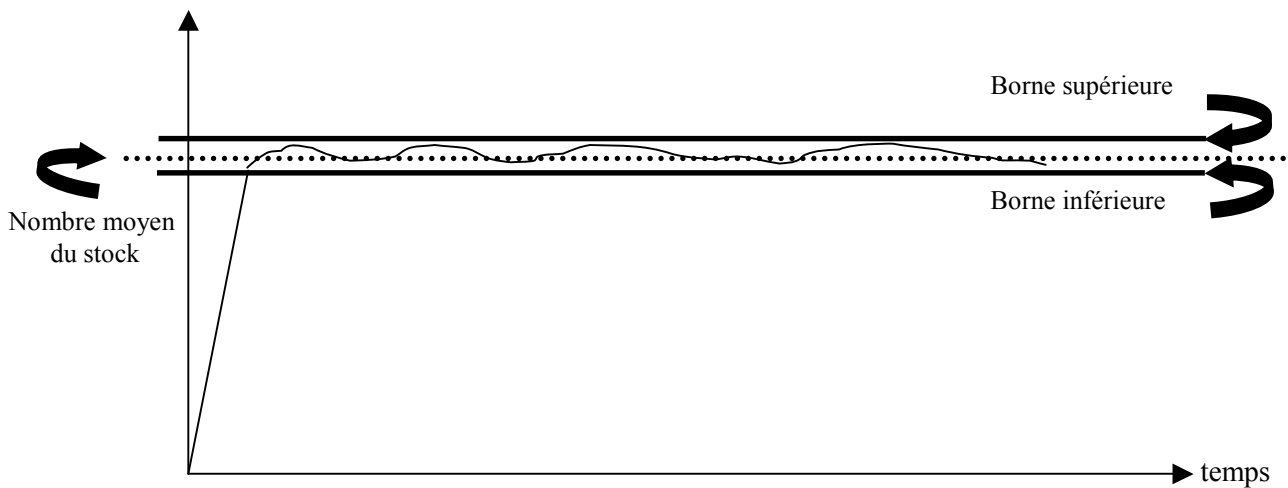


Fig. 19. *Principe de surveillance basée sur le niveau de stock.*

Le développement de cette approche présente les difficultés suivantes :

- difficulté du choix de la période d'échantillonnage adéquate au système étudié.
- les marges de référence sont déterminées de manière approximative.
- approche insuffisante pour surveiller des systèmes complexes.
- après réparation d'une machine les grandeurs de références ne sont plus valables.

### 2.2-2. Surveillance basée sur le temps de service

La durée de traitement d'une pièce sur une machine est par hypothèse une variable qui dépend de l'opération à réaliser. L'idée est d'associer à chaque machine de l'atelier de production une valeur maximale relative à sa durée de traitement. La surveillance est alors basée sur le suivi de ces durées pendant le fonctionnement réel. Une panne est détectée si cette

durée dépasse un seuil fixé. L'objectif est de déterminer la valeur du seuil noté  $\Delta$  à fixer pour optimiser le temps de détection. En effet, des seuils élevés ne permettent pas de détecter à temps une panne, par contre des faibles seuils déclenchent trop de fausses alarmes. Le principe de cette approche permet de détecter les pannes avec un retard minimal égal au seuil. C'est cette approche que nous allons exploiter pour la surveillance et le diagnostic des systèmes de production.

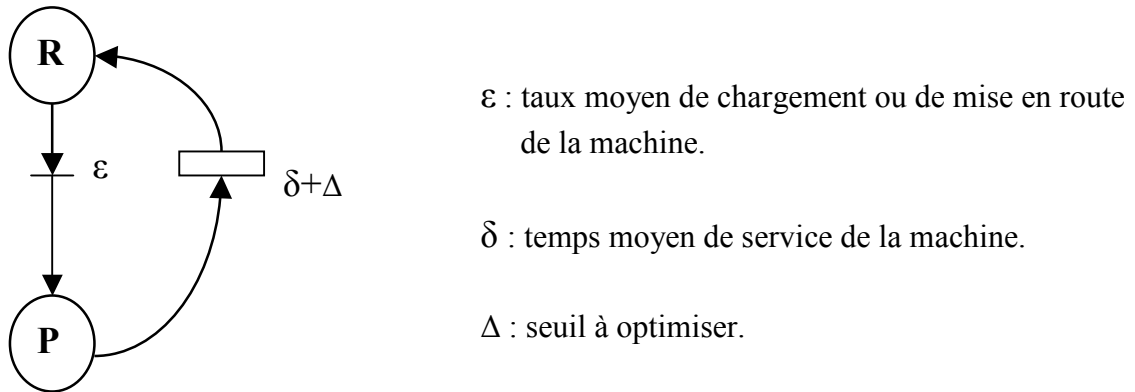


Fig. 20. *Modèle de l'approche de surveillance basée sur le temps de service.*

### 2.2-3. Détermination du temps de détection

Pour déterminer le temps de détection, nous avons procédé par simulation d'un système de production. Pour cela, nous modélisons le fonctionnement du système à l'aide du logiciel **MISS-RdP** (Interactive Modelling and System Simulation by Petri Nets). Il est basé sur les techniques des réseaux de Petri stochastiques, interprétés, temporisés, synchronisés et colorés. Il est aussi particulièrement adapté pour tenir compte des événements déterministes (les décisions, retard...) ou des événements aléatoires (pannes, temps de réparation...). Enfin, il est équipé d'une interface graphique qui permet l'édition des modèles et de simuler les comportements dynamiques des systèmes.

Dans la suite, nous nous proposons d'étudier le système de production présenté au début de ce paragraphe ; Composé de deux machines M1 et M2 en série, dont voici la modélisation par réseaux de Petri représenté par la figure 21.

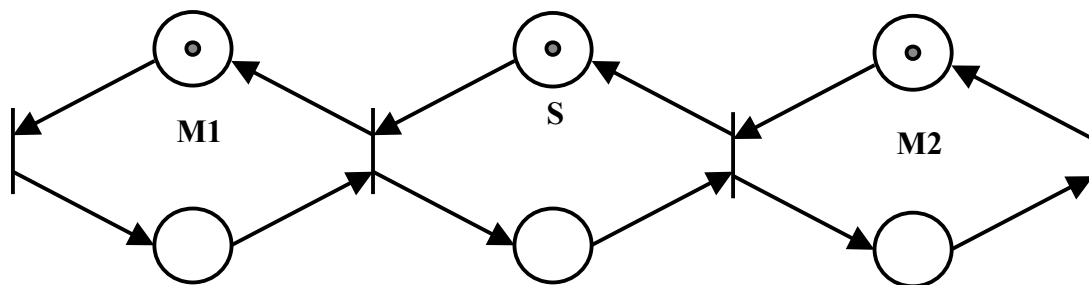


Fig. 21. *Modèle réseau de Petri d'une ligne série.*

La modélisation de ce système par MISS-RdP est donnée par la figure 22. Dans ce modèle, nous modélisons seul le fonctionnement idéal du système de production étudié, chose impossible dans la réalité à cause des perturbations fréquentes de nature très diversifiée.

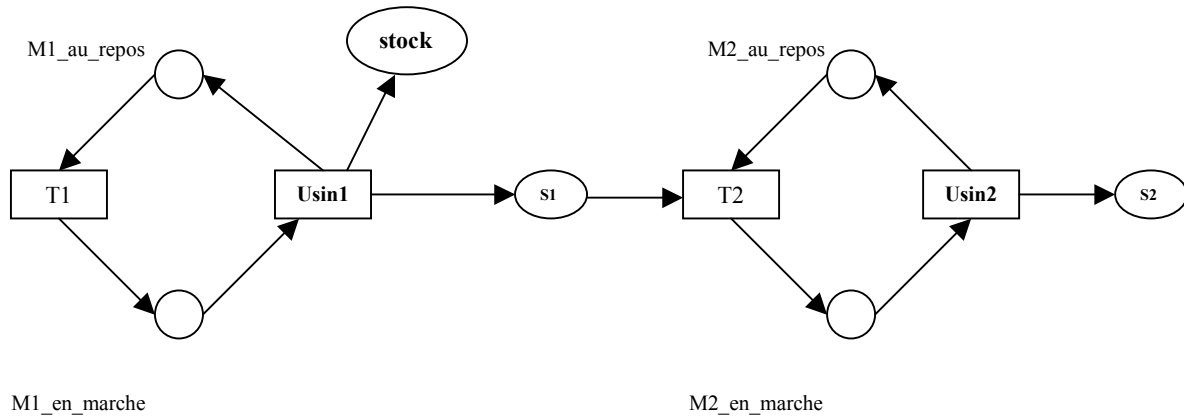


Fig. 22. Modélisation du système étudié.

Dans l'objectif de déterminer les valeurs des différents seuils, correspondant aux durées d'exécution des tâches d'usinage et ceci pour chacune des deux machines, La modélisation par l'outil MISS-RdP, du système avec prise en compte de l'injection aléatoire des défaillances est donnée par la figure 23. L'optimisation consiste à faire varier la valeur de la tolérance notée :  $\Delta$  et trouver l'optimum qui correspond au nombre minimal de fausses alarmes. La courbe qui représente les nombres de fausses alarmes et des défaillances, en fonction de la marge de tolérance notée :  $\Delta$ , nous permet de trouver graphiquement la valeur du seuil. Il correspond alors au minimum de nombre de fausses alarmes tout en détectant toutes les défaillances susceptibles de contraindre la production.

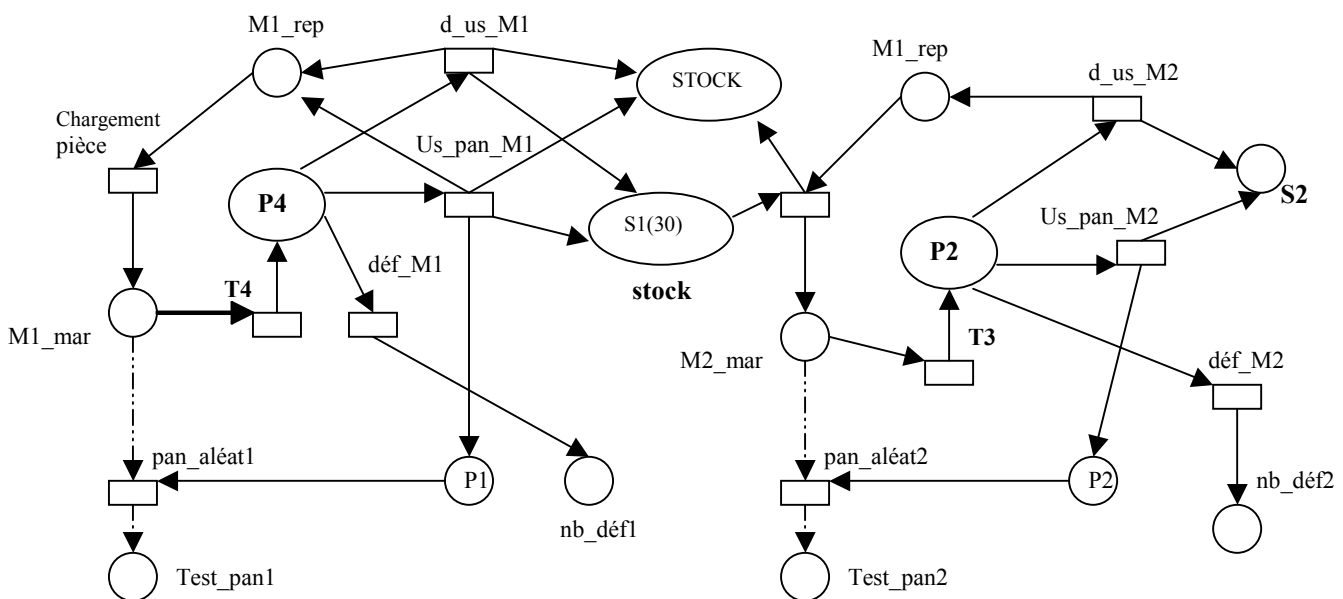


Fig. 23. Modèle de simulation pour l'optimisation du temps de détection.

#### 2.2-4. Résultats de simulations

Selon le choix de la valeur du seuil  $\Delta$  (retard toléré pour le temps d'usinage), nous obtenons un certain nombre d'alarmes. Parmi ces alarmes nous trouvons des fausses alarmes qui correspondent à des variations ponctuelles du temps d'usinage dépassant le seuil fixé. Alors les alarmes qui correspondent à des défaillances persistent et par la suite nécessitent l'intervention des équipes de maintenance pour la remise en service du système de production.

Nous avons procédé à des simulations, en faisant varier les délais de retard,  $\Delta$  et nous obtenons alors les résultats illustrés par les figures 24.a, 24.b et 24.c dans les différents cas, sachant que nous avons pris les caractéristiques suivantes pour les machines :

$\varepsilon_i = 360$  pièces/heure;

$\delta_i = 20$  pièces/heure;

Stock initial = 30 pièces;

L'injection aléatoire des défauts suit une loi exponentielle ( $\lambda=0,02$ ), ce qui correspond à 20 défauts par 1000 unités de temps.

##### a- cas de tolérance ' $\Delta$ ' faible :

Comme le montre la figure 24.a, lorsque le retard toléré est faible, lors de l'exécution des opérations d'usinage, le système continue de fonctionner en basculant en mode dégradé puis revient au mode normal. Dans ce cas le nombre de fausses alarmes est très important par rapport au nombre de défaillances effectives. A titre d'indication, nous avons obtenu en choisissant  $\Delta = 0,01$  unité de temps distribué suivant une loi uniforme, une défaillance sur 20 défauts injectés.

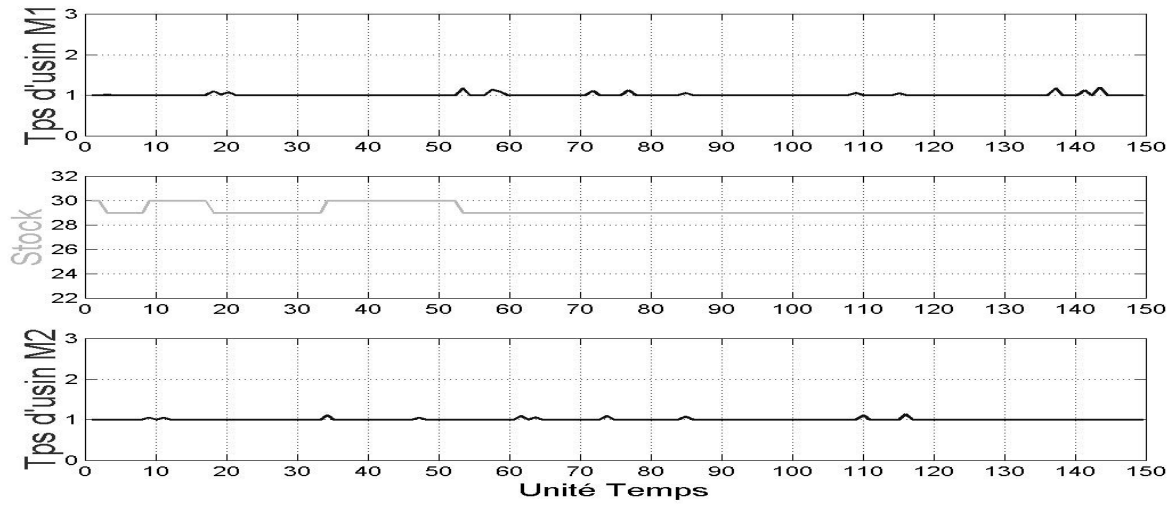
##### b- cas de tolérance ' $\Delta$ ' important :

Dans ce cas, comme représenté par la figure 24.b, lorsque le retard toléré lors de l'exécution des opérations d'usinage est assez élevé, nous ne pouvons détecter toutes les défaillances possibles. Nous ne pouvons pas tolérer de telle marge, pour la raison de sécurité et de sûreté de l'installation.

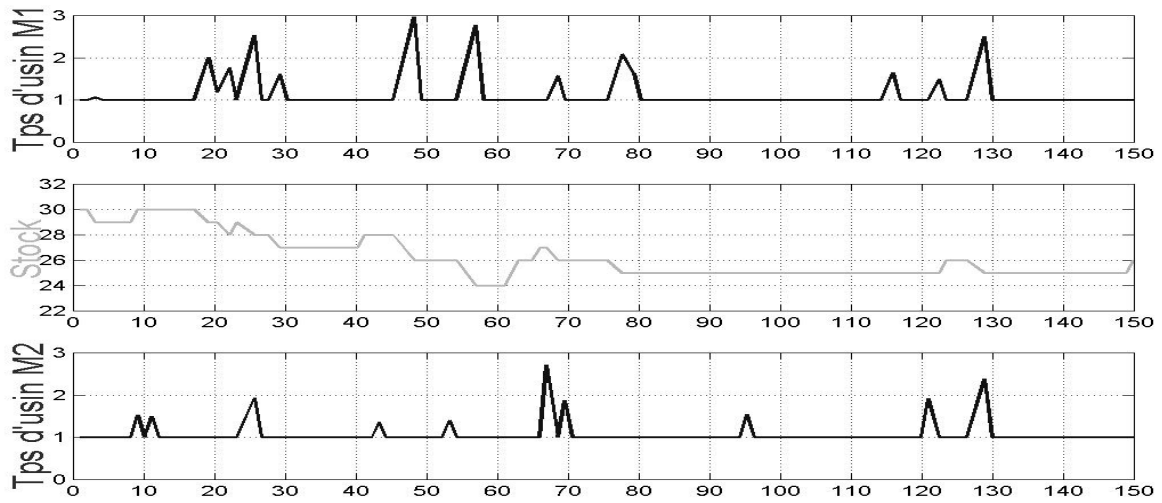
##### c-Conclusion :

La synthèse de cette optimisation, comme illustrée par la figure 24.c, nous permet de conclure que lorsque les valeurs du seuil pour chacune des machines sont choisies, alors tant que la tolérance dans les durées des opérations d'usinage ne dépasse pas ce seuil le système continue de fonctionner. Ces variations ne seront pas prises comme défaillances, ainsi seules les temps d'usinage supérieurs aux seuils seront détectés et correspondent alors à des défaillances du système. Nous obtenons dans ce cas moins de fausses alarmes.

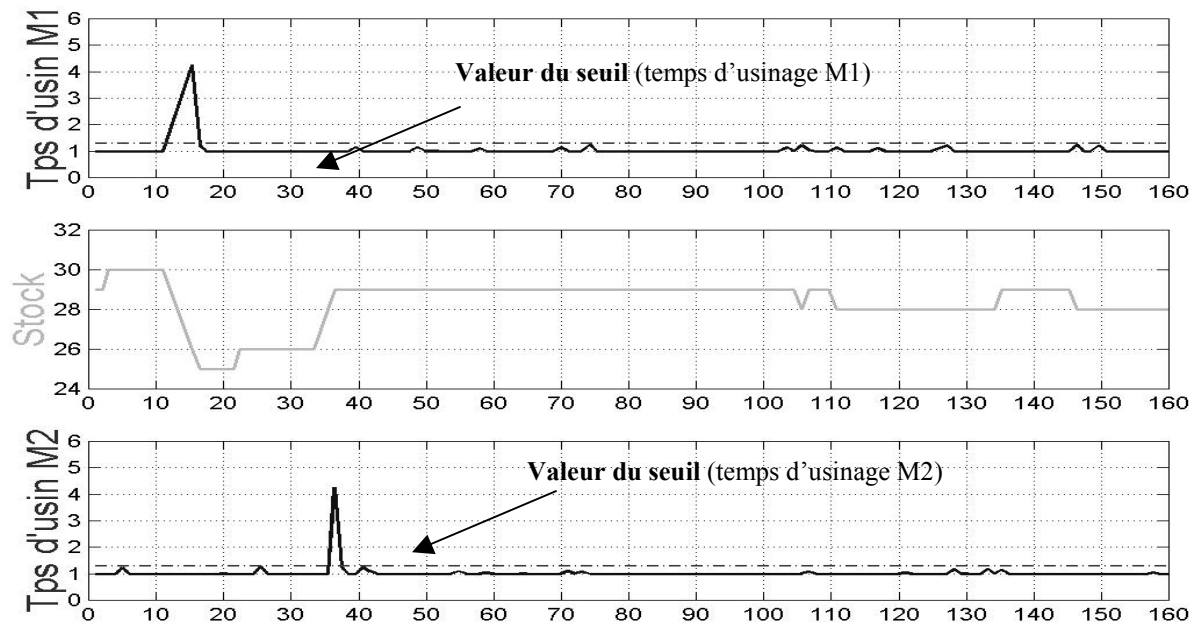
Notons que les valeurs des seuils ne correspondent pas à des valeurs optimales, ils sont donnés à titre indicatif pour des caractéristiques de machines données.



**a. Cas de tolérance de faible valeur.**



**b. Cas de tolérance importante.**



**c. Synthèse d'optimisation.**

**Fig. 24. Quelques résultats de simulation.**



### 2-3. Conclusion

Le premier résultat à retenir c'est l'existence de la relation de dualité entre le temps d'usinage et les valeurs des stocks intermédiaires. Ceci confirme la possibilité d'utiliser cette information pour connaître l'état de fonctionnement de la machine.

En conclusion, ces deux approches ne permettent pas de faire les opérations de surveillance que nous recherchons ; à savoir la détection des défaillances le plus rapidement possibles avec le minimum de fausses alarmes et le diagnostic qui revient à identifier l'organe qui est à l'origine de la défaillance. Alors surveiller le temps de service nous paraît la bonne approche, mais l'outil réseaux de Petri semble ne pas être l'outil approprié. Donc nous verrons dans la partie suivante l'utilisation de l'outil automate temporisé ; plus adapté au caractère temporel dont nous nous intéressons pour surveiller la durée d'exécution de chacune des tâches d'un système de production

Nous verrons dans le chapitre 3, la construction du modèle de surveillance qui tient compte du caractère temporel, pour la mise en œuvre du modèle dynamique du système à surveiller. Dans un premier temps, l'outil automate temporisé sera utilisé puis pour des raisons de minimiser l'explosion du nombre d'états et arcs, l'outil statecharts sera utilisé et exploité pour optimiser les temps de détection des défaillances.

## 3. MODÉLISATION PAR AUTOMATES TEMPORISÉS [RAY-03]

L'évolution d'un automate non temporisé est due à l'occurrence d'événements discrets (qui se traduit par le franchissement de transitions discrètes) grâce auxquels le procédé quitte l'état courant pour accéder à un nouvel état.

Quand un automate temporisé change d'état, le choix du nouvel état dépend à la fois de l'événement (symbole lu) et de la date d'occurrence par rapport aux dates d'occurrence des symboles précédemment lus. Dans ce but, Alur et Dill [ADL-94] ont associé à l'automate un ensemble fini d'horloges à valeurs réelles, le domaine temporel retenu est  $\mathbb{R}^+$ .

### 3-1. Introduction

Parmi les nombreuses extensions de la théorie de base initiée par Ramadge et Wonham [RAM-89], certaines approches concernent le temps : l'objectif a été (et est toujours) d'intégrer le temps dans le modèle et dans la théorie, objectif justifié par l'intérêt croissant porté aux systèmes temps réel depuis quelques années. Un SEDT est un système à événements discrets temporisé.

L'outil automate temporisé est défini comme étant une machine à états finis muni d'un ensemble de variables continues par morceaux appelées horloges [ADL-94][YOV-93]. Ces variables mesurent l'écoulement du temps. Lorsque le système séjourne dans un sommet, chaque horloge  $x_i$  est incrémentée uniformément avec le passage du temps. Les horloges sont

synchronisées et avancent avec le même pas. La dynamique d'une horloge  $x$  dans un sommet de l'automate temporisé est décrite par l'équation :  $\dot{x} = 1$ . A chaque sommet de l'automate on associe un prédicat sur la valeur des horloges appelé invariant du sommet. L'automate peut séjourner dans un sommet tant que l'invariant du sommet correspondant est vérifié par la valeur des horloges.

Le franchissement d'une transition est instantané. Il peut déterminer la mise à zéro de certaines horloges. Ce changement discret de la valeur des horloges  $x_i$  est modélisé par la relation  $x_i := 0$ . L'ensemble des horloges qui sont mises à zéro lors du franchissement d'une transition est décrit par une affectation. De plus à chaque transition on associe un prédicat sur la valeur des horloges, appelé garde. Ce prédicat indique l'intervalle temporel pendant lequel l'événement peut se produire (et va se produire) ou, plus exactement, l'intervalle temporel pendant lequel l'occurrence de l'événement va permettre une évolution de l'automate. Il s'agit donc des dates auxquelles la transition va pouvoir être franchie. La forme de garde la plus simple se traduit par une comparaison entre une horloge et une valeur appartenant à l'ensemble des rationnels.

### 3-2. Concepts de base

Nous illustrons les concepts de base, ainsi que la notation graphique du modèle automate temporisé à travers un exemple de modélisation d'un distributeur de boissons [OLI-94].

**Exemple 3.2.** Le distributeur de boissons que nous considérons permet de choisir du thé ou du café. Lorsque l'utilisateur introduit une pièce, il dispose de 10 secondes pour faire son choix. Si au bout de cette période il n'a rien choisi, le distributeur lui rend la pièce et redevient disponible. L'automate temporisé qui modélise le comportement de ce système est représenté dans la figure 25.

Cet automate temporisé est composé d'un ensemble de sommets  $\{ L_0, L_1 \}$  reliés par des transitions. Il est équipé d'une seule horloge,  $x$ , utilisée pour mesurer le temps écoulé depuis l'introduction d'une pièce. Lorsque le système séjourne dans un sommet de l'automate, la dynamique de cette horloge est décrite par l'équation :  $\dot{x} = 1$ .

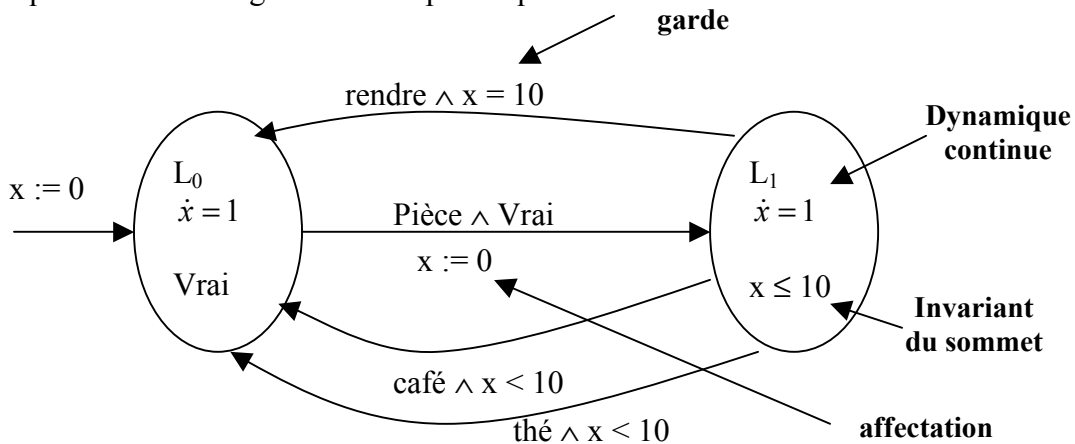


Fig. 25. Automate temporisé qui modélise le distributeur de boisson

Le sommet initial de l'automate est  $L_0$ . Il est marqué par une flèche entrante sur laquelle on marque la valeur initiale des horloges. Le sommet  $L_0$  modélise l'état de disponibilité du distributeur. Lorsque le système séjourne dans  $L_0$ , le distributeur attend que l'utilisateur introduise une pièce. Il n'y a aucune contrainte temporelle sur la durée d'attente d'une pièce. Ainsi le système peut séjourner dans ce sommet pendant un temps indéfini, donc l'invariant du sommet associé à  $L_0$  a la valeur « Vrai ».

Lorsqu'un utilisateur introduit une pièce, l'événement pièce se produit et le système franchit la transition allant de  $L_0$  vers  $L_1$ . Il n'y a aucune contrainte temporelle sur l'instant de dépôt d'une pièce par l'utilisateur. Par conséquent la garde associée à cette transition a la valeur « Vrai ». L'affectation  $x := 0$  associée à cette transition modélise l'initialisation de l'horloge  $x$  par son franchissement.

Lorsque le système se trouve dans le sommet  $L_1$ , le distributeur attend que l'utilisateur fasse son choix. La valeur de l'horloge  $x$  dans le sommet  $L_1$  est égale au temps écoulé depuis le dépôt de la pièce par l'utilisateur. L'invariant du sommet associé à  $L_1$  est  $x \leq 10$ . Ainsi le système doit quitter le sommet  $L_1$  au plus tard 10 secondes après y être arrivé. Lorsque  $x < 10$  seulement les transitions associées aux événements thé et café peuvent être franchies parce que toutes les deux ont la garde  $x < 10$ . lorsque  $x$  atteint la valeur 10, on franchit plutôt la transition associée à l'événement rendre, parce qu'elle a la garde  $x = 10$ . Et le cycle se répète lorsque le système retourne au sommet  $L_0$ .

### 3-3. Définitions

Avant de donner la définition formelle de l'outil automate temporisé, nous présentons quelques définitions concernant les horloges [ALU-99].

Soit  $X$  un ensemble fini d'horloges.

**Définition1.** Soit  $(x_1, x_2) \in X^2$ ,  $c \in \mathbf{Q}$  une constante rationnelle et  $\prec \in \{<, \leq, \geq\}$  une relation d'ordre. Une contrainte  $\rho$  sur la valeur des horloges est définie par une des expressions suivantes :

- $\rho := x_1 \prec c$ ,
- $\rho := x_1 - x_2 \prec c$ ,
- $\rho := \rho_1 \wedge \rho_2$ ,
- $\rho := \neg \rho_1$

Où  $\rho_1$  et  $\rho_2$  sont des contraintes sur les valeurs des horloges. □

L'ensemble des contraintes sur la valeur des horloges est noté  $G_x$ .

**Définition.2.** Une valuation des horloges est une fonction  $v : X \rightarrow \mathbb{R}^+$ , qui affecte un nombre réel positif à chaque horloge. □

Ainsi, une valuation des horloges définit la valeur de toutes les horloges à un instant donné. L'ensemble des valuations des horloges  $x \in X$  est noté  $V_x$ . La notation  $v \models \rho$  exprime le fait que la valuation  $v \in V_x$  vérifie la contrainte  $\rho \in G_x$ .

**Définition.3.** Une affectation, notée  $A_{m,n}$ , désigne l'ensemble des horloges qui seront mises à zéro par le franchissement d'une transition  $T_{m,n}$ . Elle est décrite par un ensemble d'équations de type  $x_i := 0$ . □

Soit  $v \in V_x$  une valuation,  $t \in \mathbb{R}^+$  un nombre réel et  $A_{m,n}$  une affectation. On note avec  $v + t$  la valuation qui affecte la valeur  $v(x) + t$  à chaque horloge  $x \in X$ .

De même, on note avec  $v[A_{m,n}]$  la valuation qui met à zéro les horloges spécifiées par l'affectation  $A_{m,n}$  et laisse inchangée la valeur des autres horloges.

Formellement, un automate temporisé est défini par les définitions 4 et 5 données ci-après [ALU-99][ADL-94] [YOV-93][GOU-99] :

**Définition 4.** Un automate temporisé est un 6-uplet  $\mathcal{A} = (\mathcal{L}, L_0, L_1, X, \Sigma, I, \mathcal{T})$ , où :

- ..  $\mathcal{L}$  est l'ensemble fini de sommets ;
- ..  $L_0 \subset \mathcal{L}$  est le sommet initial ;
- ..  $X$  est l'ensemble fini des horloges ;
- ..  $\Sigma$  est un ensemble de symboles ;
- ..  $I$  est une application qui associe un invariant du sommet  $I(L_m)$  à chaque sommet  $L_m \subset \mathcal{L}$ .
- ..  $\mathcal{T}$  est l'ensemble des transitions. Sachant qu'une transition est un 5-uplet  $(L_m, \mathbf{a}, \mathbf{g}_{m,m+1}, A_{m,m+1}, L_{m+1})$ , où :

- $L_m$  est le sommet source ;
- $\mathbf{a} \in \Sigma$  est un symbole associé à un événement ;
- $\mathbf{g}_{m,m+1}$  est la condition de franchissement, i.e. la garde ;
- $A_{m,m+1}$  est l'affectation ;
- $L_{m+1}$  est le sommet destination. □

**Définition 5.** L'état d'un automate temporisé est défini par le couple  $(L_m, v)$ , où  $L_m$  désigne le sommet et  $v$  est une valuation d'horloges qui vérifie l'invariant du sommet  $I(L_m)$ . On peut séjourner dans un sommet de l'automate temporisé tant que l'invariant associé est satisfait par la valeur des horloges. Un sommet, peut donc ne pas correspondre qu'à un seul état, mais à un espace d'état. Les valeurs que les horloges peuvent prendre pendant le séjour du système dans un sommet décrivent un espace d'états des horloges. □

### 3-4. Modèles de surveillance

Reprenons la structure globale, que nous avons proposé précédemment dans la figure 18, pour l'intégration des outils de supervision, de surveillance et d'aide à la maintenance des systèmes de production manufacturiers. Nous nous limitons au module de surveillance, objet du travail que nous présentons, nous aurons à modéliser les trois modes de fonctionnement représentés dans la figure 19. Nous nous sommes intéressés plus exactement au module de surveillance, c'est à dire aux fonctions de détection et diagnostic. Nous vous rappelons que le modèle de surveillance est basé sur les seuls états, en temps réel, correspondant aux capteurs du procédé.

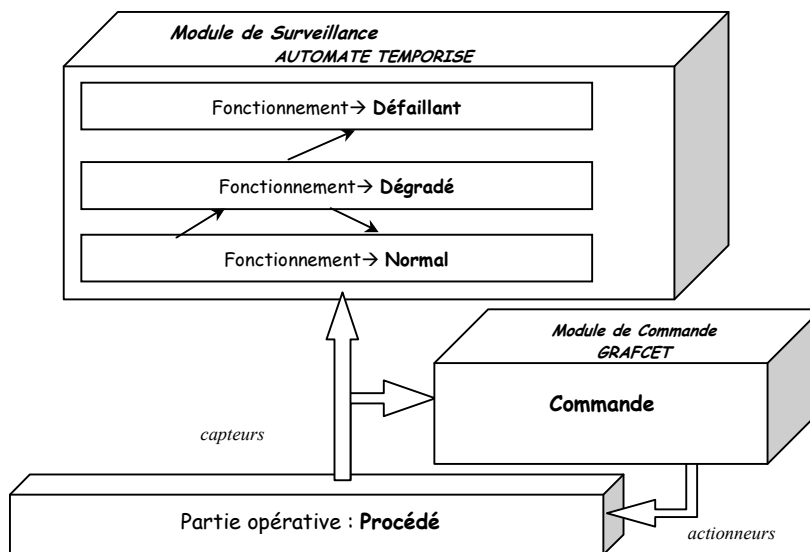


Fig. 26. *Structure proposée du module de surveillance.*

Le système de production peut évoluer entre trois états différents; il est alors soit en fonctionnement normal si les durées d'activation des tâches sont respectées (à l'intérieur de l'intervalle de bon fonctionnement, noté  $I_m$ , relative à l'horloge  $x_m$ ), soit en fonctionnement dégradé si le retard d'arrivée de l'événements fin de tâche est compris dans un intervalle de tolérance (noté  $J_m$ , relative à l'horloge  $x_m$ ), soit en état de défaillance si ces conditions ne sont pas respectées comme le montre la figure 26.

Le système oscille entre le mode de fonctionnement normal et dégradé. Du point de vu modélisation ces deux modes de fonctionnement sont similaires (on retrouve les mêmes états de fonctionnement dans les deux cas), ils ne se différencient que par la tolérance associée à la durée des tâches qui reste à estimer. L'état de défaillance quant à lui est un état absorbant qui nécessite l'intervention de l'équipe de maintenance.

### 3-5. Les modes de fonctionnement pris en compte [RAY-03], [SIM-03]

Reprenons l'exemple de la figure 15, nous remarquons que la machine peut séjourner dans l'un des 3 états : soit à l'arrêt (attente de pièce), soit en cours de distribution de la boisson sélectionnée ou dans l'état défaillant. La remise en état de l'équipement suite à une défaillance, nécessite, dans la plupart des cas, sinon souvent, l'intervention des équipes de maintenance.

L'objectif est d'identifier au plutôt la défaillance et faire appel à la maintenance au bon moment. Il est nécessaire de réagir ni trop tôt pour éviter les fausses alarmes, ni trop tard pour limiter les non-détections. Pour cela, le mode de fonctionnement intermédiaire, avant l'apparition de la défaillance, doit être identifié : c'est le mode dégradé. Une bonne modélisation de cet état permet de réagir efficacement face à une défaillance signalée.

Il nous paraît très utile, de ne pas pénaliser la production, à chaque fois que la situation ne nécessite pas l'arrêt. Si nous reprenons la contrainte temporelle comme nous avons présenté précédemment, il nous semble pratique de prévoir un état intermédiaire entre l'état de fonctionnement normal et l'état de défaillance, en prévoyant un intervalle dit « de tolérance » situé chronologiquement immédiatement après l'intervalle de bon fonctionnement.

#### 3.5-1. Critères d'évaluation

Le temps d'activité d'une tâche associée à un état est propre à une situation donnée. De ce fait on dira que le système est en mode normal, si la durée d'activation de l'état est dans l'intervalle noté  $I_m$ , il est en mode dégradé si la durée d'activation de l'état est dans l'intervalle noté :  $J_m$  ( $I_m \neq J_m$ ). Si la durée d'activation dépasse la borne  $T_c^m$ , le système est considéré comme défaillant. Comme représenté sur la figure 27.

Pour chaque tâche, nous définissons trois valeurs de temps, de la manière suivante :

- Le temps minimum nécessaire pour l'exécution correcte de la tâche  $m$ , noté :  $T_{\min}^m$ .
- Le temps maximum toléré pour l'exécution de la tâche  $m$ , noté :  $T_{\max}^m$ .
- Le temps critique au-delà duquel, il n'y a aucun doute de la présence d'un problème, noté :  $T_c^m$ .

Comme cela, on définit les intervalles de fonctionnement normal noté  $I_m = [T_{\min}^m, T_{\max}^m]$ , celui du fonctionnement dégradé noté  $J_m = ]T_{\max}^m, T_c^m]$  et celui du mode défaillant noté  $]T_c^m, +\infty[$ .

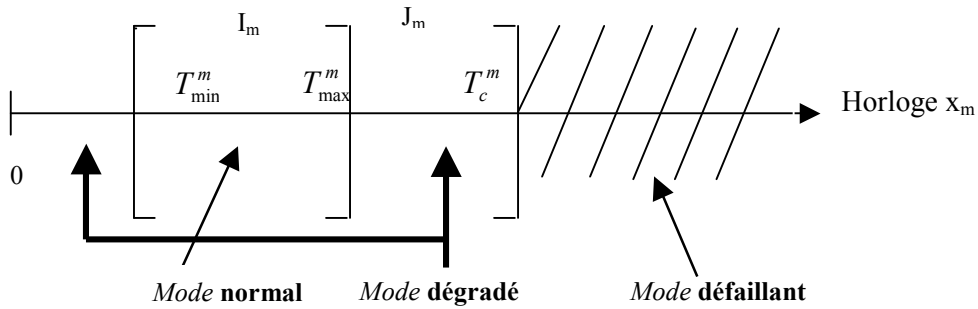


Fig. 27. *Durée d'activation et modes de fonctionnement.*

Nous verrons avec plus de détail dans les parties suivantes, la procédure d'obtention de ces intervalles. Avec cette solution, dès qu'il y a l'apparition d'un dysfonctionnement, le système de production bascule en mode dégradé. Si l'événement suivant

se produit pendant son intervalle de bon fonctionnement, alors le système redevient au mode normal. Sinon, il y reste tant que les événements se produisent pendant leurs intervalles de tolérance, dès qu'un événement se produit au-delà de la limite admissible (borne supérieure de l'intervalle de tolérance), le système bascule vers le mode défaillant comme le montre la figure 27. Notons que la tâche  $m'$  est ultérieure à  $m$ , et que la tâche  $m''$  suivant le cas, peut être la tâche  $m$  qui est la cause du basculement du système en mode dégradé ou bien d'une tâche quelconque ayant dépasser l'intervalle toléré (fonctionnement en mode dégradé).

A partir de l'exécution d'une tâche  $m$ , deux cas peuvent se produire : (voir figure 27)

- $m' \equiv m+1$  ; lorsque la tâche  $m'$  est exactement la tâche qui vient de s'exécuter à la suite de la tâche  $m$ , nous dirons dans ce cas qu'il s'agit d'une **fausse alarme**.
- $m'' \equiv m$  ; lorsque la tâche  $m$  qui a causé le basculement du mode de fonctionnement normal vers le mode dégradé est aussi la cause du passage en mode défaillant, nous dirons dans ce cas qu'il s'agit effectivement d'une **vraie défaillance détectée**.

### 3.5-2. Construction du modèle dynamique

Pour construire le modèle dynamique du système, on procède de la manière suivante : à partir du modèle de commande, on construit le graphe des situations qui représente toutes les évolutions possibles du système. A chaque état de ce graphe est associé une ou plusieurs horloges ; Chacune surveille le temps de service d'une activité en cours d'exécution, ces horloges sont toujours initialisées au début du service et ont une dynamique continue constante  $\forall i$ , Enfin, à partir de l'ensemble des états, et tenant compte des horloges de contrôle d'activité, nous construisons le modèle dynamique du système utilisant l'outil automate temporisé.

Dans le modèle dynamique, on retrouve les horloges actives dans chaque sommet, qui correspond à un état du système, et les gardes qui correspondent à l'évolution normale du système ; c'est à dire que l'occurrence « fin de service » arrive pendant l'intervalle de bon fonctionnement noté  $I_m$ . Cette démarche de modélisation est illustrée dans les exemples d'application présentés dans le chapitre 3, paragraphe 2.

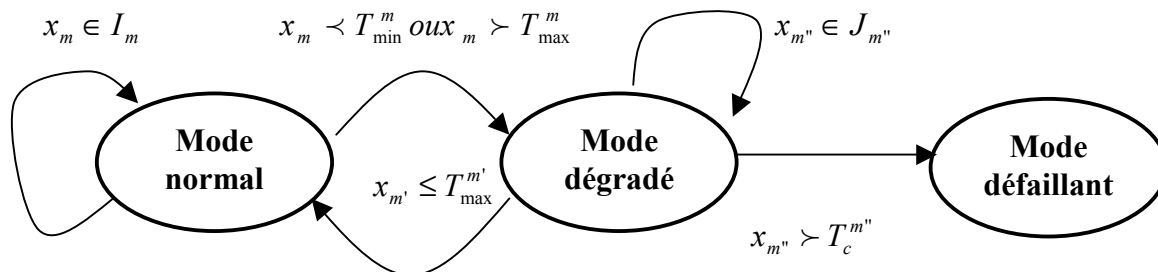


Fig. 28. Les modes de fonctionnement utilisés.

Plusieurs méthodes sont utilisées pour la détection des défaillances. Dans les systèmes à événements discrets, et comme nous l'avons vu dans les approches de surveillance existantes, toutes les méthodes sont basées sur la comparaison, à chaque instant, entre l'état réel du procédé ; donnée par l'ensemble des états capteurs, et celui donné par le modèle de comportement normal du système.

**Hypothèse :** On suppose que la commande du système à surveiller est exempte de défaillance. Donc, le modèle de comportement sera obtenu à partir de l'ensemble des évolutions possibles du système de commande ; ainsi toute déviation par rapport à ce comportement sera détectée et traitée comme situation anormale du système.

Nous construisons alors le modèle dynamique du système, à partir du graphe des états obtenu comme nous le verrons dans les exemples d'application. Ensuite, nous définissons pour chaque tâche du procédé, l'intervalle de fonctionnement normal, l'intervalle de fonctionnement dégradé et le seuil au-delà duquel nous estimons que le fonctionnement est défaillant.

Donc, après chaque ordre d'exécution d'une tâche, nous attendons l'information de fin d'exécution de celle-ci, et en fonction du temps écoulé entre ces deux instants par rapport aux intervalles spécifiés précédemment, nous dirons que le fonctionnement est normal, dégradé ou défaillant.

Tant que  $x_i$  appartient à l'intervalle de bon fonctionnement, le module de surveillance indique que le système est en mode normal. Dès que  $x_i$ , pour une tâche donnée, appartient à l'intervalle de fonctionnement dégradé, le module de surveillance bascule en mode dégradé. Il y reste jusqu'à ce que une des tâches suivantes soit achevée pendant une durée appartenant à son intervalle de bon fonctionnement.



A chaque fin de tâche, l'algorithme suivant est vérifié :

- Si  $x_i$  appartient à l'intervalle de bon fonctionnement alors le système est en mode normal.
- Sinon si  $x_i$  appartient à l'intervalle de fonctionnement dégradé alors le système est en mode dégradé.
- Sinon le système est en mode défaillant.

### 3.5-3. Conclusion

Nous allons nous rendre compte, lorsque nous traiterons le premier exemple dans la partie suivante de notre mémoire, que l'outil automate temporisé ; parfaitement adaptés aux systèmes à événement discrets temporisés, présente certaines limitations qui justifie la recherche d'un outil plus adapté à la surveillance de ce type de système.

Les problèmes posés par la modélisation avec l'outil automate temporisé sont les suivants :

- Diagrammes « plats », pas de hiérarchie ni de modularité.
- Grand nombre de transitions correspondant à un même événement.
- Explosion du nombre d'états dès que le système croit.

Pour remédier à ces problèmes, un 2<sup>ème</sup> outil de modélisation mieux adapté pour la modélisation de notre système de surveillance, sera présenté dans la partie suivante. Une comparaison indicative est faite sur la figure 29 suivante :

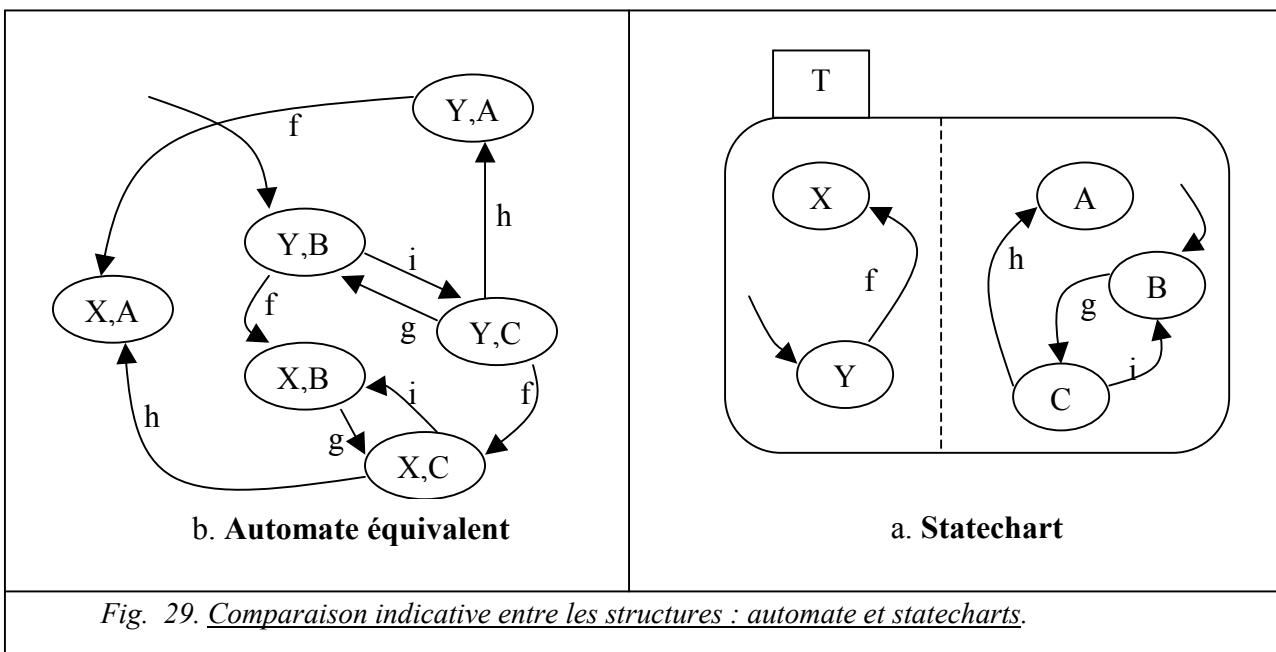


Fig. 29. Comparaison indicative entre les structures : automate et statecharts.

## 4. MODÉLISATION PAR STATEFLOW

Stateflow est un outil de description qu'on peut comparer au grafcet, lequel permet de décrire, sous une forme normalisée la logique du comportement des systèmes à événements discrets. Stateflow fonctionne en symbiose avec simulink. Il est donc possible de modéliser un processus industriel et de simuler l'enchaînement des tâches en même temps qu'elles s'exécutent.

Le comportement d'un processus physique automatisé est décrit dans un document qui définit les **étapes** du fonctionnement et leur enchaînement. Il précise les **conditions** de passage d'une étape à l'autre et les **actions** qu'elles doivent exécuter. Ce descriptif, en forme de cahier des charges, se présente en principe sous une forme normalisée dont le grafcet est exemple élaboré.

Stateflow représente lui aussi la partie « commande » d'un ensemble automatisé par un diagramme état-transition ce que les utilisateurs du grafcet devraient apprécier. Il permet de simuler le fonctionnement du diagramme. Il est donc possible en utilisant Simulink pour la partie « opérative » et Stateflow pour la partie « commande » de simuler complètement une machine automatisée. Stateflow pourra même assurer la génération et l'implantation du code exécutable dans un organe de traitement (calculateur, automate programmable, ...).

### 4-1. Généralités

On construit tout d'abord le schéma en plaçant, grâce à un éditeur, les différents éléments du diagramme appelé « Chart » à construire et qui sont essentiellement :

- Les **états** (state).
- Leurs **liaisons** (transitions).
- Ainsi que le **libellés** attribués aux **événements** ( events ) et aux **actions**.

Cette représentation graphique recouvre en fait la construction d'un programme MATLAB selon la syntaxe d'une S-fonction.

#### 4.1-1. La terminologie : une conception hiérarchisée

Un schéma **stateflow** doit être conçu de manière hiérarchisée. Les objets graphiques et non graphiques vont s'emboîter selon la hiérarchie parent-enfants.

Le plus haut niveau de la hiérarchie **stateflow** est appelé « machine » ; il correspond au niveau « modèle » dans la hiérarchie. Ce vocable est issu du concept de **machine à états finis** qui a été développé ( Moore ou Mealy ) pour rendre compte du fonctionnement des systèmes à

événements discrets. Dans ces systèmes, le passage d'un **état** à un autre est gouverné par des événements discrets, alors que dans les systèmes continus ( ou échantillonnés ) le nombre des états est infini et que le passage d'un état au suivant se fait au fil du temps. La machine peut réunir **plusieurs diagrammes stateflow** dont elle est le parent.

**Le diagramme** (Chart): Ce deuxième niveau, qui correspond au sous système Simulink, décrit une partie de la machine. Il rassemble des **états** reliés par des **transitions** dont il est le parent. Le diagramme décrit le fonctionnement logique d'un sous-ensemble particulier de la **machine**. Une machine comprend bien souvent plusieurs diagrammes qui, dans la plupart des cas, communiquent entre eux, mais elle peut ne comprendre qu'un seul diagramme.

**L'état** (State): Ce niveau correspond à l'état qu'on décrit par un vecteur d'état dans Simulink. Il représente l'**état** dans lequel se trouve la machine, en quelques sorte l'état actuel de son fonctionnement, par exemple l'état de **repos**. Cette notion intuitive mais peu vague sera précisée ultérieurement.

Un **état** peut contenir des **sous-états** qui sont ses enfants, il devient alors un super-état. Par exemple l'état **marche** d'une machine peut contenir deux sous-états, **avant** et **arrière**. L'état **avant** peut lui aussi avoir des enfants : **rapide**, **lent** et **attente**. Il n'y a pas de restrictions au nombre total de générations dans la descendance d'un parent.

**L'événement** (event): Cet objet non graphique (il n'apparaît sur le diagramme que grâce à son libellé). L'événement joue un rôle majeur dans **stateflow**. C'est lui qui réveille un diagramme qui « s'est endormi » en l'absence d'événements. C'est lui qui va déclencher le franchissement d'une transition particulière (si elle est franchissable) d'un **état** vers l'**état** suivant et assurer l'exécution des actions. Comme tout objet il a un parent ( acquit lors de sa déclaration) et il n'est actif qu'à ce niveau et dans sa descendance.

Il existe d'autres objets, graphiques ou non, que nous verrons ultérieurement.

#### 4.1-2. Présentation d'un exemple "illustratif"

Afin de permettre la présentation des principes et des règles, nécessaire pour la mise au point d'un diagramme sous stateflow, nous présentons l'exemple de mode de marche/arrêt d'un chauffe-eau ; le système bascule entre deux états. Il est en état de marche noté « M », lorsque l'interrupteur est en position de marche et que la température de l'eau est inférieure à 30 degrés centigrades. Sinon il reste dans l'état arrêt noté « A » tant que la température est supérieure à 30 degrés centigrades. A partir de l'état de marche, le système peut revenir à l'arrêt lorsque l'interrupteur passe en position arrêt. Ceci est illustré dans la figure 30 suivante.

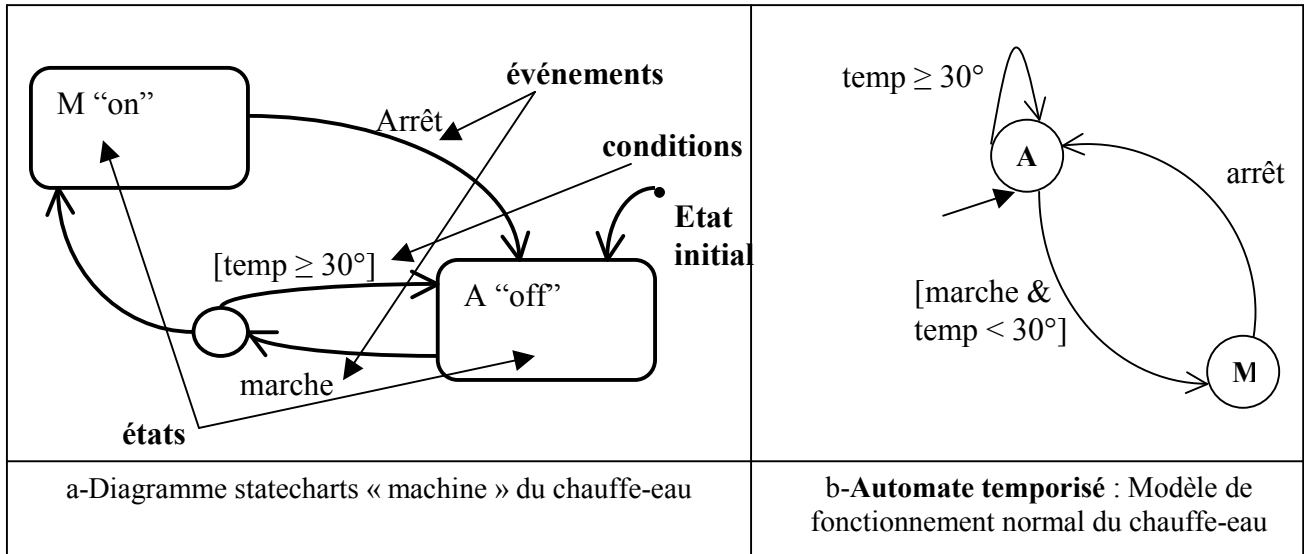


Fig. 30. *Modèle de fonctionnement du chauffe-eau*

Les avantages de cet outil résident dans le fait qu'il permet de prendre en compte les aspects de parallélisme et d'hierarchie de façon naturelle, chose difficile à gérer avec les automates qui imposent la séquentialité. Ceci permet aussi de réduire le nombre d'arcs entre les différents états du système, lorsqu'il s'agit d'une entrée vers un état composé (état parent contenant des sous-états) et aussi lors d'une transition d'un état composé vers un autre. Nous verrons ceci en détails dans l'exemple d'application ainsi que les différentes règles d'exécution et les principes d'évolution au cours du fonctionnement en temps réel.

## CHAPITRE 3. MODÈLE DE SURVEILLANCE

*Dans cette partie, nous présentons le modèle de surveillance d'un système, type manufacturier, basée sur le suivi en temps réel des temps de service. L'objectif principal est de détecter et localiser les défaillances qui peuvent affecter les performances du système en termes de productivité et de sûreté. Une procédure de surveillance a été développée. Une modélisation qui regroupe l'ensemble des états du système par automates temporisés permet de prendre en compte ses différents modes de fonctionnement. Cette approche permet ainsi d'améliorer les paramètres qui caractérisent la qualité et l'efficacité d'une procédure de surveillance, à savoir les retards de détection et les fausses alarmes.*

*Nous présentons alors, en détail, la démarche pour la mise en œuvre du modèle de surveillance du système. L'objectif principal est l'optimisation du temps de détection. Le modèle de surveillance proposé est basé sur l'utilisation combinée des automates temporisés et des statecharts. Ils permettent d'éviter l'énumération de l'ensemble des états possibles du système et donc la maîtrise de la combinatoire des états. L'application à l'atelier de traitement de surface qui inclut les opérations de transport et les opérations de traitement, permet de mettre en évidence la puissance de description de ces outils.*

## 1. DÉMARCHE GÉNÉRALE DE MISE EN ŒUVRE D'UN SYSTÈME DE SURVEILLANCE

Nous rappelons que les procédés sont déjà commandés, nous nous intéressons seulement à l'aspect surveillance du fonctionnement du système. La démarche commence par la mise en œuvre du modèle dynamique lorsqu'il s'agit d'un procédé séquentiel ou des modèles dynamiques pour chaque activité, lorsqu'il s'agit d'un procédé faisant intervenir plusieurs activités en parallèle. Ces modèles dynamiques s'obtiennent facilement à partir du modèle de commande en y intégrant le caractère temporel ; Point fort de notre démarche. Le modèle de commande peut être sous forme de grafcet, comme nous le verrons dans le premier exemple ou bien, sous la forme d'un ordonnancement de tâches, comme nous le verrons dans le deuxième exemple. La figure 31 illustre les différentes étapes de la démarche.

Le modèle de surveillance, dans les deux cas, est obtenu par l'exploitation directe des différents modèles liés au système de commande.

### 1-1. La démarche : mise au point des modèles

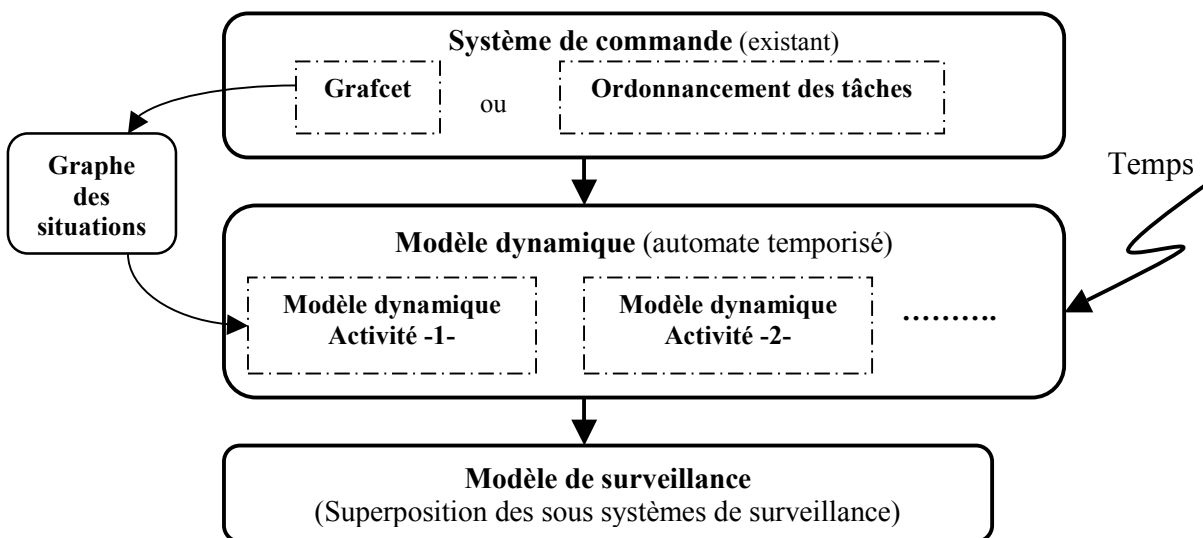


Fig. 31. Les étapes de la démarche de construction du modèle de surveillance.

Comme le montre la figure 31, A partir du ou des modèles dynamiques, nous construisons le modèle de surveillance global. Ce dernier est obtenu sur la base du principe des trois modes de fonctionnement représenté par la figure 26 du chapitre2, §3 et tenant compte du critère d'évaluation représenté par la figure 27 du même chapitre.

## 1-2. Prise en compte des trois modes de fonctionnement

Pour obtenir l'ensemble des états du système dans les trois modes de fonctionnement, nous procédons de la manière suivante :

- Le modèle de fonctionnement en mode dégradé sera alors une copie (analogue) du modèle de fonctionnement en mode normal (même ensemble états-arcs) seuls les délais seront prolongés jusqu'à la borne supérieure de l'intervalle  $J_m$ , notée  $T_c^m$ . Alors que pour le modèle de fonctionnement en mode défaillant, nous le considérerons, dans un premier temps, comme un état global de défaillance ; puisque nous ne nous intéressons pas à la localisation des défaillances pour le moment. Donc à chaque fois que le temps d'activation d'une étape  $m$  dépasse la borne supérieure notée  $T_c^m$ , le système de production bascule dans le mode de défaillance.
- Le système peut revenir dans le mode normal, depuis un état en mode dégradé, si la durée d'exécution de l'une des tâches suivantes est comprise dans l'intervalle du fonctionnement normal.
- Enfin, tenant compte de ces conditions de changement de mode, deux cas de figure se présentent dans la construction du modèle de surveillance comme le montre la figure 32.

A partir de chaque **état  $i$  en mode normal**, le système peut évoluer vers l'état suivant en mode normal ou vers l'état analogue en mode dégradé. Le système peut aussi atteindre cet état soit à partir de l'état précédent en mode normal, soit à partir d'un état en mode dégradé lorsque le système revient en mode normal depuis le mode dégradé.

A partir de chaque **état  $j$  en mode dégradé**, le système peut basculer, soit vers l'état suivant en mode normal, soit vers l'état suivant en mode dégradé, soit passer en mode défaillant lorsque le temps dépasse la valeur critique. Ce même état  $j$  peut être atteint, soit à partir de l'état analogue en mode normal ; si le temps dépasse la valeur  $T_m$  correspondante, soit à partir de l'état précédent en mode dégradé.

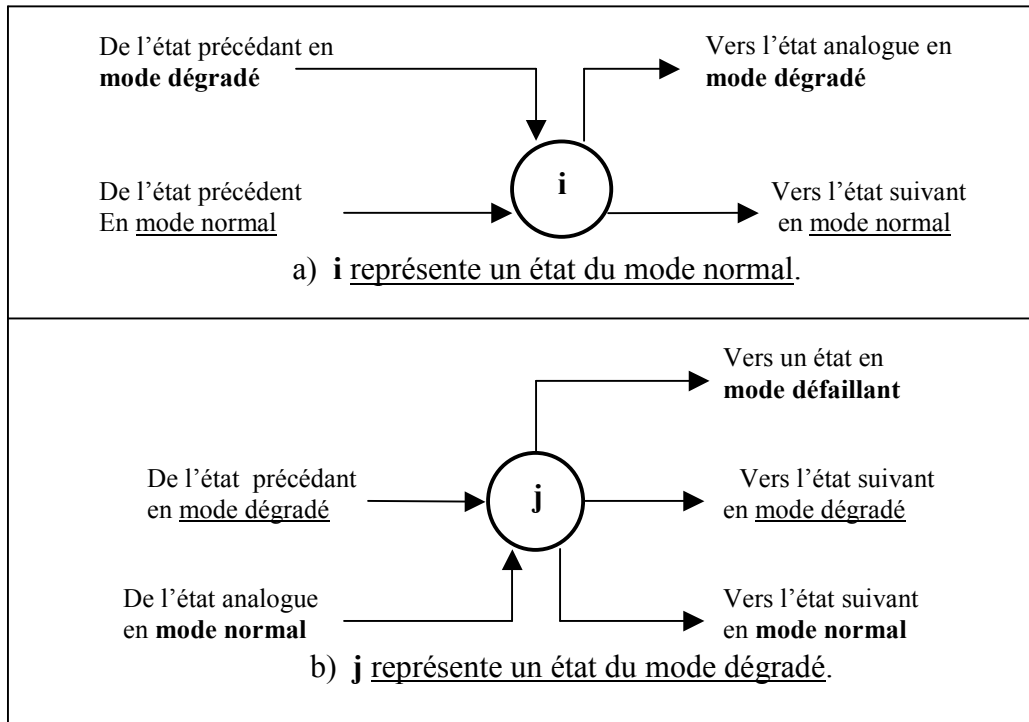


Fig. 32. *Passage entre les trois modes de fonctionnement.*

### 1-3. Mise en œuvre du modèle de surveillance

Le système de surveillance global sera composé d'un ou plusieurs sous systèmes de surveillance qui fonctionnent en permanence et en parallèle, comme le montre la figure 25, dans le cas de l'atelier de traitement de surfaces nous avons les sous systèmes suivants :

- **Surveillance de la partie opérative** : états des capteurs et les ordres d'opérateurs.
- **Surveillance des activités** : les durées des différentes tâches.

Dans les deux sous systèmes, les trois modes de fonctionnement (fonctionnement normal, dégradé et défaillant) sont pris en compte dans la modélisation du système de surveillance global comme le montre la figure 33.



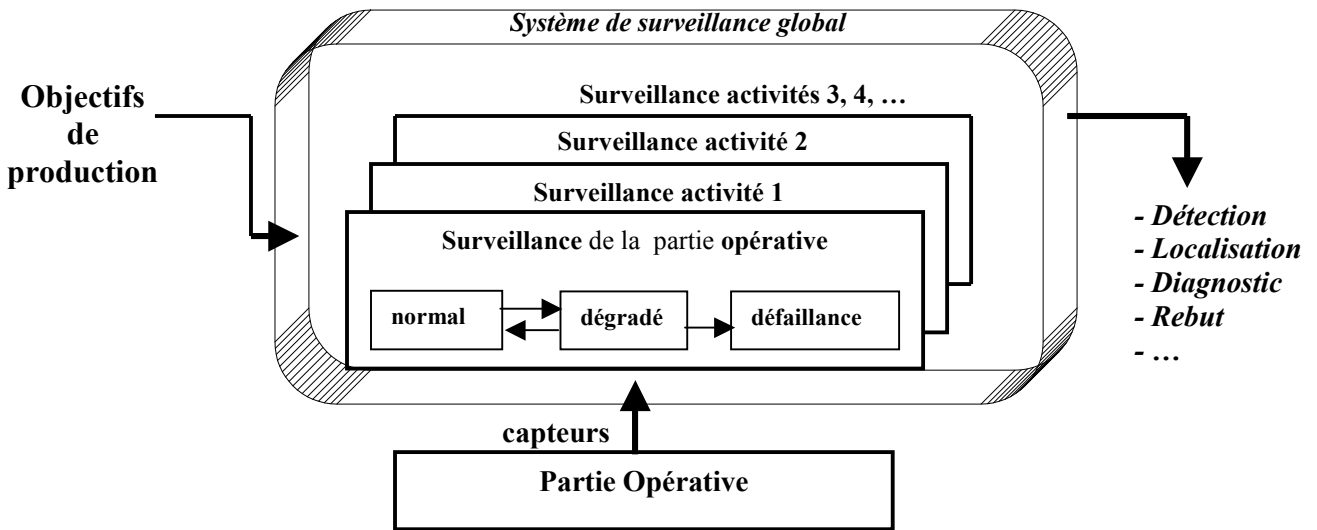


Fig. 33. *Modèle global de surveillance.*

## 2. SURVEILLANCE DU SYSTÈME : REMPLISSAGE DES BACS

À partir du grafcet de commande, nous construisons le graphe des situations. Puis le modèle dynamique sera mis en œuvre, en appliquant la démarche présentée précédemment, pour construire le modèle automate temporisé correspondant. Une fois ce modèle obtenu, en tenant compte des trois modes de fonctionnement, nous construisons le modèle de surveillance du système, comme nous allons le voir dans ce paragraphe.

On considère le système de remplissage de bacs de la figure 34. Il est composé d'un réservoir et de deux bacs. Les deux bacs sont utilisés de façon similaire. Le bac 1 est vide lorsque le niveau est au-dessous de  $b_1$  (i.e.  $b_1 = 0$ ). Il est plein lorsque le niveau est au-dessus de  $h_1$  (i.e.  $h_1 = 1$ ).

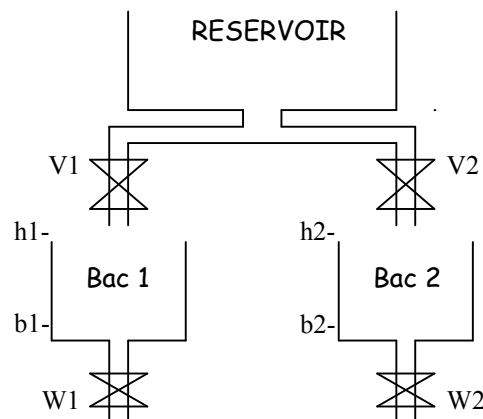


Fig. 34. *Exemple d'application.1 : Remplissage des bacs.*

A l'état initial, les deux bacs sont vides. Au moment où le contact  $m$  se ferme, les deux bacs se remplissent grâce à l'ouverture des vannes  $V_1$  et  $V_2$ . Dès qu'un bac est plein, on arrête son remplissage (fermeture de la vanne correspondante  $V_i$ ) et l'on commence à utiliser son contenu (ouverture de la vanne correspondante  $W_i$ ). Lorsque le bac est vide, on ferme la vanne  $W_i$ . Le remplissage ne pourra recommencer que lorsque les deux bacs seront vides.

Nous rappelons que le système que nous nous proposons de surveiller est déjà commandé. Le figure 35 représente le grafcet de commande de ce système. A partir de ce grafcet, nous construisons l'ensemble des états possibles du système. Cet ensemble est décrit par le graphe des situations représenté par la figure 36.

Notre objectif est de surveiller le bon fonctionnement de ce système représenté dans la figure ci-dessus, pour cela le système de surveillance que nous allons mettre en œuvre doit pouvoir détecter d'une part, les possibilités de débordement au niveau des bacs 1 et 2, d'autre part détecter la baisse du niveau dans chacun des bacs au-dessous des niveaux minimums  $b_1$  et  $b_2$ . Suivant l'environnement de l'installation l'un ou l'autre des cas pourrait être dangereux, voir catastrophique parfois. Nous disposons, pour chacun des bacs, des durées moyennes de remplissage et de vidange.

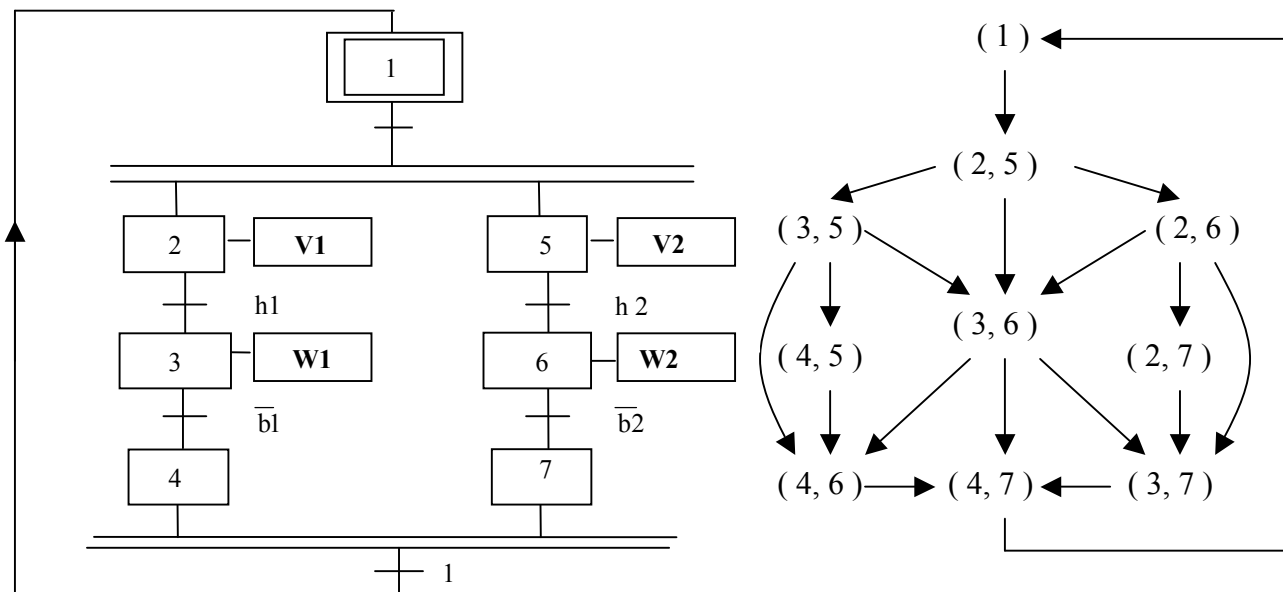


Fig. 35. Grafcet de commande du système de remplissage des bacs.

Fig. 36. Graphe de situations

A partir du grafcet de commande représenté par la figure 35, nous obtenons le graphe des situations de la manière suivante :

Depuis l'état initial noté (1), qui correspond à la situation étape\_1 active du grafcet, la seule évolution possible c'est vers l'état noté (2,5), qui correspond à la situation étape\_2 et étape\_5 du grafcet actives (V1 et V2 ouvertes). Depuis cet état, nous avons 3 évolutions possibles : soit vers l'état noté (2,6) lorsque le bac\_2 est plein (i.e.  $h_2=1$ ) donc la vidange de celui-ci commence, soit vers l'état noté (3,5) lorsque le bac\_1 est plein (i.e.  $h_1=1$ ) et sa vidange qui commence, soit vers l'état noté (3,6) lorsque les deux bacs sont remplis en même temps et leur vidange commence pour les deux. Ceci correspond à la situation, au niveau du grafcet, lorsque les étapes 3 et 6 sont actives toutes les deux. Ainsi nous obtenons le graphe représenté sur la figure 36.

## 2-1. Construction du modèle dynamique du système

Pour construire le modèle dynamique du système, nous partons du graphe des situations, puis nous associons à chaque état du graphe le nombre d'horloges nécessaires pour surveiller les durées des activités en cours. Dans le cas de cet exemple, il s'agit d'une ou deux activités simultanées ; donc une ou deux horloges associées. Notons que lorsqu'on passe d'un état à un autre, si une activité n'a pas commencé ni finie, l'horloge associée n'est pas affectée. Les horloges étant initialisées au début de l'activité à surveiller ; Par exemple à l'état initial noté (1) on initialise l'horloge  $x_1$  qui surveille le délai d'attente pour la validation de  $m$  par l'opérateur (Début du cycle). Puis à l'état (2, 5), on initialise les horloges  $x_2$  et  $x_5$  qui surveillent respectivement les durées d'activités des vannes V1 et V2 ( sur le grafcet,  $x_2$  représente la durée d'activation de l'étape 2 et  $x_5$  celle de l'étape 5).

Le temps  $T_2$  (respectivement  $T_5$ ) représente la durée nécessaire pour le remplissage du bac\_1 (respectivement bac\_2) en fonctionnement normal, cette durée est surveillée par l'horloge  $x_2$  (respectivement  $x_5$ ). Il en est de même que pour la vidange des deux bacs. Donc en fonctionnement normal, l'événement  $h_1$  (respectivement  $h_2$ ) doit arriver avant l'écoulement de la durée  $T_2$  par l'horloge  $x_2$  (respectivement  $T_5$  par l'horloge  $x_5$ ). Sachant qu'on se fixe un délai au plutôt pour l'arrivée d'un événement fin d'activité, ceci est représenté par l'équation  $x_j \in I_j$  qui figure sur les différents arcs du modèle.

### Remarque :

l'automate représenté tient compte du fait que :

- l'occurrence de deux événements indépendants n'est pas possible.
- la situation (4,7) qui correspond à un état d'attente, étant instable, les états (1) et (4,7) seront fusionnés.

Une fois cette procédure appliquée à tous les états du graphe des situations, nous obtenons alors le modèle dynamique suivant du système des bacs à surveiller voir figure 37 ci-dessous.

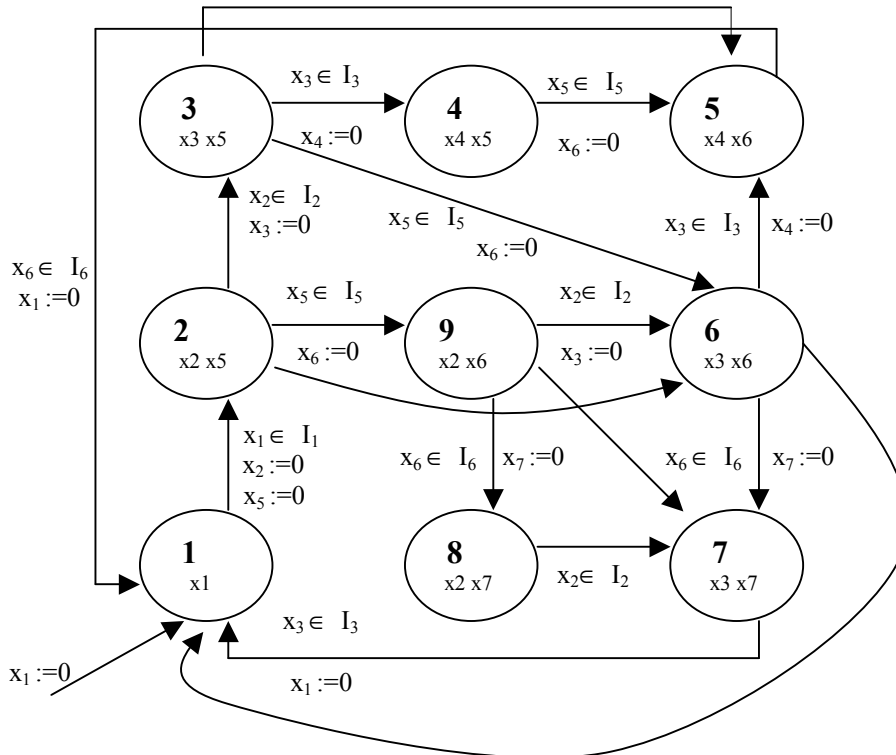


Fig. 37. Modèle dynamique du système de remplissage des bacs.

$x_m$  : horloge d'activation de l'étape  $m$  du grafset de commande.

$I_m$  : l'intervalle de bon fonctionnement de la tâche relative à l'étape  $m$  du grafset de commande.

## 2-2. Modèle de surveillance [RAY-03]

Pour construire le modèle de surveillance du système, nous partons du modèle dynamique qui correspond au fonctionnement normal, sachant que le fonctionnement dégradé lui est analogue, seules les valeurs des temps d'activité sont prolongées aux intervalles notés  $J$  précédemment définis, et tenant compte des différentes possibilités de basculement entre les trois modes ; présentées dans la figure 32, nous obtenons alors le modèle de surveillance de l'application remplissage des bacs représenté dans la figure 38.

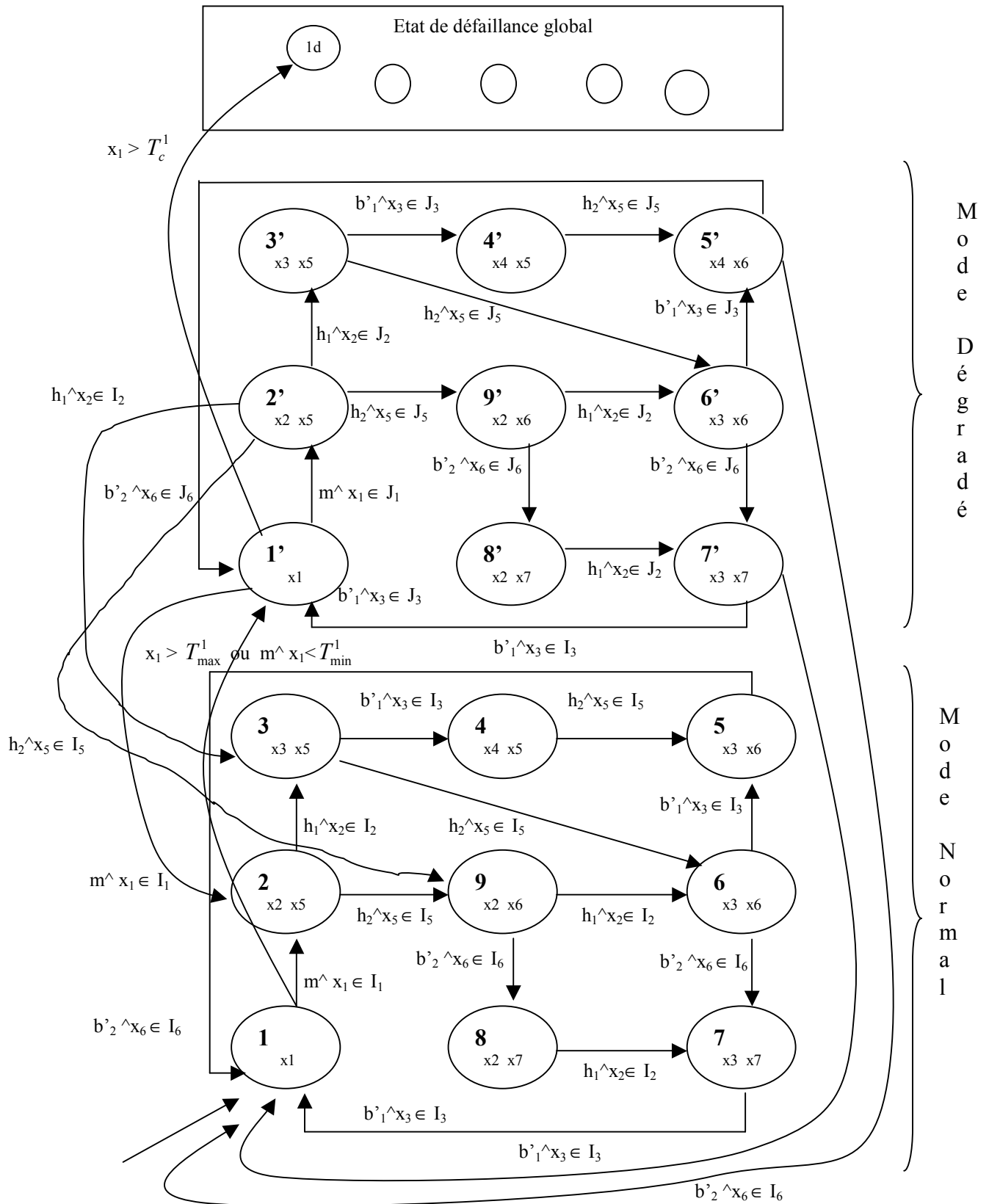


Fig. 38. Le modèle de surveillance de la station de remplissage des bacs.

**Remarque** : Pour ne pas trop encombrer le graphe, nous avons tenu compte des simplifications suivantes : (ceci par rapport à la représentation normalisée de l'automate temporisé modélisant le système de surveillance de l'application)

- l'invariant pour chaque sommet  $i$  correspond à la condition  $x_n \leq T_n$  pour les états du mode normal et  $x_n \leq C_n$  pour les états du mode dégradé, et ceci pour toutes les horloges actives.
- les affectations des horloges correspondant à chaque transition sont notées dans l'état de destination après franchissement.
- les états instables ne sont pas représentés dans le modèle, par exemple l'état où les étapes 4 et 7 actives simultanément.
- le franchissement simultané de deux transitions est supposé physiquement pas possible.

Sur ce modèle de surveillance, nous nous sommes contentés de représenter toutes les situations autour du sommet (1) qui représente l'état du système au début du cycle (étape\_1 active du grafset). Si nous construisons le modèle entier, la lecture de l'évolution du système devient très difficile.

**A partir de cet état en mode normal noté (1),**

Si l'événement  $m$  arrive pendant l'intervalle de bon fonctionnement  $I_1$ , le système évolue vers l'état noté (2) dans le même mode, sinon le système bascule vers l'état (1) du mode dégradé sachant que l'horloge  $x_1$  continue à compter la durée d'attente de fin d'activité.

**A partir de cet état en mode dégradé noté (1'),**

Si  $m$  arrive avant l'écoulement de la durée  $C_1$ , le système continue en mode dégradé (vers l'état (2')) du mode dégradé jusqu'à ce que le temps d'activité devienne normal ce qui ramène le système au mode normal, sinon le système bascule en mode défaillant et y reste jusqu'à l'intervention de l'équipe de maintenance.

**On revient vers cet état noté (1) en mode normal,**

Lorsque l'événement  $b'_1$  arrive pendant la durée limitée par l'intervalle  $I_3$  depuis l'activation de l'état (7') en mode dégradé ou depuis l'activation de l'état (7) en mode normal ou bien si  $b'_2$  arrive pendant la durée limitée par l'intervalle  $I_6$  depuis l'activation de l'état (5') en mode dégradé ou depuis l'activation de l'état (5) en mode normal.

Tenant compte du tableau de la figure 24, nous pouvons constater que le nombre d'arcs supplémentaires au niveau de chaque état du mode normal a doublé en entrée et en sortie, et pour chaque état du mode dégradé ; l'entrée est dédoublée et la sortie est triplée. Donc si nous représentons tous les arcs du modèle de surveillance, nous aurons  $13*7=91$  arcs au total ce qui rend difficile la lecture du modèle.

### 2-3. Conclusion

Jusqu'ici, nous avons présenté la structure globale du contrôle/commande d'un système de production, après nous nous sommes intéressés au module surveillance ; c'est l'élément clé pour les autres modules, la fonction **détection** des défaillances est résolue et partiellement celle du **diagnostic**. Nous avons présenté cette solution basée sur l'outil automate temporisé, en surveillant le temps d'activité d'une tâche. Cette solution présente un réel avantage par rapport à d'autre système de surveillance. En effet, la détection d'une défaillance avec cette approche ne nécessite pas la connaissance exhaustive de toutes les défaillances possibles du système. La détection se fait par le suivi des durées opératoires. Quant au diagnostic, qui correspond à la recherche de l'origine de la défaillance cela se fait indirectement par construction du modèle de surveillance. L'état de défaillance atteint n'a qu'un seul arc qui correspond à l'origine probable de la panne, et l'état qui précède correspond au début de l'apparition des symptômes de la défaillance. Le seul inconvénient est l'explosion du nombre de transition entre les états dans les trois modes de fonctionnement. Ce problème d'explosion en nombre d'arcs est résolu par l'utilisation de l'outil stateflow.

### 3. SURVEILLANCE DU SYSTÈME: ATELIER DE TRAITEMENT DE SURFACE

Un atelier de traitement de surface est composé de plusieurs postes de traitement et un ou plusieurs chariots qui effectuent les différents transferts des pièces entre les postes. La durée de traitement d'une pièce dans une cuve est par hypothèse une variable qui dépend de l'opération à réaliser. L'idée est d'associer à chaque traitement une valeur maximale relative à la durée de traitement. La surveillance est alors basée sur le suivi de ces durées.

La figure 39 montre un exemple de ligne de traitement composée de trois cuves (C1, C2, C3), une station de chargement notée par "Sc", une station de déchargement "Sd" et d'un robot. Le déplacement du robot se fait à travers le rail.

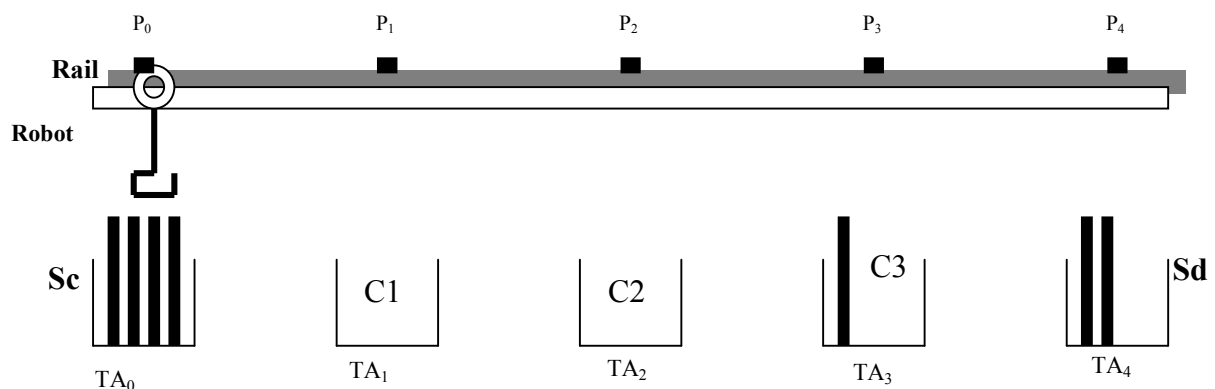


Fig. 39. *Atelier de traitement de surface à surveiller.*

Les caractéristiques techniques de cet atelier sont essentiellement les durées de traitement dans chaque cuve. Il y a aussi les temps de transfert du robot entre les différents postes, utilisés pour le calcul de l'ordonnancement optimal. Le tableau ci-dessous regroupe l'ensemble des paramètres utilisés pour cette application.

Durées de traitement (secondes)	Durées de transport (seconde)	Durées de chargement et déchargement (seconde)
$12 \leq TA_1 \leq 14$ $3 \leq TA_2 \leq 6$ $12 \leq TA_3 \leq 17$	Déplacement de $P_i \rightarrow P_{i+1}$ dure 2 seconde	1

On se propose de surveiller ce système, lors de l'exécution d'un ordonnancement optimal trouvé par des outils de calcul suivant des critères bien définis. Ceci correspond à l'ensemble des séquences  $P_0 P_1 P_3 P_4 P_1 P_2 P_3 P_0$  comme indiqué ci après sur le diagramme de GANNT de la figure 40.

Donc la première étape consiste à rechercher l'ordonnancement optimal pour un critère donné, ensuite pour surveiller la bonne exécution des séquences de l'ordonnancement obtenue. La partie opérative correspond à l'ensemble des transferts nécessaires pour la réalisation de l'ordonnancement. Il faut pouvoir surveiller en permanence la partie opérative ; ce qui revient, d'une part, à vérifier, lors des transferts des chariots entre les différents postes, les passages dans l'ordre (états des capteurs) par les postes intermédiaires et d'autre part, la durée écoulée entre les postes d'origines et de destinations.

Pour identifier les différentes tâches de la partie opérative, un ordonnancement optimal est nécessaire pour la mise en œuvre des systèmes de surveillance. Nous avons choisi un ordonnancement robuste étudié dans [KNO-03].

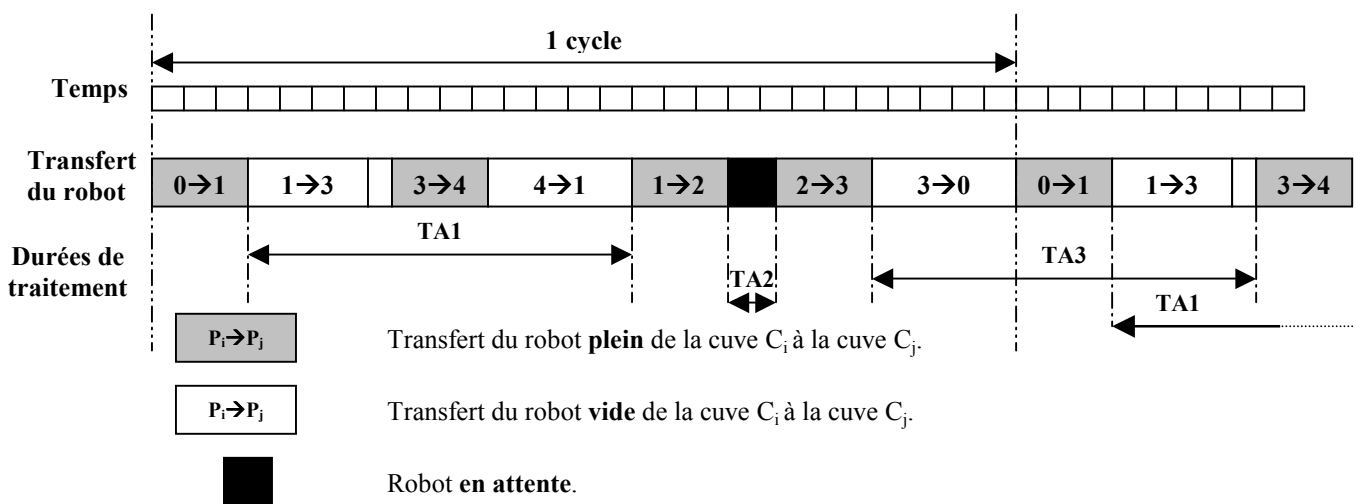


Fig. 40. *Diagramme de GANNT : ordonnancement optimal à surveiller.*



En résumé cette figure correspond à l'ordonnancement donné sous une autre forme qui est :

1. Chargement de plaque dans le poste 0 (capteur de présence  $P_0$ ).
2. Déplacement vers le poste 1 (capteur de présence  $P_1$ ).
3. Dépose de plaque dans la cuve  $C_1$ .
4. Déplacement à vide vers le poste 3 (capteur de présence  $P_3$ ).
5. Retrait de plaque de la cuve  $C_3$ .
6. Déplacement en charge vers le poste 4 (capteur de présence  $P_4$ ).
7. Dépose de plaque dans la station de déchargement Sd.
8. Déplacement à vide vers le poste 1 (capteur de présence  $P_1$ ).
9. Retrait de plaque de la cuve  $C_1$ .
10. Déplacement en charge vers le poste 2 (capteur de présence  $P_2$ ).
11. Dépose de plaque dans la cuve  $C_2$ .
12. Attendre la durée de traitement  $TA_2$ , puis retrait de plaque de la cuve  $C_2$ .
13. Déplacement en charge vers le poste 3 (capteur de présence  $P_3$ ).
14. Dépose de plaque dans la cuve  $C_3$ .
15. Déplacement à vide vers le poste 0 (capteur de présence  $P_0$ ).

### 3-1. Modélisation des opérations de transport

Pour construire le modèle dynamique du système relatif aux fonctions de transport, on procède de la manière suivante : nous rappelons que notre étude se limite à une ligne de production ; ce qui veut dire un seul robot, donc une seule horloge sera suffisante pour surveiller les déplacements entre les différents postes puisqu'il n'y a pas de déplacements simultanés. Alors pour l'ordonnancement des tâches donné précédemment, nous choisissons une horloge notée  $x_r$  qui va être mise à zéro à chaque arrivée à un poste. Et enclenchée au début de chaque départ d'un poste. Ainsi nous obtenons le modèle dynamique relatif aux opérations de transport du robot, en utilisant l'outil automate temporisé ; Comme le montre la figure 41.

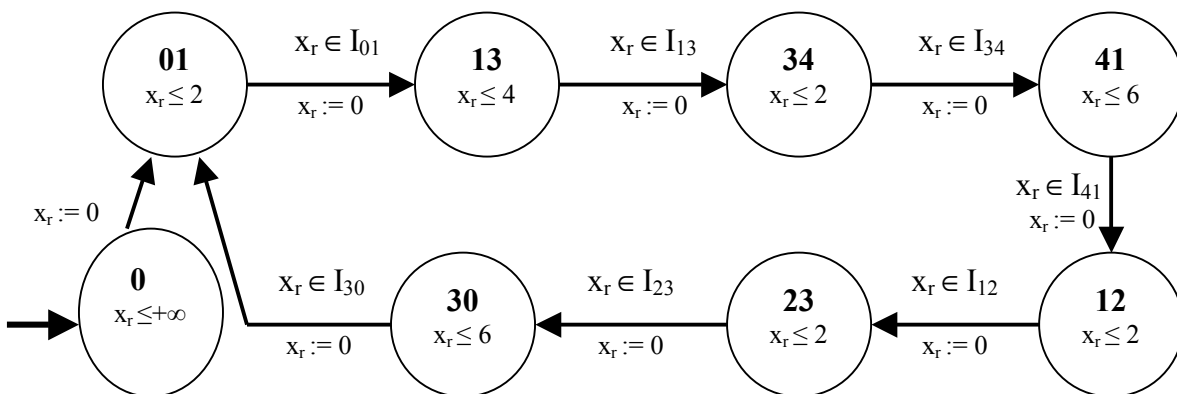


Fig. 41. Modèle automate temporisé pour les opérations de transport.

Pour la surveillance des opérations de transport, les grandeurs à surveiller sont les **états capteurs et les durées de déplacement** alors que pour les opérations de traitement, seuls les **durées de traitement** sont surveillées comme on va le voir dans la partie suivante.

### 3-2. Modélisation des opérations de traitement

Il s'agit ici de surveiller les durées de traitement dans chacune des cuves. Nous remarquons que, par exemple, pour l'ordonnancement étudié, nous pouvons avoir des plaques simultanément dans les cuves  $C_1$  et  $C_3$  ou bien dans  $C_2$  et  $C_3$ . Ces situations possibles nous conduisent à prendre une horloge qui surveille la durée de traitement dans chacune des cuves séparément  $x_{t1}$ ,  $x_{t2}$  et  $x_{t3}$ . Nous obtenons ainsi le modèle dynamique des opérations de traitement représenté dans la figure suivante.

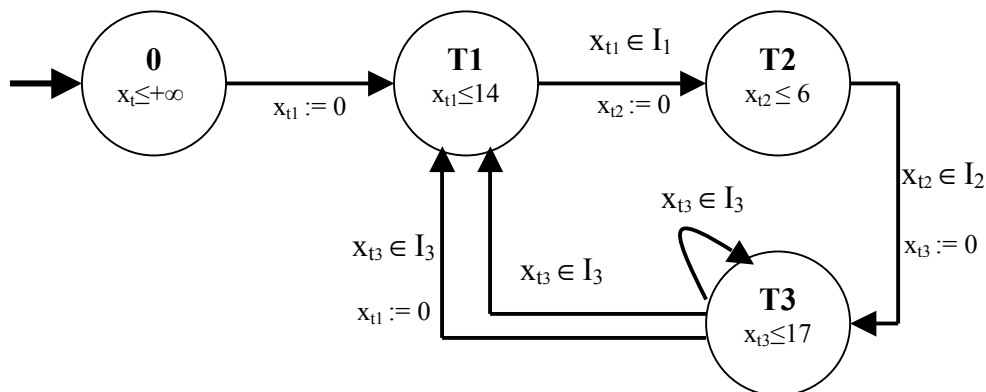
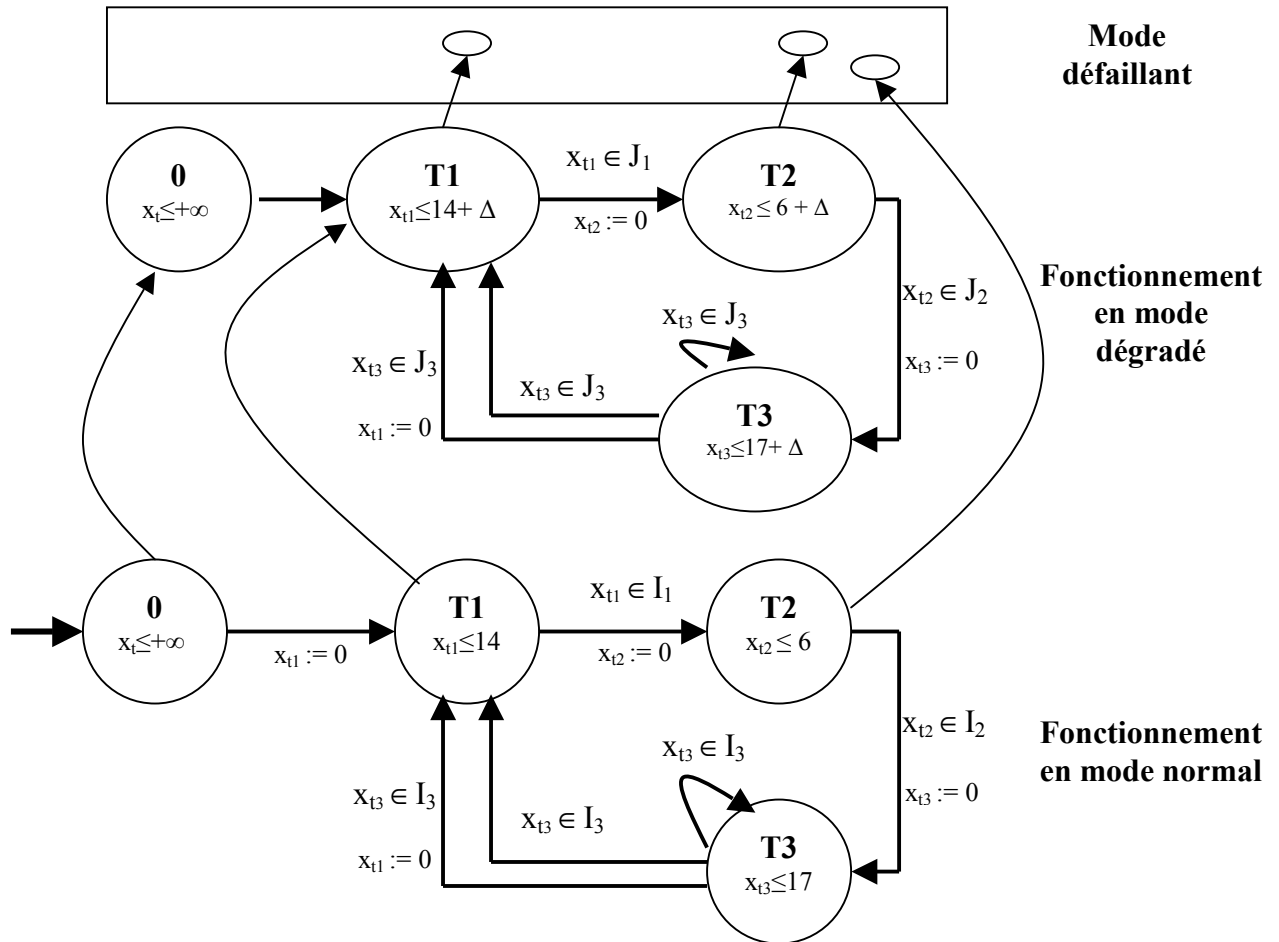


Fig. 42. Modèle automate temporel pour les opérations de traitement.

A partir de ce modèle, nous construisons le modèle de surveillance relatif aux opérations de traitement de surfaces. Ceci en tenant compte du critère d'évaluation qui définit le passage entre les trois modes de fonctionnement du système que l'on veut surveiller.

A titre d'exemple nous présentons le modèle de surveillance des tâches de traitement, il s'obtient de la même manière que pour la première application étudiée ; celle de remplissage des bacs.

Tenant compte des valeurs des caractéristiques techniques de l'atelier à surveiller, nous définissons les marges de tolérance que l'on cherche par la suite à optimiser, et ce pour chacune des tâches. Nous obtenons alors le sous-modèle de surveillance relatif à la surveillance des tâches de traitement de surfaces.



Remarquons que l'ensemble des arcs correspond à toutes les situations illustrées par la figure 32. nous aurons pour ce sous modèle  $12 \times 5 = 60$  arcs. Alors lorsqu'on construit le modèle de surveillance global de tout l'atelier la complexité est évidente pour la lecture et la lisibilité des évolutions du système. d'où le choix de l'outil stateflow pour améliorer la lisibilité et diminuer la complexité par sa structure hiérarchique.

### 3-3. Construction du Modèle de surveillance par stateflow

Le système global peut évoluer dans les trois modes de fonctionnement. Il est soit en fonctionnement normal si les durées des différentes tâches ; de transfert entre postes et de traitement dans tous les postes restent incluses dans leurs intervalles respectifs de bon fonctionnement. Lorsque la durée de traitement ou de transport a été dépassé alors le système bascule vers le fonctionnement en mode dégradé pendant un intervalle de tolérance fixé. L'intervalle de tolérance est défini en fonction de l'application. Le système se trouve dans un état de défaillance si le seuil de tolérance est dépassé, pour l'une au moins des tâches surveillées.

L'automate temporisé qui représente le modèle global de surveillance nécessite un nombre d'états élevé. La surveillance des opérations de transport nécessite  $m$  états pour un cycle et  $j$  opérations de traitement. Sachant qu'il y a trois modes de fonctionnement, le nombre

total d'état du système de surveillance est supérieur à  $3*(m+j)$ . Grâce à l'utilisation des statecharts, basés sur la structuration hiérarchique du séquençement des tâches, nous obtenons une modélisation plus concise avec moins d'états. Ceci nous permet de localiser facilement le lieu de la panne et par la suite faire le diagnostic afin de limiter la propagation de la dégradation qui peut entraîner la défaillance du système. La simulation des modes de fonctionnement, en injectant des défauts, sera présentée dans le chapitre 4.

Le modèle de surveillance global de l'atelier de traitement de surfaces sous stateflow, pour l'ordonnancement optimal préalablement calculé (§4.2). Le diagramme statecharts sera composé des trois parties suivantes : Comme le montre la figure 43.

- La surveillance des opérations de transport qui correspondent, pour un ordonnancement donné, aux différents déplacements du robot (notée : Surveillance du transport)
- La surveillance des opérations de traitement qui correspondent aux traitements à effectuer au niveau de chaque cuve, sachant que lorsque la durée de traitement a été dépassée les pièces sont mises en rebut (notée : Surveillance des traitements).
- Le suivi, en temps réel, des modes de fonctionnement qui correspondent aux trois modes (normal, dégradé et défaillant) du système (noté : état global du mode de fonctionnement du système).

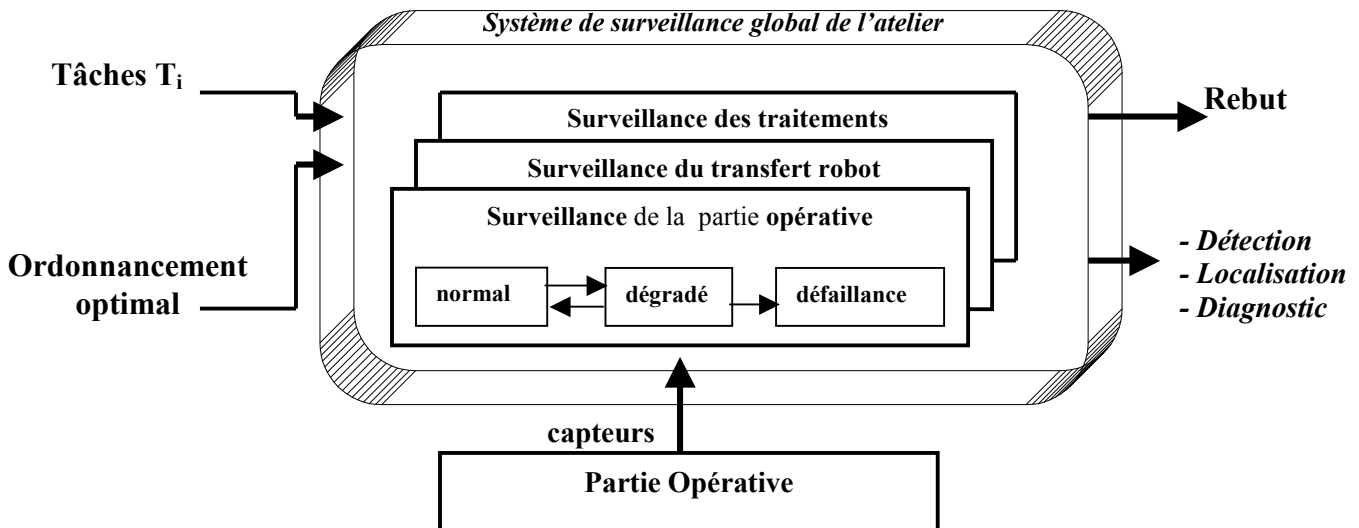


Fig. 43. Modèle global de surveillance de l'atelier de traitement de surface.

## CHAPITRE 4.

# ESTIMATION DES TEMPS DE DÉTECTION

*L'objectif de notre démarche est de surveiller, en temps réel, les durées d'exécutions des tâches. Pour se faire, nous avons défini auparavant les intervalles de bon fonctionnement et de fonctionnement dégradé. Au-delà de ces intervalles, le système est considéré comme défaillant. La finesse de la détection réside dans le choix de la largeur de l'intervalle de tolérance, qui correspond au fonctionnement en mode dégradé. L'intervalle de fonctionnement normal est défini par apprentissage direct sur le système de production qu'on se propose de surveiller.*

*L'estimation des temps de détection sera faite dans l'objectif de minimiser le nombre de fausses alarmes tout en empêchant le système d'aller vers des situations qui peuvent pénaliser la production et éviter d'atteindre des situations catastrophiques. Nous utilisons pour cela l'outil stateflow bien adapté pour la modélisation des processus industriels et la simulation des enchaînements des tâches. Nous verrons que cet outil nous permet, par sa nature hiérarchique, de faire le diagnostic des défaillances. L'application sera faite pour les deux exemples décrits dans le chapitre précédent.*

## 1. DEMARCHE D'ESTIMATION DES TEMPS DE DETECTION

Dans un système de production, généralement nous trouvons plusieurs machines et chacune de ces machines exécutent une ou plusieurs tâches. Ces différentes tâches sont réalisées en série ou en parallèle dans le temps. Pour pouvoir appliquer notre approche de surveillance à un système, il faut définir pour chacune de ces tâches les trois bornes qui définissent les trois modes de fonctionnement ; à savoir le mode normal, le mode dégradé et le mode défaillant.

### 1-1. Définitions des intervalles de fonctionnement

Chaque tâche  $m$  correspond à l'exécution d'une opération bien définie. Nous estimons que la durée d'une tâche  $m$ , lorsqu'elle est exécutée normalement, peut varier dans un intervalle compris entre  $T_{\min}^m$  et  $T_{\max}^m$ .

Nous avons appelé cet intervalle " intervalle de bon fonctionnement pour la tâche  $m$  " noté :  $I_m = [T_{\min}^m ; T_{\max}^m]$ . Cela signifie que la tâche est considérée comme réalisée lorsque le compte rendu «fin de tâche» arrive pendant cet intervalle.

- La borne inférieure de l'intervalle  $I_m$  correspond au temps minimal nécessaire pour la réalisation de la tâche  $m$ , son choix permet principalement de minimiser les fausses alarmes causées par le changement accidentel des états capteurs.
- La borne supérieure de l'intervalle  $I_m$  correspond au temps nécessaire et suffisant pour la bonne exécution de la tâche  $m$ , son choix est plus délicat que la borne inférieure. Ceci pour plusieurs raisons ; à savoir que c'est la limite entre les deux modes de fonctionnement normal et dégradé, c'est une valeur à calculer statistiquement (valeur moyenne par exemple). L'optimisation dépendra étroitement du bon choix de cette valeur.

Par exemple, un robot qui se déplace à une vitesse constante  $V$ , entre deux postes situés à une distance  $D$ , l'un par rapport à l'autre. Normalement il mettra un temps  $T=D/V$ , mais pour choisir les bornes de l'intervalle de bon fonctionnement  $I_m = [T_{\min}^m ; T_{\max}^m]$  nous devons nous référer à l'atelier, voir son milieu, sa qualité..., ce qui nécessite une analyse des durées réelles d'exécutions des différentes tâches (calcul du temps moyen d'exécution par exemple) pour la bonne détermination des valeurs limites. La détection sera d'autant meilleure que les valeurs des limites sont bien choisies.

Nous avons appelé intervalle de tolérance pour la tâche  $m$ , celui qui correspond au fonctionnement en mode dégradé, noté :  $J_m = [T_{\max}^m; T_c^m]$ . Il délimite le retard toléré pour recevoir l'information de fin de tâche. En effet, il tient compte des retards qui peuvent être dus, soit aux vieillissements des machines par exemple, soit aux surcharges, soit aux problèmes mécaniques ou autres.

Enfin, le système de production sera en mode défaillant lorsque l'événement de fin de tâche arrive au-delà de l'écoulement du temps critique noté :  $T_c^m$  pour une tâche  $m$  donnée. Lorsque le système se trouve dans cet état, il nécessite l'intervention des équipes de maintenance pour la remise en service après avoir fait les réparations nécessaires. Ceci pénalise le temps de production et par la suite le coût. Il faut donc éviter au maximum d'aller dans cet état tout en surveillant l'installation contre les défaillances possibles.

L'objectif de notre travail est d'optimiser la largeur de cet intervalle, pour éviter le grand nombre d'arrêts de la production lorsqu'elle est très petite et en même temps éviter de ne pas détecter des défaillances qui peuvent causer par la suite des dégâts au système de production.

## 1-2. Algorithmes d'optimisation des temps de détection

### a- Les algorithmes existants : rappel

Les algorithmes d'optimisation permettent de s'affranchir en grande partie de la méthode d'essai et erreur. Parmi ceux-ci nous trouvons les **méthodes de recherche aléatoire** puis les **algorithmes du recuit simulé** et les **algorithmes Evolutionnaires**.

Les **méthodes de recherche aléatoire**, sont intéressantes lorsque le nombre de paramètres reste inférieur à trois ou quatre. Elles ont l'avantage de la simplicité, de plus elles permettent d'avoir un bon aperçu de l'espace de recherche : Les nuages de points que l'on peut en tirer sont souvent riches en informations. Parmi celles-ci, nous citons la méthode des sauts aléatoires et la méthode de la promenade aléatoire.

Dans la **méthode des sauts aléatoires**, nous utilisons l'algorithme suivant :

Soit  $x = [x_1, x_2, x_3, x_4, \dots, x_n]^t$  avec  $l_i \leq x_i \leq u_i$   $i = 1 \dots n$

Soit  $r = [r_1, r_2, r_3, r_4, \dots, r_n]^t$

avec  $r_i$  variables aléatoires uniformément distribuées sur  $[0,1]$ .

$$\text{Alors } \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} l_1 + r_1(u_1 - l_1) \\ l_2 + r_2(u_2 - l_2) \\ \vdots \\ l_n + r_n(u_n - l_n) \end{bmatrix}$$

Dans la méthode de la promenade aléatoire, nous utilisons l'algorithme suivant :

$x_{k+1} = x_k + \alpha_k \cdot d_k$  avec  $d_k$  vecteur aléatoire

conditions initiales :  $x_0$   $\alpha_0$   $k=0$  et  $cc=1$

1°- soit  $\{r_i\}$  n variables aléatoires uniformément distribuées sur  $[-1, 1]$ .

$$\text{Si } R = \sqrt{r_1^2 + r_2^2 + \dots + r_n^2} > 1 \text{ alors nouveau } \{r_i\} \text{ avec } d_k = 1/R \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} \quad R \leq 1.$$

2°-  $x_{k+1} = x_k + \alpha_k \cdot d_k$

3°- si  $f(x_{k+1}) < f(x_k)$  alors  $k = k+1$   $cc = 1$  puis retour en 1°

    sinon si  $cc < N_{cc}$  alors  $cc = cc+1$  puis retour en 1°

        sinon  $\alpha_k = \alpha_k/2$   $cc = 1$  puis retour en 1°

        si  $\alpha_k < \varepsilon$  alors stop.

L'avantage de ces méthodes c'est que  $f(x)$  peut être discontinue et non dérivable. Elle permet aussi de trouver un minimum global pour la fonction d'optimisation.

Les algorithmes du **recuit simulé** (simulated annealing) sont basés sur l'analogie avec un processus thermique en physique de la matière : le recuit (faire fondre un solide puis le refroidir lentement).

L'algorithme de Métropolis : simulation du recuit, se présente par les relations :

Soit un état  $x_k$  de niveau d'énergie  $E_k$ .

Après une perturbation aléatoire, l'état suivant noté :  $x_{k+1}$  avec niveau  $E_{k+1}$ .

Si  $E_{k+1} \leq E_k$  alors  $x_{k+1}$  est le nouvel état

Si  $E_{k+1} > E_k$  alors  $x_{k+1}$  est le nouvel état avec probabilité  $\exp(-(E_{k+1} - E_k)/k_B \cdot T)$

$k_B$  : constante de Boltzmann et  $T$  : température (Kelvin).

$$P(x_{k+1} \text{ soit accepté}) = \begin{cases} 1 & \text{si } f(x_{k+1}) \leq f(x_k) \\ e^{-(f(x_{k+1}) - f(x_k))/c_k} & \text{sinon} \end{cases} \quad c_k : \text{Paramètre de contrôle.}$$

Initialisation :  $x_0$   $c_0$   $L_0$  et  $k=0$

Pour  $i = 1$  générer aléatoirement  $x_{ki}$  dans un voisinage  $x_k$

    Si  $f(x_{ki}) \leq f(x_k)$  alors  $x_k = x_{ki}$

    Sinon si  $e^{-(f(x_{ki}) - f(x_k))/c_k} >$  variable aléatoire uniforme sur  $[0,1]$

    Alors  $x_k = x_{ki}$

$k=k+1$   $x_{k+1} = x_k$

modifier  $L_k$

modifier  $c_k$



les avantages de cette méthode réside dans la possibilité d'échapper d'un minimum local ainsi que l'ajustement possible de l'agitation.

Les algorithmes **Evolutionnaires** sont basés sur la théorie de l'évolution de Darwin. Par analogie avec le monde biologique, l'algorithme fait évoluer une population de dispositifs à l'aide de divers opérateurs : sélection, croisements et mutations. Parmi ces algorithmes, on distingue les algorithmes **génétiques** qui utilisent un codage des paramètres sous forme de chaînes binaire, par analogie avec l'ADN.

Principe :

Population initiale  $P(0)$  avec  $x \in P(0) \iff x \in \Omega$

Naissance d'une population  $P(1)$  par croisements et mutations

Le population  $P(1)$  va engendrer ensuite une population  $P(2)$ , etc...

Le codage des points de  $\Omega$  construit sous forme de suites de  $L$  symboles tirés d'un alphabet.

#### b- L'approche d'estimation choisie

La méthode de base pour optimiser un dispositif est la méthode d'essai et erreur ; basée sur l'évaluation des performances du système pour différentes valeurs de  $\Delta$ . Il s'agit de tester un certain nombre de solutions potentielles jusqu'à l'obtention d'une solution adéquate. Nous traçons alors la courbe nombre de d'alarmes par rapport aux différentes valeurs de  $\Delta$ .

Comme nous l'avons vu dans l'utilisation des réseaux de Petri, nous allons procéder à l'estimation des temps de détection, par simulation appliquée aux deux exemples déjà étudiés dans la partie précédente. Nous explicitons tout d'abord l'algorithme utilisé pour l'estimation des temps de détection.

Nous allons exposer la méthode d'estimation en faisant le raisonnement pour une tâche  $m$  donnée. Il s'agit toujours de calculer les temps de détection dans un système de surveillance basé sur le suivi, au cours du fonctionnement, des durées d'exécutions des tâches. Donc nous partons du modèle de commande, qui peut être sous forme d'un grafcet (i.e. le cas du système de remplissage des bacs) ou bien sous forme d'ordonnancement choisi (i.e. le cas de l'atelier des traitements de surfaces).

Ensuite nous construisons le ou les modèles dynamiques relatifs à chaque activité du système de production, lorsqu'il y en plusieurs. Pour cela nous aurons besoin de définir les intervalles de fonctionnement normal et de fonctionnement dégradé. Alors généralement, les valeurs limites des différents intervalles sont déterminées de la manière suivante :

- La valeur minimale notée :  $T_{\min}^m$  représente le temps minimal nécessaire pour la bonne exécution de la tâche m, il est généralement donnée par le constructeur du système de production.
- La valeur critique notée :  $T_c^m$  représente le temps maximal au-delà duquel, si l'événement «fin de tâche» n'est pas arrivé, alors nous estimons que la tâche n'a pas été réalisée. Généralement choisi en fonction de la criticité de la tâche ; par exemple dans le domaine nucléaire, nous avons intérêt à le choisir avec une erreur de l'ordre de la seconde afin d'éviter des accidents catastrophiques. Par contre, dans le domaine agricole, l'erreur peut être de l'ordre de la minute sans courir de risques majeurs.
- La valeur maximale notée :  $T_{\max}^m$  représente le temps nécessaire pour l'exécution de la tâche dans les conditions normales de fonctionnement. Au-delà de cet instant, nous estimons qu'il y a des possibilités d'avoir un début de déviation par rapport au fonctionnement normal, d'où le passage en mode dégradé à partir de cet instant.

Toute la difficulté réside dans le choix de la valeur de  $T_{\max}^m$  dans un premier temps, ensuite la largeur de l'intervalle noté :  $J_m$  qui correspond au retard toléré pour attendre l'arrivée de l'événement «fin de tâche» tout en continuant la production en mode de fonctionnement dégradé. Nous proposons dans ce qui suit, l'algorithme d'estimation des temps de détection ; présenté par la figure 45, par évaluation des performances, pour minimiser le nombre de fausses alarmes, minimiser les temps d'arrêt de production, tout en limitant les risques d'accidents ou dégâts aux installations.

## 2. EVALUATIONS DES PERFORMANCES

Pour évaluer les performances du système de surveillance obtenu après optimisation des fausses alarmes, nous définissons le **taux de couverture**. Pour calculer le taux de couverture, pour un seuil de tolérance trouvé, nous devrions faire varier le temps de service en fixant la valeur de tolérance. Ceci dans le but de voir l'influence du temps de service sur le seuil calculé. Ce seuil sera t-il correct pour des tâches qui durent plus ou moins que le temps de service pour lequel le seuil a été déterminé. Si oui quelles sont les limites possibles pour que ce seuil reste applicable.

Dans ce but, nous définissons le taux de couverture noté :  $C_d = \frac{N_d}{N_i}$  avec ;

$N_d$  : Nombre de défaillances détectées ;

$N_i$  : Nombre de défauts injectés.

Ce taux de couverture  $C_d$  permet de classer les performances du système de surveillance. Nous classons alors la surveillance, selon trois niveaux de performance, comme indiqué dans la figure 44, de bonne, moyenne ou mauvaise, ceci suivant la valeur du rapport  $C_d$ .

$0,8 < C_d < 1$	$0,4 < C_d < 0,8$	$0 < C_d < 0,4$
Bonne	moyenne	mauvaise

*Fig. 44. Classification de la surveillance selon la valeur du taux de couverture.*

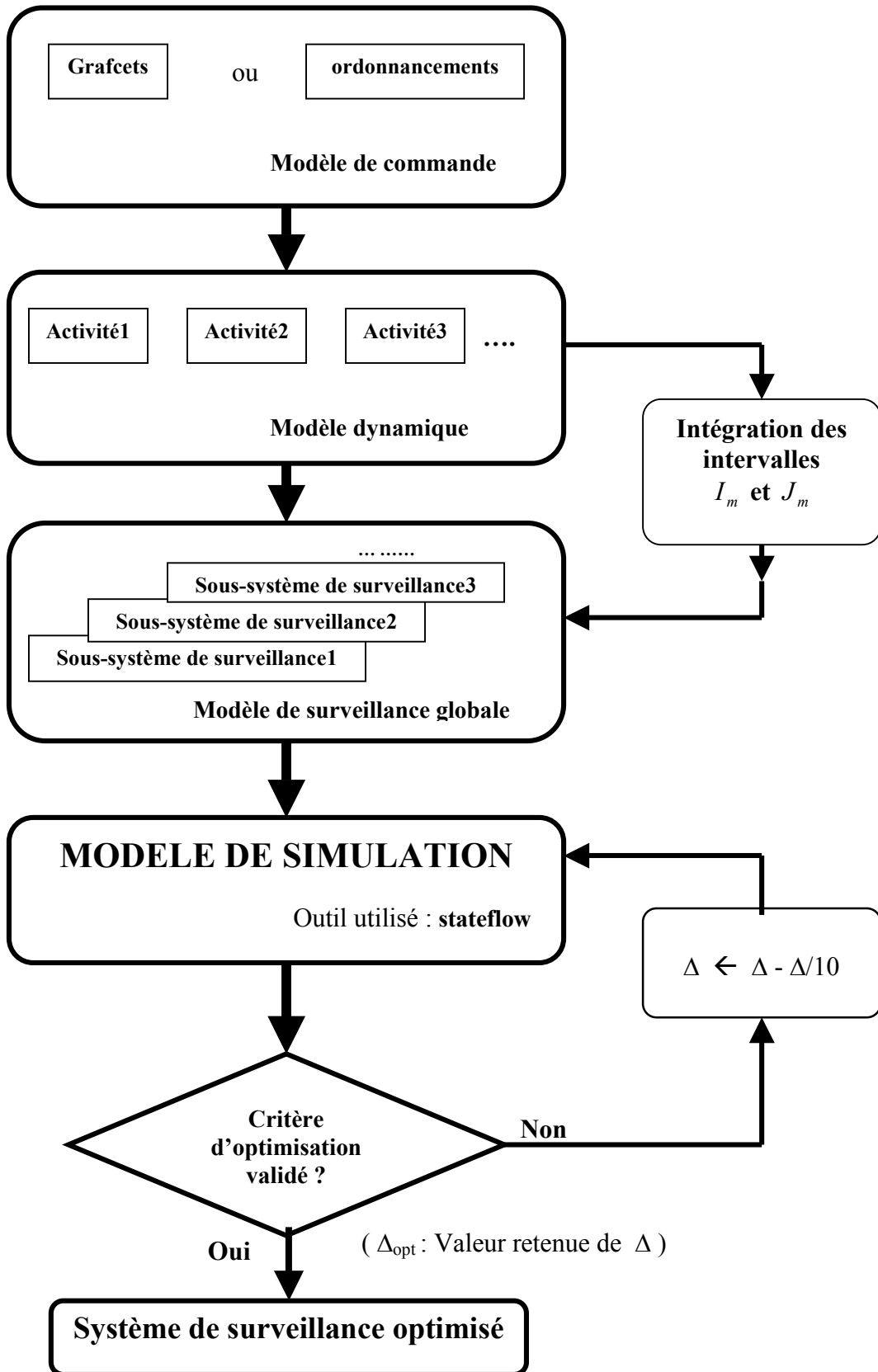


Fig. 45. *Algorithme d'optimisation des temps de détection des défaillances.*

## 2. PRÉSENTATION DU LOGICIEL STATEFLOW

Le comportement d'un processus physique automatisé est décrit dans un document qui définit les étapes du fonctionnement et leur enchaînement. Il précise les conditions de passage d'une étape à l'autre et les actions qu'elles doivent exécuter. Le descriptif, en forme de cahiers des charges, se présente en principe sous une forme normalisée dont le grafcet est un exemple élaboré.

Stateflow est un outil de description et de simulation qu'on peut comparer au grafcet, lequel permet de décrire, sous une forme normalisée la logique du comportement des parties séquentielles. Stateflow fonctionne en symbiose avec simulink. Il est donc possible de modéliser un processus industriel et de simuler l'enchaînement des tâches en même temps qu'elles s'exécutent. Une fois mis au point ; le code C obtenu par le compilateur peut, grâce au logiciel temps réel Real Time Workshop, être téléchargé dans une cible temps réel et faire fonctionner un automate programmable industriel.

Nous commençons ce chapitre, par la méthode de construction, grâce à un éditeur approprié, les différents éléments du diagramme (chart) dans le cas général. Puis nous l'appliquons pour la simulation des systèmes de surveillance relatifs aux deux exemples précédemment étudiés dans le chapitre 3. Nous terminons par l'exploitation des résultats afin d'optimiser les temps de détection des défaillances.

### 2-1. Fonctionnement d'un diagramme stateflow

Stateflow s'utilise de manière analogue à simulink. On construit tout d'abord le schéma en plaçant, grâce à l'éditeur, les différents éléments du diagramme (chart) nécessaires et qui sont essentiellement les **états** (state), leurs **liaisons** (transitions) ainsi que les **libellés** attribués aux **événements** (events) et aux **actions**.

Il est bon de remarquer la présentation extrêmement concise du modèle à l'aide de sous-systèmes, y compris le sous-système stateflow. On y distingue clairement les signaux échangés entre ses différentes parties. Un schéma stateflow doit être conçu de manière hiérarchisée. Les objets graphiques et non graphiques vont s'emboîter selon la hiérarchie parent-enfants.

Le plus haut niveau de la hiérarchie stateflow est appelé « Machine » ; il correspond au niveau « Modèle » dans la hiérarchie simulink. Ce vocable est issu du concept de **machine à états finis** qui a été développé (Moore ou Mealy) pour rendre compte du fonctionnement des systèmes à événements discrets. Dans ces systèmes, le passage d'un **état** à un autre est gouverné par des événements discrets, alors que dans les systèmes continus (ou échantillonnés)

le nombre des états est infini et que le passage d'un état au suivant se fait au fil du temps. La **machine** peut réunir **plusieurs diagrammes stateflow** dont elle est le parent.

Au deuxième niveau, nous avons le diagramme qui correspond au sous-système simulink et décrivant une partie de la machine. Il rassemble des **états** reliés par des transitions dont il est le parent. Le **diagramme** décrit le fonctionnement logique d'un sous-ensemble particulier de la **machine**. Une machine comprend bien souvent plusieurs diagrammes qui, dans la plupart des cas, communiquent entre eux, mais elle peut ne comprendre qu'un seul diagramme. Ce sont les **événements** qui pilotent le diagramme et notamment le franchissement des transitions franchissables. Sachant qu'un événement est un objet non graphique qui a été déclaré par le concepteur à un certain niveau de la hiérarchie du diagramme et pour lequel a été précisé son domaine d'action (local, input from simulink... notamment).

Les objets dont nous aurons besoins, dans la plupart des cas, pour construire un diagramme sont :

- l'**état**, représenté par un rectangle aux coins arrondis.
- la **transition**, représentée par un lien orienté entre deux états. La **transition-défaut** est particulière, elle indique l'état qui devra être actif à l'initialisation du diagramme, c'est à dire à son premier réveil.
- des **objets** non graphiques qui sont les libellés donnés aux états, aux transitions, aux actions, aux événements..., ces noms sont associés à des variables manipulées par le programme (S-fonction) sous-jacent et ne sont donc pas de simples étiquettes de reconnaissance.

## 2-2. Construction d'un diagramme

Avant de représenter le diagramme il faut procéder à l'analyse du problème à traiter ; dans notre cas il s'agit de simuler le modèle de surveillance d'un système de production. De cet examen doivent se dégager peu à peu les différents **états** qui correspondent aux fonctions du système. Puis chaque fonction sera détaillée à son tour et les états correspondant mis à jour, et ainsi jusqu'au stade ultime où tous les aspects auront été pris en compte, toutes les actions repérées, toutes les conditions répertoriées. C'est une opération difficile de modélisation mais dont la pertinence est essentielle.

La descendance d'un parent est soit du type **ou exclusif (or)**, soit du type **parallèle (and)**. C'est par leur bordure qu'on distingue les deux types, l'état **ou exclusif** a une bordure pleine et celle de l'état **parallèle** est en trait pointillé. Lorsqu'un parent est actif tous ses enfants le seront si la décomposition est de type or. Il appartient au concepteur de désigner ce

seul état actif au moyen d'une transition particulière appelée, en anglais, **default transition** ou d'une jonction particulière appelée **history junction**.

La figure 46 présente un diagramme, de surveillance sous stateflow, pour un atelier de production composé de deux machines. La première machine réalise simultanément trois activités en plus de la partie opérative (ordres opérateur, capteurs et actionneurs), la deuxième machine réalise deux activités, une de transport et l'autre de traitement ; un seul traitement se fait à la fois. L'état global de l'atelier permet de résumer l'état de fonctionnement de l'atelier.

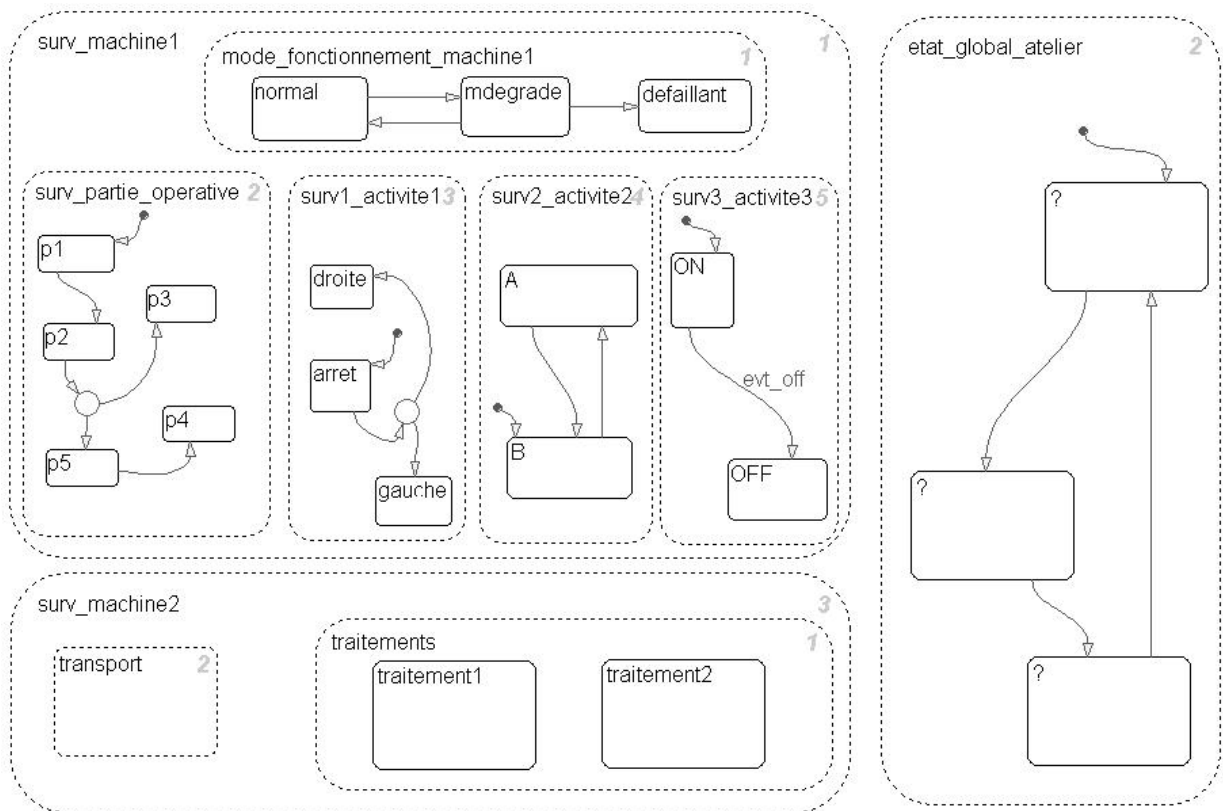


Fig. 46. Exemple de diagramme 'statecharts' représentant un modèle de surveillance.

### 3. SIMULATIONS ET ANALYSES

Dans ce paragraphe, nous allons appliquer l'algorithme d'estimation des temps de détection des défaillances, que nous avons présenté dans la figure 45. L'objectif de ces travaux de simulations consiste à déterminer :

- le **seuil optimal**, noté : que l'on peut tolérer pour le système à surveiller.
- le **temps de détection minimal** calculé pour le seuil  $\Delta_{opt}$  choisi, il est noté :  $Td_{opt}$ .

Nous revenons aux deux applications, pour lesquelles nous avons construit les modèles de surveillance correspondants, à savoir celle de la station de remplissage des bacs et celle de l'atelier de traitement de surface. Notons encore une fois que nous sommes partis du modèle de commande sous forme de grafcet pour les bacs et d'un ordonnancement optimal pour le traitement de surface.

### 3-1. Résultats de simulation : application de remplissage des bacs

Pour cette première application, nous avons présenté par la figure 38 le modèle de surveillance en utilisant l'outil automate temporisé, d'ailleurs nous n'avons pas représenté toutes les différentes liaisons de passage entre les états du système à cause de leur nombre élevé, il y en a 91 arcs. C'est la raison pour laquelle nous avons proposé l'outil stateflow, que nous estimons, bien adapté à notre objectif de surveillance. La figure 46 présente le diagramme statecharts, qui représente le modèle de surveillance de l'application de remplissage des bacs.

Nous retrouvons dans cette figure, les neuf états du système avec moins d'arcs et plus de lisibilité pour savoir à tout instant la situation dans laquelle se trouve notre système.

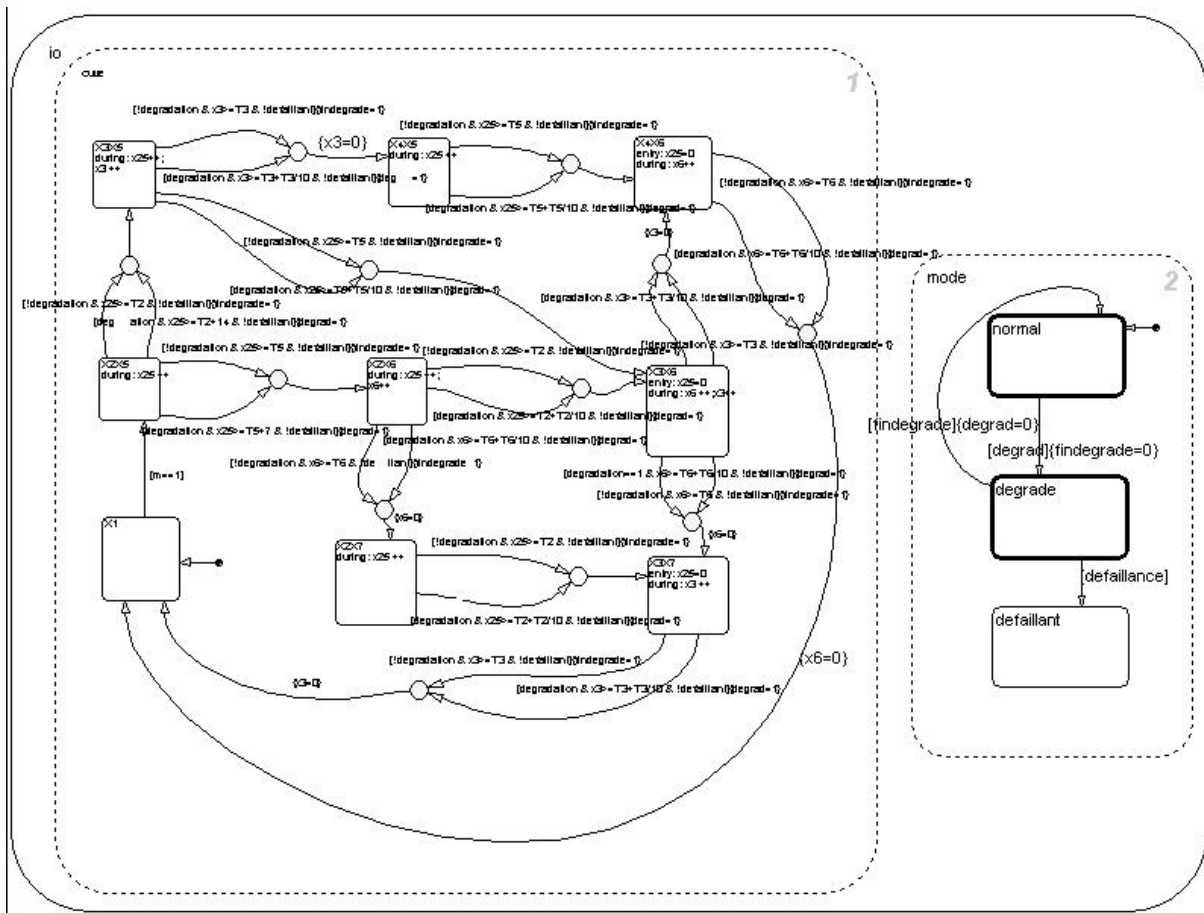


Fig. 47. Modèle de surveillance de la station de remplissage des bacs.





Si nous représentons ces résultats sous formes graphiques, pour situer le seuil optimal ; ce qui correspond au minimum de fausses alarmes tout en détectant les défaillances du système.

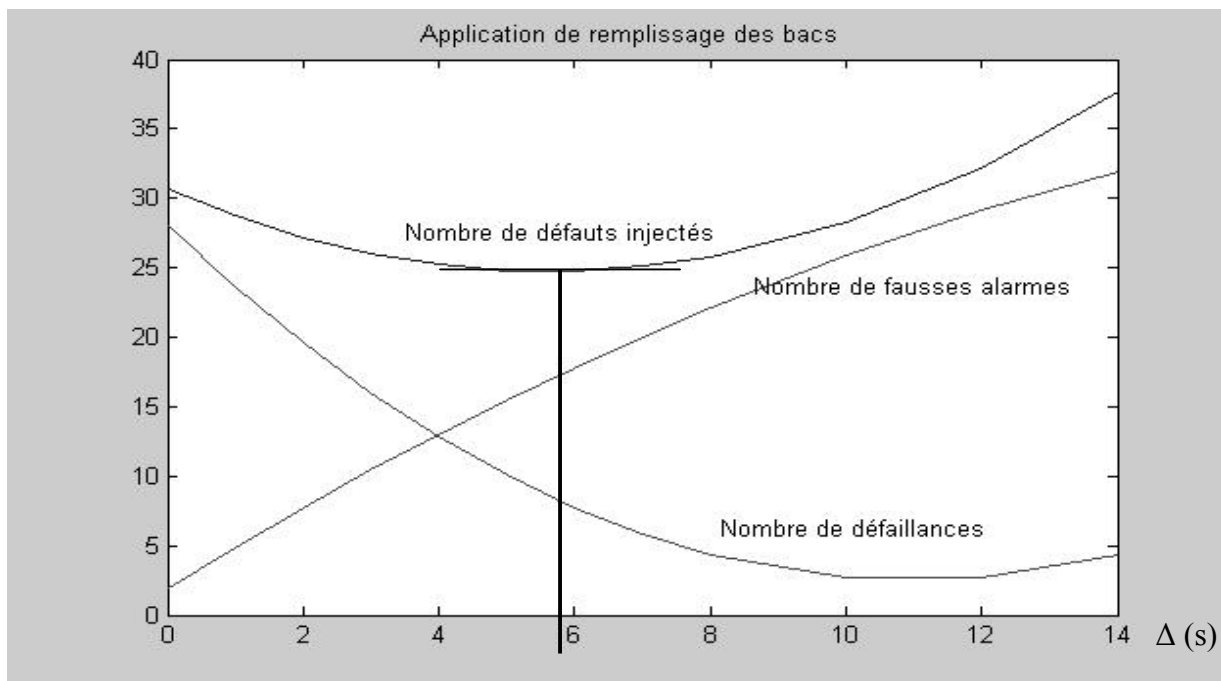


Fig. 49. Détermination du seuil de tolérance.

D'après les courbes de la figure 49, nous pouvons choisir la valeur de 6 secondes comme seuil de tolérance. Ensuite, nous passons à l'étape d'évaluation des performances du système de surveillance. Pour cela nous allons fixer cette fois-ci le seuil et nous cherchons jusqu'à quelle valeur, ce seuil reste satisfaisant. Ceci en évaluant le taux de couverture de ce seuil. Nous faisons varier le rapport  $\Delta/\delta$  et nous notons le nombre de fausses alarmes par rapport au nombre de défauts injectés. Nous obtenons les résultats présentés sous formes du tableau suivant :

Rapport des temps $\Delta/\delta$ en [%]	Nombre moyen des défauts injectés ' $N_i$ '	Nombre moyen des défauts détectés ' $N_d$ '	Taux de couverture	classification
5	34,33	33,60	0,98	bonne
8,5	29,66	28,33	0,96	
10	28,00	26,67	0,89	
15	27,67	24,66	0,80	
20	26,33	23,67	0,78	moyenne
25	24,67	21,00	0,76	
30	28,33	22,67	0,69	
35	29,00	23,67	0,65	
40	26,67	21,33	0,61	
50	31,33	15,67	0,57	

Fig. 50. Classification de la surveillance en fonction du seuil de tolérance.

### 3.1-2 Détermination du temps de réponse

Nous retenons la valeur du seuil calculé précédemment égale à 6 secondes. Nous faisons alors quelques simulations pour déterminer le temps de réponse du système de surveillance proposé. Ceci nous donnera l'ordre de grandeur du temps de détection sous ces conditions.

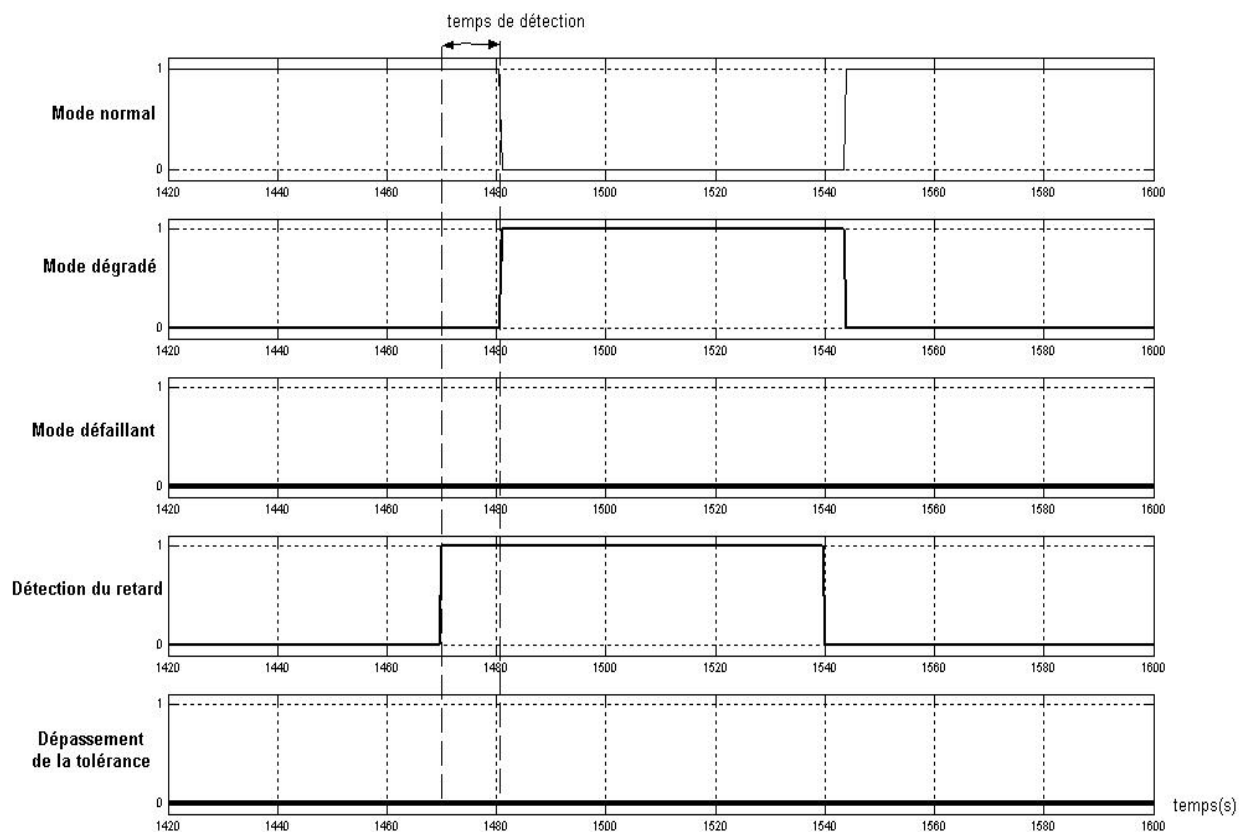


Fig. 51. *Détermination des temps de réponse du système de surveillance.*

#### Conclusion :

D'après les résultats illustrés par la figure 50, nous constatons que le seuil calculé pour le temps de service initialement égal à 70 secondes, reste satisfaisant pour des tâches similaires de durées pouvant aller jusqu'à un rapport de 15% tout en gardant une bonne qualité de surveillance (ce qui correspond à une durée de service maximale de 90 secondes).

Comme le montre la figure 51, le temps de détection trouvé qui correspond à la tâche de remplissage du bac 2 avec le seuil de tolérance égal à 6 secondes, est de l'ordre de 11 secondes environ. Relativement à la durée de la tâche, il nous paraît très rapide.

### 3-2. Résultats de simulation : application de traitement de surface

Cette application diffère de la précédente du fait qu'elle contient plusieurs activités qui s'exécutent en parallèle, les traitements dans les bacs ainsi que les déplacements du robot entre les différents postes de traitement. Nous remarquons dans le diagramme représenté par la figure 52 que les tâches de surveillances de chacune des activités sont actives simultanément. Les caractéristiques de l'atelier sont représentées dans le chapitre 3 §3, à savoir les durées de traitement et les durées de déplacement du robot entre les différents postes.

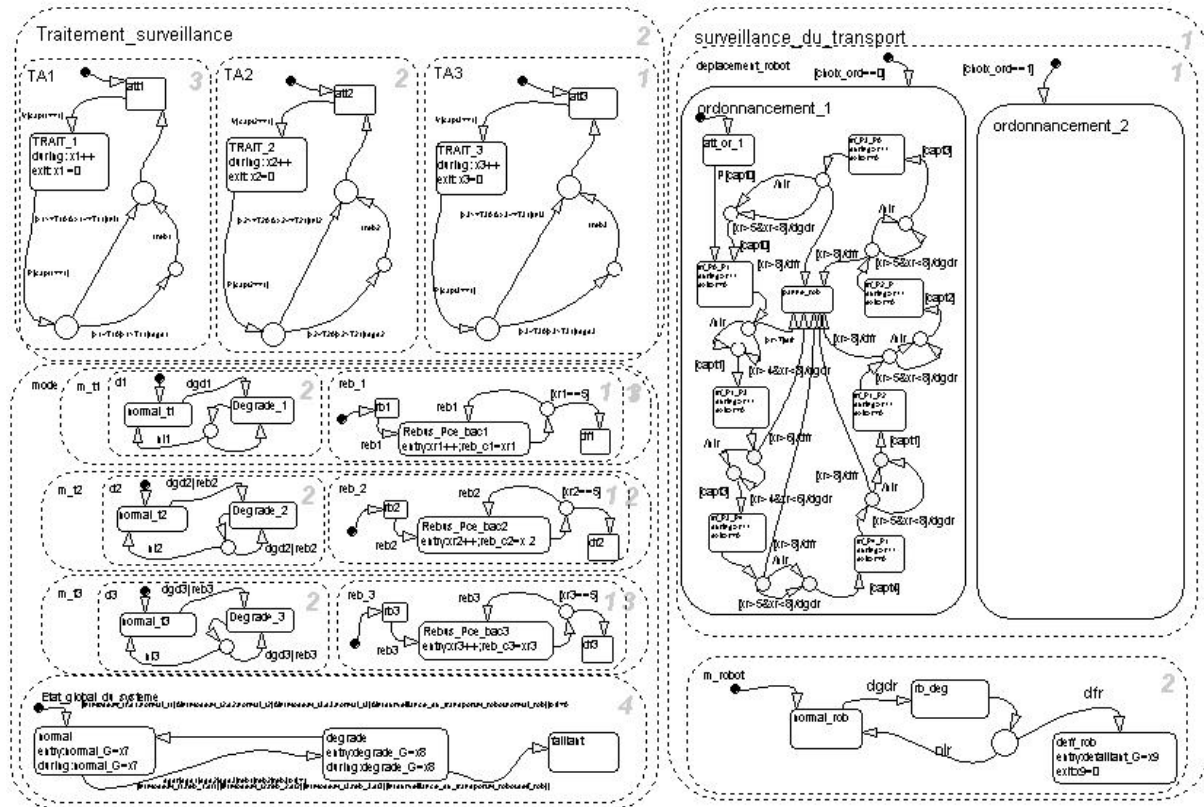


Fig. 52. Modèle de surveillance de l'atelier de traitement de surface.

L'ensemble des simulations a été fait avec les paramètres suivants

$$\delta_{transport} = 2 \text{ secondes entre deux postes successifs ;}$$

$$\delta_{t\grave{a}che1} = 12 \text{ secondes ;} \quad \delta_{t\grave{a}che2} = 4 \text{ secondes ;} \quad \delta_{t\grave{a}che3} = 12 \text{ secondes ;}$$

#### 3.2-1. Calcul de la valeur du seuil $\Delta_{opt}$

De même que pour la première application, nous procédons à l'optimisation de la tolérance dans la durée de traitement dans chacune des tâches du système. Chacune prise indépendante puis un vecteur ayant les seuils optimaux comme composantes pour le système global à surveiller. Nous choisissons par exemple, pour illustrer les résultats de l'algorithme, de calculer le seuil  $\Delta_{opt}$  pour la tâche de transfert du robot. Sachant qu'elle dure normalement 3 secondes.

Pour la surveillance des opérations de transport, nous surveillons ici le déplacement entre deux postes successifs. Le tableau suivant montre les résultats de simulation obtenus:

<b>Intervalle de tolérance <math>\Delta</math>(s)</b>	<b>Rapport : <math>\Delta / \delta</math> en %</b>	<b><math>N_{\text{injectés}}</math></b>	<b><math>N_{\text{détectées}}</math></b>	<b><math>N_{\text{alarmes}}</math></b>
1.20	60	42	3	39
1.08	54	42	4	38
0.92	46	41	5	36
0.83	41,5	40	6	34
0.75	37,5	38	7	31
0.69	34,5	37	8	29
0.63	31,5	38	10	28
0.56	28	40	13	27
0.5	25	40	15	25
0.45	22,5	40	16	24
0.4	20	40	17	23
0.36	18	41	20	21
0.2	10	41	27	14
0.1	5	43	38	5

*Fig. 53. Résultats de simulations : Alarmes en fonction du retard  $\Delta$ .*

$N_{\text{injectés}}$  = Nombre total de défauts injectés ;       $N_{\text{détectées}}$  = Nombre fautes détectées ;  
 $N_{\text{alarmes}}$  = Nombre de fausses alarmes ;

**Remarque :**

Toutes les valeurs relatives aux nombres d'erreurs qui figurent dans le tableau ci-dessous sont des valeurs moyennes des différents résultats de simulation.

De même que pour la première application, nous représentons ces résultats sous la forme de courbes d'erreurs. Ceci nous permet de situer la valeur du seuil de tolérance ; ce qui correspond au compromis entre le moins possible de fausses erreurs tout en détectant toutes les défaillances possibles du système à surveiller.

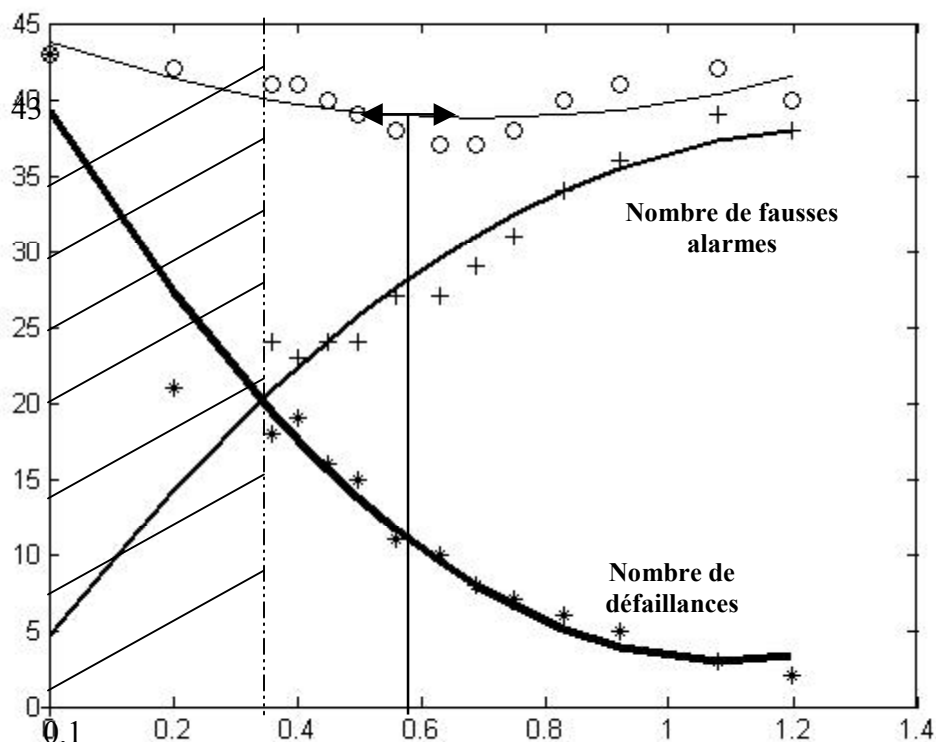


Fig. 54. *Détermination du seuil de tolérance.*

D'après les courbes de la figure ci-dessus, nous pouvons choisir la valeur du seuil de tolérance égal à 0,6 secondes.

Une fois la valeur du seuil de tolérance optimale obtenu, nous allons chercher les limites de validité de cette valeur par rapport au temps de service. Pour cela, nous faisons varier le temps de service en fixant la valeur du seuil à celle choisie et supposée être optimale. Nous relevons les nombres de détections, qu'elles soient fausses ou effectives, les résultats sont présentés dans le tableau suivant :

Rapport des temps $\Delta/\delta$ en [%]	Nombre moyen des défauts injectés ' $N_i$ '	Nombre moyen des défauts détectés ' $N_d$ '	Taux de couverture	classification
10	21	20,37	0,97	bonne
20	19	17,48	0,92	
25	20,5	18,65	0,91	
30	19,66	17,30	0,88	
40	20,33	11,99	0,59	Moyenne
45	18,75	9,00	0,48	
50	19,67	6,88	0,35	Mauvaise
55	20	6,60	0,33	
60	22,33	2,06	0,28	

Fig. 55. *Classification de la surveillance en fonction du seuil de tolérance.*

### 3.2-2. Détermination du temps de réponse

Nous déterminons le temps de détection pour les valeurs retenues, à savoir le seuil de tolérance  $\Delta = 0,6$  secondes et la durée de la tâche de déplacement du robot est de 2 secondes.

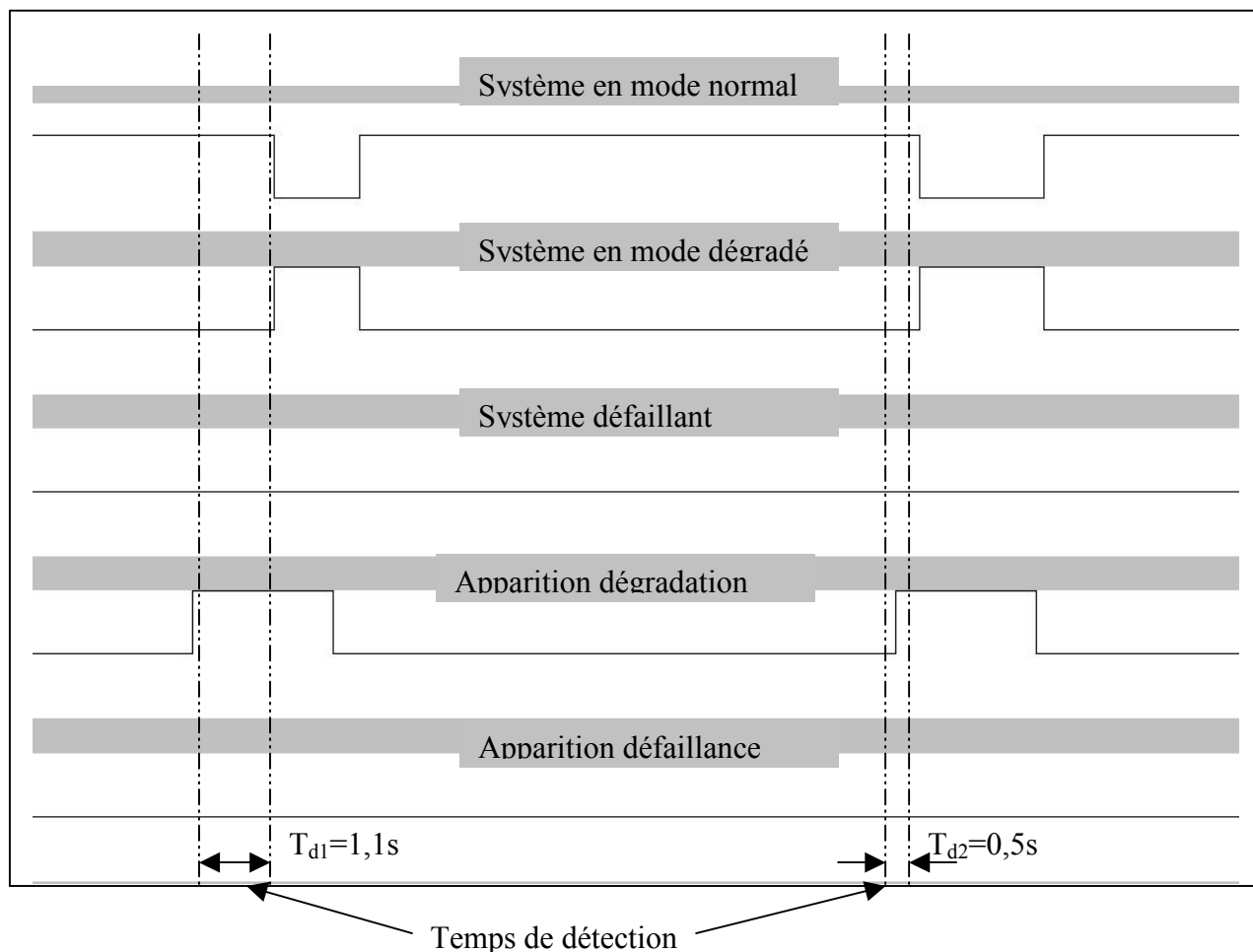


Fig. 56. *Détermination des temps de réponse du système de surveillance*

**Conclusion** : d'après les résultats du tableau représenté par la figure 55, nous concluons que le seuil choisi, de valeur égale à 0,6 secondes correspond à une bonne tolérance et ce pour des tâches qui durent un temps de service correspondant au rapport  $\Delta/\delta$  de 40%, c'est à dire  $\delta = 1,5s$ .

Pour les résultats de simulation, représentés par la figure 56, nous avons relevé les temps de réponse,  $T_{d1}=1,1s$  et  $T_{d2}=0,4s$ . nous attirons l'attention sur le fait que lorsqu'il s'agit des temps de service courts, nous rencontrons des problèmes au niveau des temps de détection. Ici nous obtenons deux temps pour la même tâche qui des fois passe du simple au double.

## APPLICATION AUX SYSTEMES DE PRODUCTION

*Dans cette dernière partie, nous allons procéder à l'application des modèles de surveillance développés pour un système manufacturier. Il s'agit de l'application à différentes structures systèmes de production. Nous commençons alors par définir les différentes composantes de ce type de système. Puis un modèle de surveillance sera proposé utilisant l'outil de description et de simulation stateflow. L'algorithme d'optimisation des temps de détection des défaillances, est alors appliqués afin de pouvoir choisir le retard toléré pour la bonne conduite et le bon fonctionnement du système.*



## 1. SYSTEME DE PRODUCTION

Produire c'est transformer. Le lieu et les moyens de cette transformation, c'est le système de production 'SdP'. Cette transformation, de produits peu élaborés en produits plus élaborés, prend place dans un processus de production appelé usine ou encore atelier manufacturier. Un processus de production est généralement composé d'un grand nombre d'opérations ou de transformations entrées/sorties organisées en réseaux. Ces opérations assurent des transformations de forme (modifications de produits eux-mêmes), des transformations dans le temps (fonction de stockage) ou dans l'espace (fonction de transport). Les systèmes de production, diffèrent par les objectifs que s'assigne le producteur. Ils diffèrent par les attributs des objets transformés et par les caractéristiques des processus de production. Ils ne sont pas statiques et ils évoluent sous les effets conjugués de l'évolution des marchés, des technologies et des sociétés dans lesquelles ils sont insérés. Ils se définissent par des attributs de sorties : coût du produit assemblé, qualité de l'assemblage, quantité de produits par unité de temps, délai moyen de livraison d'un produit (ou temps de service), diversité des modèles susceptibles d'être assemblés au sein du processus existant, ...etc.[SAS-98]

### 1-1. Définitions

Formellement, un système de production, noté : 'SdP' est un triplet  $S = \langle M, S, [A] \rangle$  tel que :

$M = \{M_1, M_2, M_3, \dots, M_n\}$  est un ensemble fini et non vide de machines ;

$S = \{S_1, S_2, S_3, \dots, S_m\}$  est un ensemble fini et non vide de stocks ;

$[A] : M \times S \rightarrow \{-1, 0, 1\}$  est l'application d'agencement des machines et stocks.

□

Un système de production noté : 'SdP' est formé par des produits, des moyens de production et des opérateurs. Les moyens de production peuvent être répartis en deux catégories : les **machines** ; utilisées pour la production et le transport, et les **stocks** utilisés pour le magasinage.

Les machines peuvent être spécialisées, à changement d'outil, des robots, des convoyeurs, des chariots de manutention, ...etc. Une machine spécialisée est conçue pour traiter spécifiquement une catégorie de produits. Sa particularité est que, dès l'instant où le produit est engagé dans la machine, il y subit normalement une transformation sans temps mort jusqu'à la sortie. L'activité d'une machine à changement d'outil est faite de campagnes de lancement au cours desquelles sont produites suffisamment de pièces d'une référence donnée, pour couvrir les besoins sur une période de consommation déterminée, avant un changement d'outillage adapté pour un autre type de produits. Un robot industriel est par vocation un moyen de production polyvalent, adaptable et programmable. Il est capable d'exécuter plusieurs tâches (de nature pas forcément identique) sur des produits qui se présentent successivement à son entrée. Le transport des charges (ou produits ou pièces) s'effectue à

l'aide de chariot de manutention : avec conducteur, automoteurs filoguidés, optoguidés, radioguidés, ...etc. Une machine spécialisée ou à changement d'outil, effectue des modifications (de forme) sur les produits eux-mêmes alors qu'un robot, un convoyeur ou un chariot de manutention joue le rôle d'une fonction de transport (dans l'espace) [ROD-89].

Le stockage est représenté par un endroit réservé (bacs, palettes,...etc.) ayant la possibilité de contenir un nombre limité de produits. Il permet de conserver des matières, composants ou produits finis pendant une certaine durée sans transformer leurs états. Un produit en cours de stockage est en arrêt structurel. Les capacités et moyens de stockage influencent considérablement les performances d'un système de production. Ces stocks permettent de protéger le système de production des phénomènes aléatoires qui se produisent sur le marché et de régulariser les flux et les charges.

## 1-2. Les structures de systèmes de production

L'exécution d'une quelconque opération nécessite une machine ou un endroit aménagé spécifiquement. Un poste de travail est défini par une activité (fabrication, contrôle, manutention,...etc.), un aménagement, un outillage nécessaire et une description de la compétence de l'opérateur. Le premier souci d'un chef de production, dans les années cinquante, était de réaliser des gains de productivité. Pour cela, le premier réflexe a été de constituer des lignes de fabrication dans lesquelles les opérations de montage, de démontage, d'usinage et de manutention s'enchaînent de façon automatique.

Une **ligne de fabrication** ou de transfert est composée d'une succession de machines séparées éventuellement par des stocks. Dans ce type de système, une pièce brute provenant de l'extérieure passe successivement sur toutes les machines constituant la ligne dans un ordre immuable. La recherche de gain de productivité sans perte de flexibilité a permis le développement des machines outils à commande numérique. Dans les dernières décennies, les conditions de la production industrielle ont été bouleversées par de profondes mutations d'ordre économiques (par exemple, diversités des produits) technologiques (électroniques, informatique,...etc.), sociales,...etc. le dilemme fondamental entre productivité d'une part, flexibilité et innovation d'autre part, est remis en cause par les technologies flexibles qui permettent l'automatisation de la diversité et de la complexité.

Dans la littérature, divers types d'ateliers de production ont été introduits :[DAN-95]

a) **Atelier de type série** :

Dans ce type d'atelier, chaque pièce doit obligatoirement passer sur toutes les machines de la ligne de production. Les machines étant séparées par des stocks intermédiaires. Ceci en plus des stocks d'entrée et de sortie, comme le montre la figure 48.

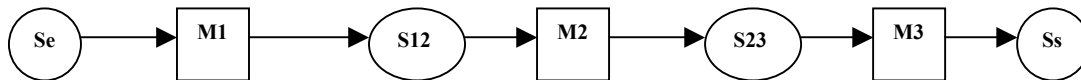


Fig. 57. *Système de production de type série.*

b) **Atelier de type assemblage** :

L'assemblage est une opération fréquemment rencontrée dans un système de production manufacturier. Cette opération consiste à regrouper n pièces (nombre  $n > 1$ ) puisées dans n stocks différents afin d'en faire un seul objet qui sera déposer dans un stock unique. Nous pourrons trouver ce type de système sous plusieurs variantes, comme le montre la figure 49.

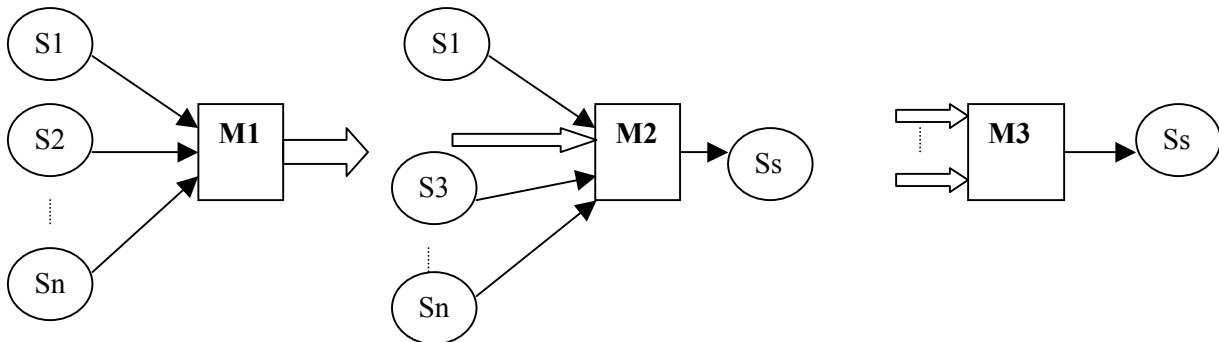


Fig. 58. *Système de production de type assemblage.*

c) **Atelier de type désassemblage** :

Ce type de système est rencontré par exemple dans l'industrie automobile. Les carrosseries sont peintes et séchées complètes, c'est à dire avec les portières, le coffre et le capot. Par la suite, pour des impératifs de productivité et d'accessibilité, les portières sont séparées de la caisse et sont équipées sur une ligne parallèle. Cette séparation est un désassemblage. On pourra trouver différentes variantes, comme indiquée par la figure 50.

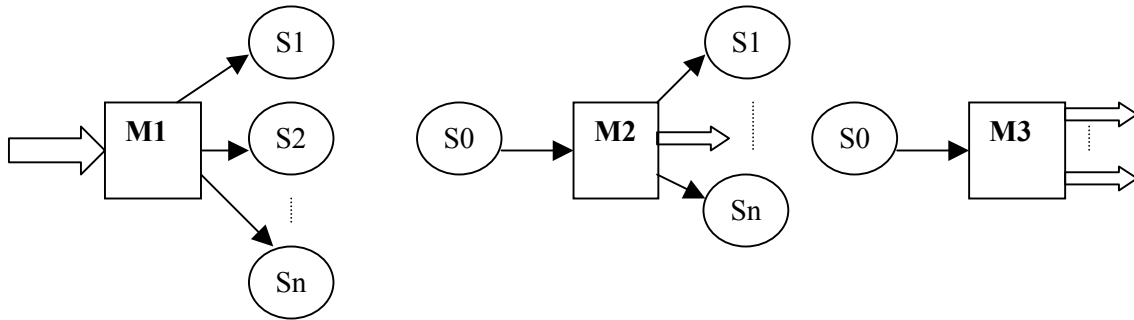


Fig. 59. *Système de production de type désassemblage.*

Il existe aussi d'autre type de structures combinaisons de celles que nous venons de présenter. Des ateliers qui réalisent des opérations de transformations en parallèle ou de synchronisation, telles que des divergences stock-machine ou des convergences machine-stock.

## 2. MODELE GENERIQUE : MACHINE-STOCK

La figure 51 présente le modèle du système de production, de type série, auquel nous appliquons notre approche de surveillance puis l'algorithme d'optimisation des temps de détection des défaillances que nous avons présenté dans le chapitre 4.

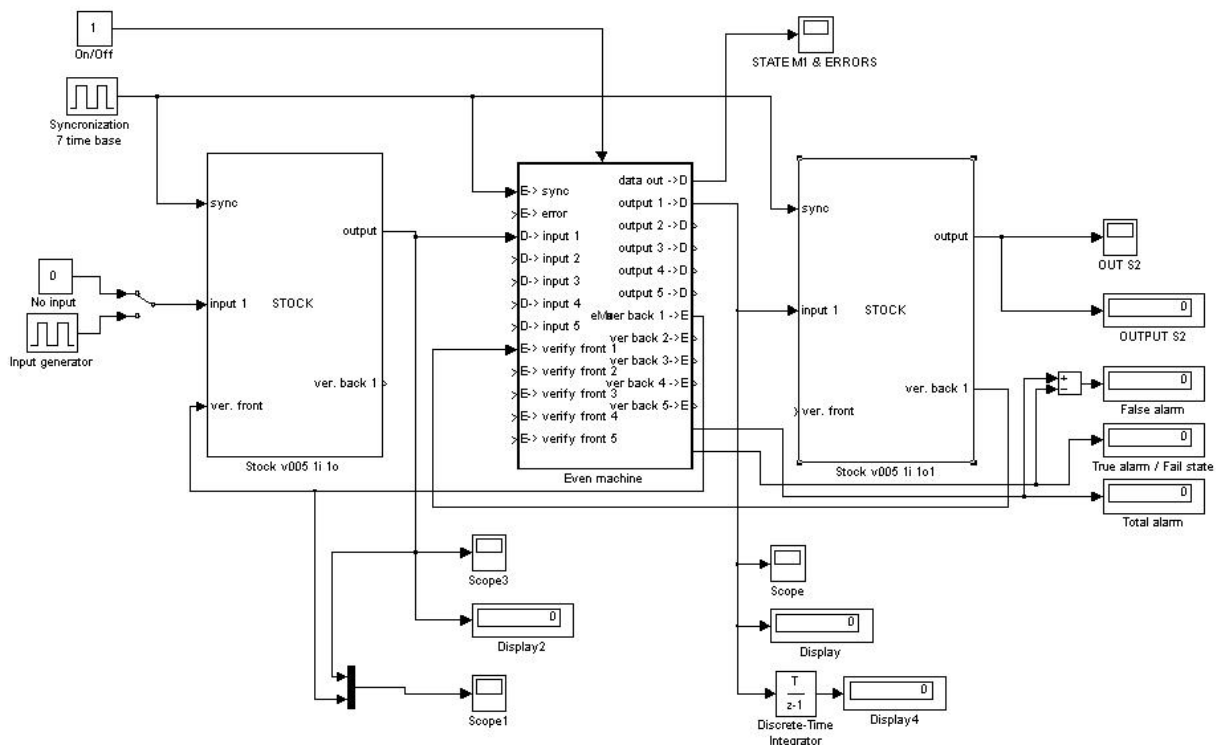


Fig. 60. *Structure d'analyse d'un système de production de type série.*

Notons que les caractéristiques des stocks n'ont aucune influence sur l'analyse que l'on se propose de faire, dans l'objectif est l'optimisation des temps de détection des défaillances. Il suffit que le stock d'entrée puisse mettre à la disposition de la machine des pièces sans interruption (valeur de stock supposé infinie). De même que pour le stock de sortie, il devrait avoir la capacité suffisante pour recevoir toutes les pièces usinées par la machine.

## 2-1. Présentation du modèle de surveillance

Pour faire ce travail, nous devons choisir des paramètres de la machine, tels que le temps d'usinage, le temps de préparation, ...etc. nous injectons de façon aléatoire des défauts, au niveau des temps d'usinage, et nous comparons leur nombre à celui des défaillances détectées par le système de surveillance. La différence entre ces deux nombres correspond au nombre de fausses alarmes (basculement successif mode normal, dégradé puis retour au mode normal. Le critère d'évaluation, comme nous l'avons vu dans les chapitres 2 et 3, consiste à tolérer un retard pour l'arrivée de l'événement indiquant la fin d'exécution de l'opération d'usinage. Enfin, nous procédons à l'optimisation du temps de détection des défaillances en choisissant la valeur optimale de la tolérance notée  $\Delta$  et ce pour chaque opération du système de production.

La modélisation du système de surveillance en utilisant l'outil statecharts est donnée par la figure 52 (seule représentée ici la boucle principale de surveillance). notons que tout le raisonnement qui suit est appliqué à une tâche donnée du système de production. Nous retrouvons bien les cinq situations possibles pour une machine, dans le cas général des systèmes de production manufacturiers, à savoir ; état d'**attente**, état de **préparation**, état de **fonctionnement normal**, état de **fonctionnement dégradé** ou état de **défaillance**.

Initialement, le système se place dans l'état de **préparation** à la production, dès que la préparation est finie le système passe à la phase de production en **mode de fonctionnement normal**. Si l'événement noté '**prod\_ok**' se produit, cela signifie que le temps d'usinage est compris dans l'intervalle, dit de bon fonctionnement et précédemment noté :  $I_m$  relativement à une tâche m, alors le système revient à l'état de **préparation** pour une nouvelle opération d'usinage. Sinon, lorsque le temps d'usinage dépasse la borne supérieure de  $I_m$  noté :  $T_{\max}^m$ , alors un événement est généré par le système de surveillance noté 'degrad' et permettant de faire basculer le système en mode **dégradé**. Ensuite, le système revient dans l'état de préparation si l'événement noté : 'false\_alarm' est généré ; ceci signifie que l'opération d'usinage est terminée pendant l'intervalle de tolérance, et par suite correspond à une fausse alarme. Dès que le délai de tolérance est épuisé, le système bascule vers l'état de défaillance, dans ce cas il s'agira d'une alarme effective. Après réparation, le système peut reprendre le fonctionnement en se plaçant dans l'état de préparation.

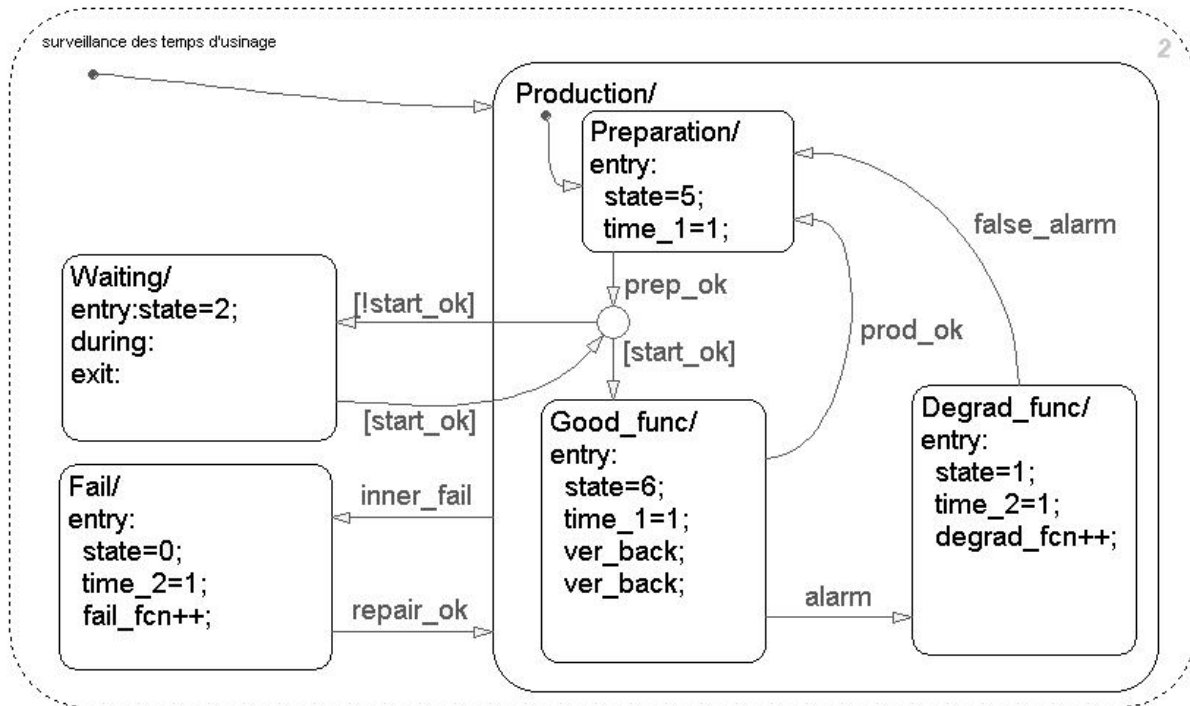


Fig. 61. Modèle de surveillance générique pour la durée d'une tâche

## 2-2. Simulation : résultats et analyses

L'objectif de ces simulations étant de déterminer le délai optimal que l'on peut tolérer pour recevoir le signal indiquant la fin d'exécution d'une tâche d'usinage, et ceci délimite l'instant au-delà duquel le système peut être considéré en état de défaillance.

Nous avons choisi les caractéristiques suivantes, afin de déterminer la valeur optimale du seuil de tolérance :

- Temps d'usinage, noté  $\delta = 50$  secondes.
- Délai de tolérance, noté  $\Delta$  variant de 0 à 10 secondes.
- Quantité de pièces à l'entrée du système : 100 pièces.

### 2.2-1. détermination de seuil de tolérance

Le temps d'usinage ' $\delta$ ' reste constant puisqu'il s'agit de surveiller une tâche donnée. En faisant varier le délai de tolérance ' $\Delta$ ' (i.e. rapport  $\Delta/\delta$ ), et pour le même nombre de pièces traitées, nous relevons le nombre de défauts injectés, le nombre de pièces obtenues à la sortie. En même temps, nous relevons le nombre de total de détection et le nombre de passage en mode de défaillance, leur différence nous donne le nombre de fausses alarmes. Ceci est présenté sous forme du tableau de la figure 53.

Valeur de la tolérance noté : $\Delta$	Temps de production noté : $\delta$	Rapport $\Delta/\delta$ En%	Nombre de défauts injectés	Nombre de détection	Nombre d'alarmes			
10	50	20	53	53	0			
			44	44	0			
			51	51	0			
9		50	18	54	54	0		
				51	50	1		
				40	38	2		
8			50	16	53	48	5	
					49	47	2	
					42	40	2	
7				50	20	45	41	4
						45	41	4
						41	36	5
6	50				20	53	47	6
						52	41	11
						39	29	10
5		50			10	50	37	13
						50	40	10
						47	34	13
4			50		8	48	40	8
						44	35	9
						41	30	11
3				50	6	57	41	16
						50	33	17
						45	33	12
2	50				4	48	34	14
						54	31	23
						43	24	19
1		50			2	55	36	19
						51	30	21
						45	31	14
0			50		0	55	0	55
						53	0	53
						48	0	48

Fig. 62. Résultats de simulation : Cas d'une ligne de production série.

A partir de ce tableau, nous avons tracé les courbes de la figure 54 qui illustrent l'influence du rapport  $\Delta/\delta$  sur le nombre des alarmes qu'elles soient fausses ou effectives.

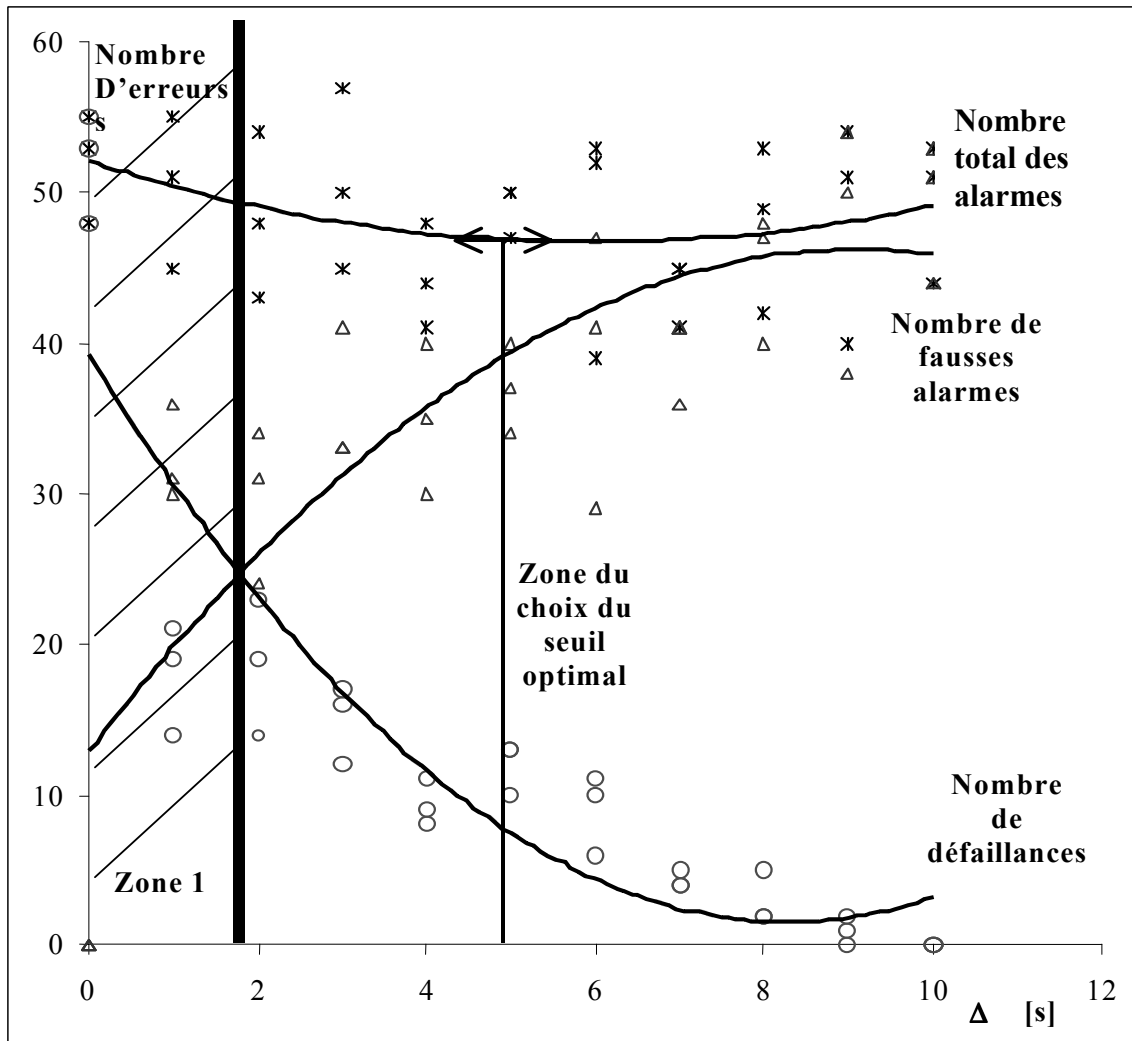


Fig. 63. Représentation graphique : nombre d'erreurs en fonction du seuil de tolérance.

D'après les courbes de la figure 63, nous constatons que lorsque le seuil  $\Delta$  est petit (ici de l'ordre de 1,8 secondes) nous aurons un nombre très important de fausses alarmes, ceci correspond sur le graphe à la zone 1. Au-delà du point d'intersection entre les courbes qui représentent successivement le nombre de fausses alarmes et celui des défaillances, nous pouvons choisir un compromis qui nous semble adapté au type du système de production à surveiller. Pour cet exemple, nous choisissons le seuil  $\Delta = 5$  s, qui correspond au minimum de la courbe qui représente la somme des défaillances et des fausses alarmes.

### 2.2-3. Evaluation des performances de la détection

Nous allons maintenant chercher les limites entre lesquelles le seuil calculé restera applicable. Pour cela nous allons faire varier le temps de service tout en gardant le seuil de tolérance calculé précédemment et choisi égal à 5 secondes. Ceci a été appliqué à la même ligne de production de type série présentée au début de ce paragraphe. Nous obtenons alors les



résultats contenus dans le tableau de la figure 64.

Rapport des temps $\Delta/\delta$ en [%]	Nombre moyen des défauts injectés ' $N_i$ '	Nombre moyen des défauts détectés ' $N_d$ '	Taux de couverture
5	42,33	41,67	0,98
10	41,33	39,67	0,96
15	39,67	35,33	0,89
20	36,67	32,00	0,80
25	41,33	32,67	0,78
30	40,33	31,00	0,76
35	42,00	29,33	0,69
40	42,00	27,67	0,65
45	43,00	26,67	0,61
50	44,00	25,33	0,57

Fig. 64. *Résultats de simulation: Cas du seuil fixe, temps de service variable.*

Ces mêmes résultats sous forme d'une courbe, qui représente le taux de couverture en fonction du rapport  $\Delta/\delta$ , sont illustrés sur la figure 65. Ces résultats confirment bien que le taux de couverture soit étroitement lié au temps de service des tâches à surveiller. D'après le tableau de classification, la valeur de  $\Delta = 5s$  était bonne et nous assure la bonne qualité de surveillance mise en œuvre.

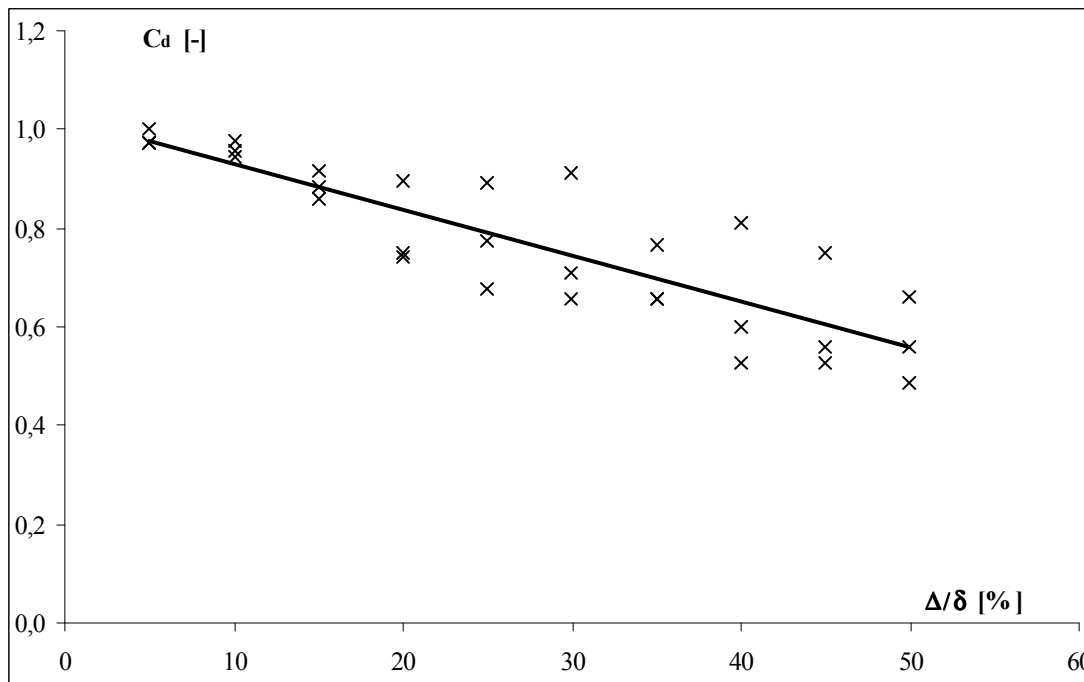


Fig. 65. *Relation entre taux de couverture et seuil de tolérance.*

---

## *CONCLUSION GÉNÉRALE*

L'objectif du travail présenté dans ce mémoire est de développer une nouvelle approche de surveillance des systèmes à événements discrets. Pour atteindre cet objectif, nous avons proposé une approche basée sur la surveillance du temps de service

Dans les travaux effectués ces dernières années, toutes les approches de surveillance sont basées sur le principe de filtre et ou de comparateur. La capacité de modélisation et d'analyse de l'outil automate temporisé a permis d'intégrer le caractère temporel du procédé dans le modèle de surveillance que nous avons proposé.

Partant d'un système de production qui est supposé commandé, nous nous sommes intéressés à la surveillance des différentes activités du système. Cette démarche revient à vérifier que toutes les activités sont exécutées dans délais préalablement définis. L'objectif premier de notre travail est l'amélioration de la disponibilité du système. Ceci est traduit par la minimisation du nombre d'arrêts qui pénalisent la production en suivant en temps réel, l'état de fonctionnement des différents capteurs du système. Ainsi, pour chaque tâche nous surveillons la durée entre l'instant où la commande a donné l'ordre de son exécution et l'instant où le capteur a indiqué la fin d'exécution. De ce fait, nous pouvons dire que lorsque le temps écoulé entre ces deux événements n'est pas respecté, qu'il y a risque de défaillance. Il s'ensuit, une opération de diagnostic, qui consiste à déterminer les causes du problème observé. Dans le cas d'une défaillance intempestive causé soit par le capteur ou par l'organe responsable de l'exécution de la tâche surveillée des alarmes sont déclenchées.

Dans un premier temps, nous construisons le modèle dynamique qui correspond à la commande du système, celle-ci est supposée parfaite. La commande appliquée peut être sous forme de grafset ou d'ordonnancement de séquences d'opération bien définies. Pour mettre en évidence la flexibilité de la démarche proposée et la puissance de l'outil de modélisation choisi, nous avons pris deux exemples d'application. Dans l'un nous partons d'un modèle de commande sous forme de grafset et dans l'autre c'est un modèle de commande sous forme d'ordonnancement des tâches que nous avons utilisé. Le modèle dynamique du système à surveiller est mis en œuvre en utilisant l'outil automate temporisé, qui est bien adapté à la surveillance du temps de service. Le modèle dynamique que nous avons développé présente le point important dans notre modèle de surveillance : **Pour la détection d'éventuelles défaillance, il n'est pas nécessaire d'avoir la liste exhaustive de toutes les défaillances possibles** du système. Jusqu'à présent toutes les approches existantes utilisent un modèle de

---

référence qui nécessite la connaissance à priori de toutes les situations interdites du système.

Pour construire le modèle de surveillance global du système, nous avons introduit ce que nous avons appelé « retard de tolérance », auquel nous avons fait correspondre un seuil appelé ‘seuil de tolérance’. Ceci traduit le fait que le système de surveillance ne doit décider de l’existence d’une défaillance que lorsque la fin d’exécution d’une tâche est réellement dépassée. Nous introduisons un retard, qui est fonction du type du système de production et de la nature de la tâche surveillée, pendant lequel le système sera considéré fonctionnant en mode dit « **dégradé** ». Puis au-delà de ce retard, le système sera effectivement dans un **état de défaillance**. Les exemples étudiés illustrent bien la difficulté à choisir ces seuils de tolérances.

Nous avons construit le modèle de surveillance global pour le premier exemple, et nous nous sommes rendu compte d’un problème au niveau de l’explosion du nombre d’arcs, entre les différents états existant sous les trois modes de fonctionnement. De ce fait nous avons utilisé un outil structurel adapté à la description et la simulation des systèmes à événement discrets ; c’est **l’outil stateflow**. Nous avons vu sur le deuxième exemple, qu’à partir de la commande sous forme d’ordonnancement et malgré la complexité du système ( un atelier ayant plusieurs activités en parallèles), la modélisation avec cet outil est plus souple et pratique. Nous avons mis en œuvre le modèle de surveillance correspondant, en passant par le modèle dynamique des différentes parties qui composent le système. Ce deuxième outil présente la solution aux problèmes d’explosion du nombre d’arc soulevé en utilisant les automates temporisés. Grâce à sa structure hiérarchique cet outil, présente une grande lisibilité et reste bien adapté pour la fonction de diagnostic.

Nous nous sommes intéressés par la suite à l’optimisation des seuils de tolérance des défaillances. Un algorithme général pour cette optimisation est présenté. Ensuite, lorsque le seuil optimal est validé, alors nous calculons les temps de détection des défaillances. Une évaluation des performances du système de surveillance a permis de valider ce seuil.

En résumé, notre approche de surveillance basée sur la surveillance des temps d’activités de toutes les tâches qui constituent le système, présente les avantages suivant :

- **intègre le caractère temporel** que la commande n’en dispose pas.
- **ne nécessite pas la connaissance de la liste exhaustive des défaillances** possibles.
- **permet au système de fonctionner en mode dégradé** au lieu de basculer directement en mode défaillant, ceci minimise les temps d’arrêts de la production.

Les perspectives que nous envisageons pour notre travail de recherche sont les suivantes :

1- Une perspective immédiate pour la continuation du travail de recherche présenté dans ce mémoire est la recherche d'algorithmes d'optimisation plus adaptés aux calculs des seuils de tolérance, ainsi qu'une méthode générique pour le calcul de ces seuils en fonction des temps de service et/ou en fonction du type du système de production à surveiller. De plus, le développement informatique de la méthode de calcul des seuils de tolérance rendrait moins ardu ce calcul. Nous nous orientons vers la mise en œuvre d'un programme permettant l'obtention des seuils de tolérance et l'évaluation des performances du système de surveillance.

2- Une deuxième perspective qui nous paraît primordial, c'est la deuxième fonction contenue dans le module de surveillance retenu dans notre structure globale de surveillance proposée. Il s'agit de la fonction de diagnostic, nous avons vu que notre modèle de surveillance permet de tracer facilement le chemin qui conduit vers un état de défaillance, jusqu'à présent considéré comme un état global, du fait que nous nous sommes limités à la détection et l'optimisation des alarmes.

3- Une de nos préoccupations à cours terme, est la mise en œuvre de notre démarche pour la surveillance en temps réel de la cellule flexible de l'EMI. En effet, cette cellule présente, par la diversité des postes et des opérations exécutées une bonne application de notre système de surveillance qui pourra s'étendre vers un suivi à distance des différentes activités.

**REFERENCES BIBLIOGRAPHIQUES**

- [ADL-94] R. ALUR and D. L. DILL « A theory of timed automata » Theoretical Computer Science, 126:183-235, année 1994.
- [ALA-03] A. ALAMIN, « Mise en oeuvre d'un système de surveillance de procédé: optimisation du temps de retard » rapport de DEA Automatique-Productique, LAG / ENSIEG, septembre 2003.
- [ALN-86] P. ALANCHE, P. LHOSTE, G. MOREL, M. ROECH, M. SALIM et P. SALVI « Application de la modélisation de la partie opérative à la structuration de la commande » Journées AFCET, Montpellier, France, mars 1986.
- [ALU-99] R. ALUR « Timed automata » 11<sup>th</sup> International Conference on Computer aided verification, L N C S, 1633 : 8-22, année 1999.
- [BEN-97] Z. BEN AZZOUZ, « optimisation des temps de détection » Rapport de DEA Automatique-Productique, LAG / ENSIEG, juin 1997.
- [BER-95] E. BERNAUER, H. DEMMOU, « Les méthodes de détection, localisation et diagnostic » Rapports LAAS, année 1995.
- [CHA-95] A. CHAILLET-SUBIAS, « Approche Multi-modèles pour la commande et la surveillance en temps réel des systèmes à événements discrets. » Thèse de doctorat de l'INPT, année 1995.
- [COM-91] M. COMBACAU, « Commande et Surveillance des systèmes à événements discrets complexes » Thèse de l'université Paul Sabatier de Toulouse, déc. 1991.
- [COM-98] M. COMBACAU, « Contribution à la surveillance hiérarchisée des systèmes complexes. » Habilitation à Diriger les Recherches, année 1998.
- [DAN-95] O. DANIEL, « les réseaux de Petri stochastiques pour l'évaluation des attributs de la sûreté de fonctionnement des systèmes manufacturiers » Thèse de doctorat de l'INPG, année 1995.
- [DAV-98] R. DAVID, H. ALLA « Du grafcet aux réseaux de Petri » traité des nouvelles technologies série Automatique édition Hermès, année 1998.
- [DUB-90] B. DUBUISSON, « Diagnostic et reconnaissance des formes », traité des nouvelles technologies, série Diagnostic et maintenance, édition Hermès, année 1990.

- [GOU-99] A. GOUIN, J. L. FERRIER « Commande supervisée temporisée » GRP'99.
- [HOL-90] L. E. HOLLOWAY et B.H. KROGH « Fault detection and diagnosis in manufacturing systems: a behavioral model approach » IEEE, International Conference on Computer Integrated Manufacturing, Mai 1990.
- [HOL-91] L. E. HOLLOWAY et B.H. KROGH « Monitoring behavioral evolution for on-line fault detection » IFAC / IMACS International Conference on Fault Detection, Supervision & Safety for Technical Process, SAFEPROCESS'91? Baden Baden, Germany, Septembre 1991.
- [KNO-03] Michal KNOTEK, « Robustness study of production systems: Application to surface treatment workshop » année 2003.
- [LEF-00] D. LEFEBVRE, « Contribution à la modélisation des systèmes dynamiques à événements discrets pour la commande et la surveillance. » Dossier habilitation, année 2000.
- [NID-96] R. NIDHAL, « Contribution à la sécurité opérationnelle des systèmes : Mise en œuvre d'une structure de surveillance basée sur les réseaux de Petri objets contrôlés. » Thèse de doctorat de l'INSA, année 1996.
- [NOU-97] M. NOURELFATH, « Extension de la théorie de la supervision à la surveillance et à la commande des systèmes à événements discrets : application à la sécurité opérationnelle des systèmes de production. » Thèse de doctorat de l'INSA, 1997.
- [OLI-94] A. OLIVERO, « Modélisation et analyse des systèmes temporisés et hybrides » Phd thesis, VERIMAG-INPG, année 1994.
- [PET-92] R. DAVID, H. ALLA, « Du grafcet aux réseaux de Petri » Livre, 1992 Edition: Hermès.
- [RAM-89] JG. RAMADGE, WM. WONHAM « The control of discrete event systems » IEEE, vol.77, N° 1, Janvier 1989, pages 81-89.
- [RAY-03] H. RAYHANE, Z. SIMEU-ABAZI, T. BENNANI « Surveillance des systèmes de production par automates temporisés » Alger, CIP'2003.
- [RIV-00] M. RIVOIRE, J. -L. FERRIER, « Matlab Simulink et Stateflow » Edition TECHNIP 2000.
- [ROD-89] G. RODDE, « Les systèmes de production, modélisation et performances » Edition Hermès, Paris 1989.

- [SAH-92] A.E.K. SAHRAOUI « Vers une approche globale de surveillance des systèmes à événements discrets », RAIRO, APII, Vol. 26, n° 2, pp 91-106, année 1992.
- [SAR-99] P. SARRI, « Stabilisation optimale des systèmes à événements discrets à structure vectorielle : application à la sécurité opérationnelle des systèmes de production. » Thèse de doctorat, année 1999.
- [SAS-98] C. SASSI, « Intégration des politiques de maintenance dans les systèmes de production manufacturiers » Thèse de doctorat de l'INPG, année 1998.
- [SAV-01] A.T SAVA, « sur la synthèse de la commande des systèmes à événements discrets temporisés » thèse de doctorat, année 2001.
- [SIM-98] Z. SIMEU-ABAZI « Surveillance des temps de service d'une ligne série » Rapport interne n°98-162.
- [SIM-03] H. RAYHANE, Z. SIMEU-ABAZI, T. BENNANI « optimisation des temps de détection dans la surveillance des systèmes de production » Montréal, congrès génie industrielle 2003.
- [TOG-92] A. K. A. TOGUYENI, « Surveillance et Diagnostic en ligne dans les ateliers flexibles de l'industrie manufacturière » Thèse de doctorat de l'université des Sciences et Techniques de Lille, nov. 1992.
- [VAL-00] R. VALETTE, E.NIEL, « Supervision, surveillance d'ateliers de fabrication. » Rapports LAAS, année 2000.
- [YOV-93] S. YOVINE « Méthodes et outils pour la vérification symbolique des systèmes temporisés » Phd thesis, VERIMAG- INPG, année 1993.
- [ZAM-97] E. ZAMAI, « Architecture de Surveillance-Commande pour les systèmes à événements discrets complexes. » thèse de doctorat, année 1997
- [ZWI-95] G. ZWINGELSTEIN « Diagnostic des défaillances », traité des nouvelles technologies série diagnostic et maintenance. Edition Hermès 1995.

## Résumé

Le but de la thèse est de développer une nouvelle approche de surveillance des systèmes à événements discrets. La méthode proposée est basée sur la surveillance du temps de service en utilisant l'automate temporisé comme outil de modélisation.

Partant d'un système commandé, le premier objectif est la surveillance des différentes activités de ce système afin d'améliorer sa disponibilité. Ceci se traduit par la minimisation du nombre d'arrêts qui pénalisent la production, en suivant en temps réel, l'état de fonctionnement des différents capteurs du système. Ainsi pour chaque tâche, la surveillance du temps écoulé entre deux événements (l'instant où la commande a donné l'ordre de démarrer une tâche et l'instant où le capteur indique la fin d'exécution de la tâche) permet de détecter au plutôt d'éventuelles défaillances. La deuxième phase correspond au diagnostic qui consiste en la détermination des causes du problème observé.

Jusqu'à présent les approches de surveillance utilisent un modèle de référence basé sur la connaissance à priori de toutes les situations interdites du système. Le modèle de surveillance proposé présente un réel avantage. En effet, la détection d'éventuelles défaillances ne nécessite pas une liste exhaustive de toutes les défaillances possibles du système. En fonction de la catégorie du système de production et de la nature des tâches à surveiller nous introduisons une tolérance sur la durée de la tâche à surveiller. Ainsi, la notion de fonctionnement en 'mode dégradé' est introduite. Au-delà de cette tolérance, le système sera effectivement dans un état de 'défaillance'. Différents exemples illustrent la démarche proposée. Ils permettent de montrer la puissance d'une telle approche. De plus, un algorithme permettant le calcul du seuil optimal de tolérance est proposé, ainsi que l'évaluation des performances du système de surveillance.

**Mots clés :** Systèmes à Événement Discrets (SED), Réseaux de Petri (RdP), Automate Temporisé, Surveillance, Détection, Diagnostic.

## Summary

The goal of the thesis is to develop a new approach of monitoring discrete events systems. A method based on the monitoring of the service time was developed using timed automata as a modelling tool.

From a controlled system, the first objective is the monitoring of the different activities of this system in order to improve its availability. It is approved by the minimization of the number of stops which penalize the production; by following in real time, the operating condition of all sensors of the system. Thus, for each task, the monitoring of spending time between two events (the moment when the command is ordered the task and the moment when the sensor indicates the end of task execution) allows to detect possible failures. The second step corresponds to the diagnosis which is the determination of the causes of the problem observed.

Until now, the approaches of monitoring use a reference model based a priori on the knowledge of all the prohibited situations of the system. The proposed model of monitoring has a real advantage. Indeed, the detection of possible failures does not require an exhaustive list of all the possible failures of the system. We introduce a tolerance; depending of the category of the production system and the nature of the tasks to be supervised. Thus the concept of functioning on 'degraded mode' is introduced. Beyond this tolerance, the system will be in 'failure mode'. Various examples illustrate our method, which show the power of such an approach.

Then an optimisation algorithm allowing to calculate the fault latency is proposed, as well as the performance evaluation of the monitoring system.

**Keys words:** Discrete events systems, Petri Nets, Timed Automata, Monitoring, Detection, Diagnosis.