50376
1986
48

N° d'ordre : 672

THÈSE

présentée à

L'UNIVERSITE DES SCIENCES ET TECHNIQUES DE LILLE FLANDRES ARTOIS

pour obtenir le titre de

DOCTEUR ES-SCIENCES PHYSIQUE PAR

Jean DEFRENNE

MODELISATION DE LA PARTIE OPERATIVE - IMPACT SUR LA SECURITE ET LA MAINTENANCE DES AUTOMATISMES A EVOLUTION SEQUENTIELLE

Soutenue le 25 Février 1986 devant la Commission d'Examen

MM.	P.	VIDAL	Président
	A.	COSTES	Rapporteur
	F.	PRUNET	Rapporteur
	J.M.	TOULOTTE	Rapporteur
	J.C.	GENTINA	Examinateur
	S.	THELLIEZ	Examinateur
	P.	ALANCHE	Invité

C H A P I T R E 2 : INFLUENCE DE LA SYNTHESE DES DISPOSITIFS DE COMMANDE DES AUTOMATISMES A EVOLUTION SEQUENTIELLE

C H A P I T R E 3 : INFLUENCE DES PANNES NON CONSISTANTES

I N T R O D U C T I O N

Tout système est sujet à des défaillances susceptibles d'entraîner à terme des interruptions de service aux conséquences préjudiciables. La fiabilité s'est montrée insuffisante pour appréhender l'ensemble des implications socio-économiques résultant des défaillances, notamment pour les systèmes réparables.
La sûreté de fonctionnement est une discipline qui vise à prévoir, de façon quantitative, le comportement d'un système, compte tenu de la probabilité de défaillance de chacun de ses composants.
Cette évaluation permet de comparer l'efficacité de différents moyens mis en oeuvre pour minimiser les effets néfastes de ces défaillances.
La disponibilité, la sécurité, la maintenabilité et la fiabilité sont les principaux estimateurs de la sûreté de fonctionnement des systèmes.

Dans le cadre de cette discipline, nous nous sommes interessés plus spécialement, aux performances des systèmes utilisés pour la production automatisée, dont la commande est confiée à des calculateurs industriels. Ces équipements sont caractérisés par l'importance et la diversité des éléments mécaniques mis en oeuvre et par le grand nombre d'entrées / sorties des calculateurs utilisés.

De nombreuses raisons justifient la définition d'un niveau minimum de sûreté de fonctionnement. Nous citerons comme exemple: la complexité croissante des automatismes entraînant une croissance des taux de pannes; l'augmentation des cadences rendant les coûts de production plus sensibles aux arrêts; les puissances mises en jeu, susceptibles de rendre ces systèmes dangereux...

Pour atteindre les objectifs de sûreté de fonctionnement définis par le cahier des charges, il est possible d'agir dès la conception du produit (choix de composants pour leur fiabilité intrinsèque, choix d'architectures...), comme dans la phase d'exploitation (définition de la politique de maintenance: organisation, moyens...).

Notre étude est une contribution à l'amélioration de la sûreté de fonctionnement et à la prévision des performances obtenues en ce domaine. Nos propositions s'appliquent à la phase de conception des systèmes de production automatisée sus visés. L'originalité de notre étude est liée au fait que nous avons plus particulièrement pris en considération les éléments du système autres que le calculateur. Pour celui-ci, il est effectivement possible d'adopter les solutions proposées pour les systèmes de traitement et de communication de l'information.

Dans la première partie de notre étude, nous présentons les outils utilisés pour évaluer la sûreté de fonctionnement. Nous y justifions notre choix de considérer la sécurité comme la caractéristique de première importance dans l'hypothèse des systèmes réparables. Nous montrons également comment différentes transcriptions d'un même cahier des charges peuvent agir sur la susceptibilité aux défaillances fugitives et intermittentes. Enfin, nous analysons la sûreté de fonctionnement que l'on peut attendre de différentes architectures matérielles couramment proposées. En dehors de l'unité de traitement qui peut être rendue autotestable par des techniques déjà traditionnelles, il apparaît que toutes les solutions présentées conduisent à dupliquer les éléments les moins fiables du système.
Notre objectif est alors de proposer un mécanisme de test en ligne des différents constituants de l'automatisme qui évite cette duplication.

Dans la deuxième partie, nous abordons l'étude théorique d'un tel mécanisme basé sur une méthode d'analyse syntaxique.
Nous évaluons qualitativement les performances que l'on peut attendre de ce test pour différentes grammaires nécessitant un niveau de connaissance croissant du système.

Dans la dernière partie, nous proposons une mise en oeuvre du dispositif de test en ligne. Cette étude porte sur le choix d'une architecture matérielle et sur la création de la grammaire. Enfin, nous définissons des estimateurs qui permettent de prévoir le taux de couverture obtenu.

PREMIERE PARTIE

SURETE DE FONCTIONNEMENT

INHERENTE A LA PARTIE COMMANDE

INTRODUCTION DE LA PREMIERE PARTIE

Dans cette première partie, nous définissons le cadre dans lequel s'inscrit notre travail. Nous rappelons, dans le premier chapitre, ce qu'est la sûreté de fonctionnement, ainsi que les moyens d'évaluation dont nous disposons. Ceci nous amène à analyser les contraintes qu'impose la sûreté de fonctionnement dans le cadre des applications envisagées.

Les deux chapitres suivants introduisent la notion de susceptibilité aux pannes fugitives et étudient l'influence de la conception du logiciel sur cette caractéristique.

Le quatrième chapitre analyse et quantifie la sûreté de fonctionnement obtenue pour diverses architectures matérielles couramment proposées.

CHAPITRE I

EVALUATION DE LA SURETE DE FONCTIONNEMENT D'UN AUTOMATISME

L'automatisation des activités industrielles conduit à la mise en place de systèmes de plus en plus complexes. Aux impératifs purement fonctionnels, s'ajoutent des considérations socio-économiques relevant de la sûreté de fonctionnement.
Le cahier des charges doit définir des spécifications dans chacun de ces domaines.
Nous introduisons, dans ce chapitre, deux notions d'états distinctes. L'une est relative à l'aspect fonctionnel, l'autre à la sûreté de fonctionnement du système.

Après avoir adopté une définition des automatismes qui sert de cadre à cette étude, nous présentons les moyens d'évaluation de la sûreté de fonctionnement habituellement mis en oeuvre.

1.1 SURETE DE FONCTIONNEMENT D'UN AUTOMATISME

1.1.1 NOTION D'AUTOMATISME - ASPECT FONCTIONNEL

Tout système est créé et mis en place en fonction d'une mission qui vise à satisfaire un besoin. Dans le cadre de la production automatisée, une mission correspond à un ensemble de transformations portant sur un ou plusieurs produits.

Parmi les moyens mis en oeuvre pour satisfaire une mission de production automatisée, il existe deux classes de sous-systèmes.

Tout sous-ensemble qui opère une transformation du produit a une fonction de partie opérative. C'est une sous-partie opérative.

Tout sous-ensemble qui participe à la gestion de l'ordonnancement des actions, ou tâches, réalisées par les sous-parties opératives assure une fonction de commande. C'est une sous-partie commande.

Ces sous ensembles sont regroupés dans ce que nous appelons la Partie Opérative d'une part, la Partie Commande d'autre part. Ces deux classes seront notées ultérieurement P.O. et P.C..

Ces deux sous-ensembles communiquent par l'intermédiaire des ordres émis de la P.C. vers la P.O. et des comptes rendus, désignés ultérieurement par C.R., élaborés par les capteurs et mis en forme par les transducteurs.

Nous considérons l'automatisme placé dans un environnement avec lequel il coopére. Cette coopération se traduit par des échanges d'informations dites de conduite, des transferts de pièces brutes ou usinées, d'outils ...

La notion d'automatisme adoptée est récursive dans la mesure où un un automatisme, pris comme un sous système monolithique, peut être considéré comme une sous P.O. d'un automatisme qui l'englobe.

Exemple:
Une machine automatisée contenant une boucle de régulation est un automatisme. La boucle de régulation avec son régulateur, ses capteurs et ses éléments de réglage forme également un automatisme englobé dans le précédent.
Un atelier contenant des machines automatisées peut être considéré comme un automatisme.

Un opérateur humain ayant des fonctions de P.O. ou de P.C. est éventuellement considéré comme faisant partie de l'automatisme. Son influence sur la sûreté de fonctionnement du système étudié est à ce titre tout à fait significative.

La transformation apportée au produit par un outil de production nécessite l'exécution d'actions effectuées par les sous P.O. dans un ordre préétabli. Dans ce sens, les automatismes que nous considérons relèvent de la classe des systèmes à évolution séquentielle. Les actions simultanées sont évidemment possibles.

Un automatisme se présente alors comme un ensemble multitâches pour lequel les tâches s'identifient aux actions en cours. Le rôle de la partie commande est de gérer cet ensemble de tâches. Nous restreignons notre étude aux automatismes pour lesquels la P.C. peut être modélisée par une machine séquentielle dont la définition est rappelée ultérieurement. Dans ces conditions, il existe à chaque instant un état interne et un état de sortie de la P.C. . L'état total ainsi défini est appelé état de l'automatisme dans la suite de l'exposé.

Nous appelons événement normal ou attendu toute modification d'un compte rendu (événement interne) ou d'une information de conduite (événement externe) prévue par le cahier des charges et susceptible de modifier l'état de l'automatisme.
Ces événements sont supposés de type booléen; ils sont indifféremment fournis par des dispositifs tout ou rien, ou par des grandeurs numériques (voire analogiques) prisent en compte à l'intérieur de prédicats.

De même, les ordres (internes) et les informations de sortie (externes) peuvent être de tout type.

Dans la pratique, il est possible qu'apparaissent des événements anormaux liés à des défaillances matérielles, logicielles ou humaines. L'aptitude de l'automatisme à remplir sa mission, compte tenu de ces occurences, relève du domaine de la sûreté de fonctionnement. Nous donnons ci-après les définitions des termes utilisés dans la suite de l'étude.

1.1.2 COMPORTEMENT DES AUTOMATISMES EN PRESENCE DE DEFAILLANCES: TERMINOLOGIE

1.1.2.1 Fautes et erreurs

Dans la phase d'exploitation de l'automatisme, l'occurence d'événements anormaux liés à des défaillances matérielles, logicielles ou humaines est probable. Ces événements sont appelés fautes |ALA-83|.

La sollicitation d'un sous-ensemble de l'automatisme contenant une une panne conduit à la génération d'une erreur.
La faute altère donc les aptitudes de ce sous-ensemble à remplir sa mission. L'erreur est le résultat anormal fourni par le sous-système défaillant. A ce titre, c'est une information.

Exemple:
- la rupture d'un foret (faute) entraîne la production de pièces mauvaises (erreur);
- le collage d'un fin de courses (faute) entraîne une erreur sur la localisation de l'actionneur.

Tout traitement d'une erreur, même par un sous-ensemble sain, peut provoquer une nouvelle erreur. C'est le phénomène de propagation des erreurs.

Exemple:
Après collage du fin de course, la P.C. prenant en compte une position erronée de l'actionneur génère des ordres erronés.

Le traitement d'une erreur peut aussi provoquer une panne. Ceci est particulièrement vrai pour les automatismes compte tenu des puissances mises en jeu.

Exemple:
Le taraudage de la pièce non percée peut entraîner une rupture du taraud.

1.1.2.2 Réparation - Correction

Une réparation est un événement normal qui élimine au moins une faute.

Une correction supprime au moins une erreur.

Exemples:
- changer le foret cassé sur un poste de travail est considéré comme une réparation;
- effectuer une retouche des pièces défectueuses est une correction;
- un code correcteur d'erreur élimine l'erreur mais pas la panne. Il effectue une correction et non une réparation.

1.1.2.3 Comportement vis-à-vis des fautes

- Une faute est acceptée par le système si elle ne fait décroître la probabilité de réalisation correcte d'aucune tâche.
- Une faute est tolérée si la probabilité de réalisation de toutes les tâches dans un temps acceptable est non nulle.

  Exemple:
  Une automobile munie de sa roue de secours tolère une crevaison.

- Une faute est latente à un instant t si l'utilisateur ignore, à cet instant, l'existence d'une faute. Une faute non latente est dite révélée. Une faute est révélée en général, par un mécanisme de détection qui constate l'erreur induite par cette faute.

1.1.2.4 Structure de l'automatisme

Les automatismes considérés sont formés de sous-systèmes matériels et éventuellement logiciels. L'organisation de ces sous-ensembles est définie par l'architecture du système. Cette architecture est choisie par le concepteur de façon à satisfaire le cahier des charges sous son aspect fonctionnel, mais aussi de façon à assurer la sûreté de fonctionnement attendue.

Dans le cadre de la sûreté de fonctionnement, l'ensemble des moyens matériels et logiciels mis en place est communément appelé "structure" du système.
Nous définissons alors l'état de la structure par la présence ou l'absence de panne dans les sous-systèmes ainsi rassemblés. En d'autres termes, nous dirons que l'apparition d'une panne (ou d'une faute), à la suite d'une défaillance, ou sa suppression, à la suite d'une réparation, modifie l'état de la structure.

Il ne saurait y avoir de confusion entre ces deux notions d'état relatives au système étudié. L'état de l'automatisme, attaché à l'aspect fonctionnel du cahier des charges, évolue en fonction d'événements dits normaux ou attendu, alors que l'état de la structure est lié à l'existence et à la localisation des défaillances éventuelles.

Il est clair que toute altération de la structure peut avoir une répercussion sur l'état de l'automatisme par l'intermédiaire de l'erreur engendrée.

1.1.3 PARAMETRES D'EVALUATION DE LA SURETE DE FONCTIONNEMENT

D'une manière très générale, la sûreté de fonctionnement d'un système peut être définie comme son aptitude à minimiser le nombre et l'effet des fautes.

Les constituants principaux de la sûreté de fonctionnement des systèmes sont:
la fiabilité, la disponibilité, la maintenabilité, la sécurité.
Ces différentes grandeurs peuvent se définir en termes de probabilité. |COS.76|

- La fiabilité (reliability) R(t) est la probabilité de fonctionnement à tout instant. $\tau \in [0,t]$

- La disponibilité (availability) A(t) est la probabilité de fonctionnement à l'instant t.

- La sécurité (safety) S(t) est la probabilité de fonctionnement de sécurité à tout instant $\tau \in [0,t]$.
  La notion de fonctionnement de sécurité doit être précisée.

- La maintenabilité M(t) est la probabilité pour que le système fonctionne à l'instant t sachant qu'il était en panne à tout instant $\tau \in [0,t]$.

D'autres propriétés sont parfois évoquées et notamment la crédibilité et la survivabilité qui se définissent comme suit:

- La crédibilité V(t) est la probabilité pour que le temps moyen de latence d'une erreur soit inférieur à t. Les erreurs de conception ne sont généralement pas prises en compte dans le calcul.

- La survivabilité SU(t) est la probabilité de fonctionnement à l'instant t, sachant qu'une défaillance s'est produite entre les instants 0 et t.

Les calculs de probabilité étant assez fastidieux, les industriels utilisent souvent d'autres estimateurs dont les plus courants sont:

- la disponibilité stationnaire (stationary availability) $SA = \lim_{t \to \infty} D(t)$

- le temps moyen jusqu'à la première défaillance (mean time to first failure) $MTFF = \int_0^\infty R(t)dt$

- le temps moyen jusqu'à la première défaillance maligne (mean time to first maliguant failure) $MTFMF = \int_0^\infty S(t)dt$

- le temps moyen jusqu'à la réparation (mean time to repair) $MTTR = \int_0^\infty M(t)dt$

- le temps moyen entre défaillances $MTBF = MTFF + MTTR$

  en fait MTBF =MUT + MDT qui correspondent aux valeurs respectives de MTFF et MTTR atteintes en régime stationnaire.

Toutefois, les temps moyens (MTBF, MTFF...) sont mal adaptés aux comparaisons entre architectures. |LAP.75|

1.4 AMELIORATION DE LA SURETE DE FONCTIONNEMENT

En fait, il n'y a pas indépendance entre les différents paramètres de la sûreté. L'amélioration de la sécurité, par exemple, entraîne souvent une augmentation du nombre des composants. Ceci peut avoir un effet néfaste sur la fiabilité.

Pour réaliser un automatisme sûr de fonctionnement, le concepteur dispose de deux moyens d'action. |AVI.75|

a) Il peut diminuer la probabilité d'apparition des fautes et erreurs; dans ce cas, on parle d'intolérance aux fautes.
b) Il peut limiter les effets des perturbations, c'est-à-dire accroître la probabilité qu'une erreur soit acceptée ou tolérée; on parle alors de tolérance aux fautes.

1.1.4.1 Intolérance aux fautes

Cette approche nécessite la suppression des erreurs de conception et une adaptation du système à son environnement. Ceci implique:

- le choix d'éléments matériels très fiables,
- une étude poussée des conditions de fonctionnement,
- une conception déductive validée à chaque niveau,
- une politique efficace de maintenance préventive,
- un déverminage destiné à éliminer les défauts de jeunesse.

1.1.4.2 Tolérance aux fautes

La tolérance aux fautes nécessite:

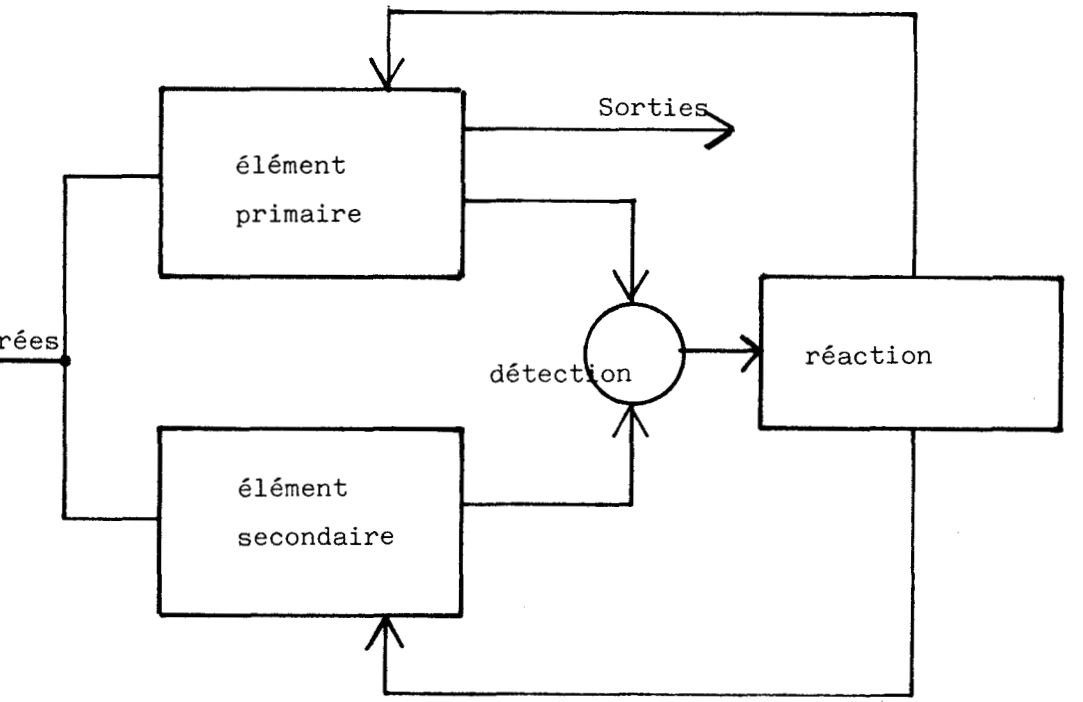
- la détection et la localisation des erreurs,
- un mécanisme de réaction:
  - . confinement des erreurs (limitation de leur propagation)
  - . correction des erreurs
  - . reconfiguration du système...

Cette approche est basée sur l'utilisation de redondances.
Un dispositif redondant est un élément du système qui lui permet d'accepter ou de tolérer certaines erreurs.

Un système redondant comprend (fig. 1.2):

- . un élément primaire
- . un élément redondant
- . un élément de détection des erreurs
- . un élément de traitement des erreurs (correction, confinement, reconfiguration).

Le programme SURF |COS.80| permet de faire des études comparatives d'architectures sur le plan de la sûreté de fonctionnement, en fonction de la fiabilité propre des éléments du système.



- figure 1.2 -

1.1.4.3 Classification des redondances

Deux types de classification sont proposés en fonction:
- de l'état de l'élément redondant avant erreur ;
- du comportement par rapport aux tâches.

a) Redondance passive et active
- Si l'élément redondant n'est actif qu'après détection d'une erreur, la redondance est dite passive.
- Si l'élément redondant est en service avant détection d'une erreur, la redondance est active. Ceci est vrai même si cet élément ne participe pas au traitement des tâches avant erreur.

b) Redondance dynamique ou statique

La redondance est dite dynamique si l'élément redondant ne participe à la réalisation des tâches qu'après détection et réaction à une erreur.

Elle est de plus:

- sélective, si l'élément redondant remplace l'élément primaire,
- temporelle, si l'élément redondant se contente de mémoriser les informations avant erreur, et de les restituer après détection et réaction.

La redondance est statique si l'élément redondant prend une part active à la réalisation des tâches avant même la détection d'une erreur. Le cas le plus connu est la redondance massive formée d'un nombre impair n de systèmes traitants les mêmes informations. Les sorties sont alors élues par un voteur parmi les n sorties.

Par principe, toute redondance statique est active.

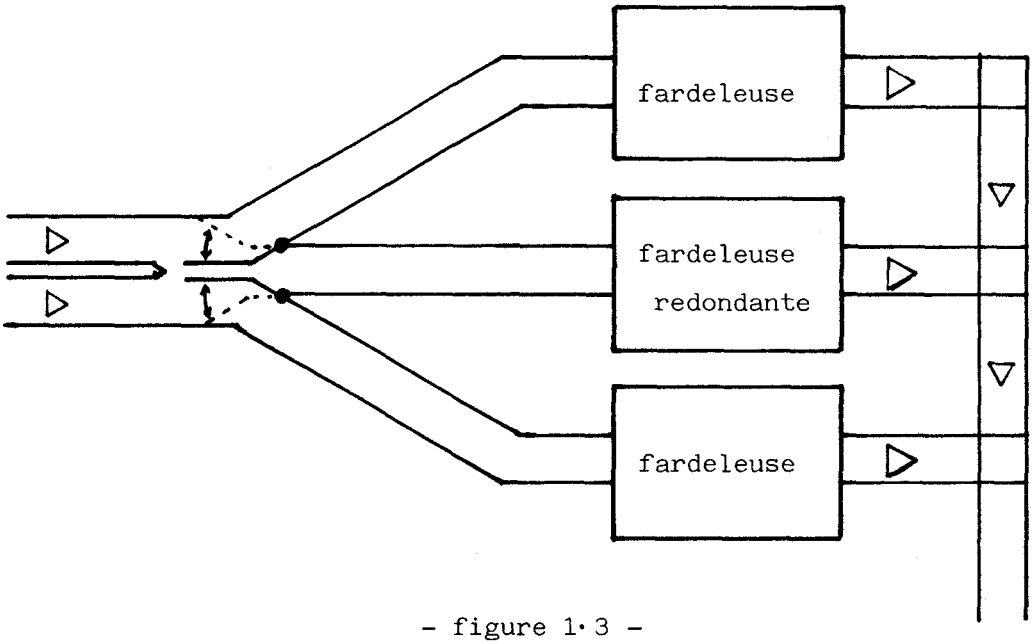
1.1.4.4 Introduction de redondances dans les automatismes

Selon le système considéré, l'introduction de redondance relève de disciplines différentes.

La redondance peut s'envisager au niveau de l'atelier.

Exemple SEITA Lille:

En bout d'une ligne double de fabrication de cartouches de cigarettes, la société a placé trois fardeleuses qui forment des paquets de 24 cartouches (figure 1.3).



- figure 1·3 -

Ceci correspond à une redondance dynamique sélective passive.
La redondance est optimalisée par l'adoption d'un système redondant pour deux systèmes primaires. Chaque emballeuse est un automatisme formé d'une partie opérative et d'une partie commande. La redondance est bien au niveau de la chaîne. Dans une telle installation, la détection et la réaction aux erreurs sont confiées à un opérateur.

Un autre exemple fréquent sur les lignes de production est l'adoption de stocks tampons entre les machines (ou groupe de machines).
Il s'agit alors de redondances temporelles introduites pour satisfaire aux exigences de la gestion de production.

1.2 SECURITE DES AUTOMATISMES

Le fonctionnement de sécurité auquel il est fait référence dans la définition de l'estimateur S(t) vu précédemment doit être précisé.

Celle retenue par la "Société pour l'Avancement de la Sécurité des Systèmes en France"*est une base interessante. Elle stipule que:

" La sécurité d'un système dans l'accomplissement de sa mission serait réalisée de façon absolue par l'impossibilité de tout événement ou concours de circonstances entraînant blessure ou morts de personnes, ou dégats matériels catastrophiques".

La définition que nous avons adoptée dans le cadre de la sûreté est:

" La sécurité S(t) est la probabilité pour que le système soit dans un fonctionnement de sécurité dans l'intervalle $]0,t]$".

Le caractère probabiliste de cette sécurité est à rapprocher de l'expression: "serait réalisée de façon absolue". Il est clair que la sécurité absolue n'existe pas.
Le "fonctionnement en sécurité" correspond alors à un fonctionnement tel que "blessure ou morts de personnes, ou dégats matériels catastrophiques" soient impossibles. Cette définition reste dans bien des cas assez subjective.

* 3SF 98, rue de La Bruyère 78300 Poissy

La sécurité relative aux machines et appareils est régie par décrets. Voyons ce qu'apportent ces textes législatifs par rapport aux définitions ci-dessus.

1.2.1 LA SECURITE DANS LE CODE DU TRAVAIL

Les règles générales codifiées relatives à la sécurité, applicables à l'ensemble des machines forment les articles R 233-84 à R 233-107 du Code du Travail. Elles sont complétées pour certaines machines dites dangereuses, par le décret n° 80544 du 15 juillet 1980.

Par ces décrets, le législateur vise surtout à assurer la protection des personnes. On y trouve notamment la sécurité intrinsèque, les problèmes d'ergonomie et de convivialité avec la machine.

La protection vis-à-vis des parties mobiles mues par une force autre que celle de l'opérateur y est particulièrement prise en compte. Elle impose les dispositifs de protection (carters fixes et mobiles) et la notion de protection des zones dangereuses (barrière immatérielle, plancher sensible...).

Au niveau de la commande, nous retenons surtout les articles R 233-95 à R 233-99 qui fixent les règles générales dont l'application a pour objectif d'éviter les situations dangereuses qui peuvent résulter notamment:

- de la mise en mouvement intempestive ou de l'emballement d'éléments mobiles de machines,
- du desserrage accidentel de dispositifs de maintien, de bridage,
- de collisions d'éléments mobiles entre eux ou avec des éléments fixes,
- de l'impossibilité d'arrêter un mouvement,
- de la neutralisation accidentelle de dispositifs de protection.

Par leur application, tous ces articles visent essentiellement à une réduction des risques en l'absence de défaillance. Par construction, la machine doit être la moins dangereuse possible en cours d'exploitation. Ceci doit être vrai également lors des marches de réglages ou en période d'entretien.

L'article R 233-97 se préoccupe de façon précise du comportement du système en présence d'une défaillance.

> Article R 233-97: "Une défaillance, une panne ou une détérioration du système d'alimentation en énergie ou du circuit de commande des machines et appareils ne doit:
> - ni provoquer la mise en marche intempestive d'un élément mobile de la machine,
> - ni empêcher l'arrêt automatique ou manuel des éléments mobiles quels qu'ils soient,
> - ni rendre inefficaces les dispositifs de protection des éléments mobiles".

Ces articles extraits du Code du Travail interessent la protection des utilisateurs. Ils ne couvrent évidemment pas l'ensemble des aspects législatifs de la sécurité de tous les types d'installations. Ils conduisent le concepteur à introduire sur la machine un certain nombre de capteurs, notamment pour la vérification de mise en place des dispositifs de protection et la surveillance de zones.
Ils influent également sur le choix de ces capteurs et sur la façon d'exploiter l'information fournie.
Par exemple, pour vérifier la permanence du bridage d'une pièce pendant une phase d'usinage, il sera souhaitable de placer un pressostat sur l'alimentation des vérins de serrage. Au niveau commande, toute disparition du serrage doit provoquer l'arrêt de l'usinage.

Notre préoccupation est en fait d'appréhender l'évolution de la sécurité face à une défaillance de l'automatisme.

1.2.2 COMPORTEMENT DE L'AUTOMATISME CONTENANT UNE PANNE

Tous les éléments d'un automatisme sont sujets à défaillance.
Le choix de l'architecture est guidé par les probabilités de défaillance de chaque élément et la gravité de l'accident qui peut en résulter. Pour chaque élément, les différents modes de défaillance doivent être envisagés.

Nous définissons deux types d'action visant à limiter les risques d'événements catastrophiques. Nous distinguons:
- la sécurité statique,
- la sécurité dynamique.

La sécurité statique consiste à choisir les éléments de la structure et leur câblage de façon à ce que l'apparition d'une des pannes les plus probables amène naturellement l'automatisme dans un fonctionnement de sécurité. Un tel choix relève des règles de l'art.
Dans cette catégorie, nous rangeons le câblage dit "en sécurité" qui permet d'éliminer la défaillance la plus dangereuse sur un dispositif d'arrêt, compte tenu du risque de rupture des liaisons électriques dans une installation. Cette approche de la sécurité permet une amélioration sans apport de composants supplémentaires. Elle permet de diminuer dans de nombreux cas, les risques vis-à-vis des défaillances des dispositifs de protection, d'arrêt ou d'alimentation.

La sécurité dynamique est celle que l'on obtient par la mise en place de mécanismes de détection et de réaction. Une modification de structure matérielle et/ou logicielle est alors obligatoire.

La sécurité dynamique n'entraîne pas forcément l'existence de redondances, puisque la détection de la panne n'entraîne pas systématiquement sa tolérance.
Par contre, la sécurité obtenue avec un système tolérant relève de la sécurité dynamique vis-à-vis des fautes qu'il tolère.

Pour un automatisme, il est possible de définir un état de sécurité global ou localisé à un ou plusieurs actionneurs. Nous classons les automatismes en fonction des moyens permettant de garantir la sécurité en cas de défaillance.
Dans les cas les plus simples, cet état est directement accessible. Il suffit de choisir les préactionneurs et actionneurs afin que la probabilité d'atteinte de cet état soit la plus grande possible. Classiquement, la sécurité statique relève de cet aspect.

Dans d'autres cas, cet état ne peut être atteint que par déroulement d'une séquence particulière.

Nous sommes alors obligés de prévoir une continuité de service totale ou partielle de l'automatisme, même en présence de pannes. Ceci constitue une marche dégradée. Dans les cas extrêmes, la poursuite intégrale de la mission est impérative. La mise en oeuvre de redondances est alors indispensable.
Ceci constitue une contrainte qui intervient au moment du choix d'une architecture.

Remarque:

Les automatismes pour lesquels la continuité de mission, partielle ou totale, est impérative jusqu'à réparation, s'apparentent aux systèmes non continûment réparables.

Il reste à envisager le comportement dynamique d'un automatisme contenant une défaillance non tolérée.
L'interaction partie commande, partie opérative est telle que l'erreur engendrée par la panne se propage jusqu'à la commande. Par définition, l'événement anormal que constitue l'apparition de cette erreur, n'a pas été pris en compte lors de la conception de la commande. Il y a alors génération de ce qu'il est convenu d'appeler une séquence aberrante pouvant rendre le système dangereux malgré les précautions prises par ailleurs.

Comme dans |LAP 75| et |CLA 84| nous adopterons la convention suivante:

Tout système qui contient une défaillance non acceptée ou tolérée est dans un fonctionnement de non sécurité.

C'est à partir de cette convention que nous calculerons la probabilité S(t).

1.3 EVALUATION DE LA SURETE DES SYSTEMES REPARABLES

Pour une première approche, on se référera à |SCH 69|.Pour approfondir l'étude, on se reportera à |COR 74| et |LAP 75|.

Le système est présenté ici sous la forme d'une machine d'états stochastique ; c'est-à-dire qu'à chaque état de la structure est associée la probabilité d'être dans cet état à l'instant t.
La somme des probabilités est évidemment égale à 1 à chaque instant.
A chaque arc Ei, Ej est associé un taux de transition λij.
Dans le cadre de la sûreté de fonctionnement, les λij correspondent à des taux de défaillances ou à des taux de réparations.

Si nous étudions l'influence des fautes accidentelles en dehors des périodes de jeunesse et de vieillissement, nous pouvons adopter l'hypothèse couramment admise de taux de défaillances constants. La même hypothèse est généralement admise pour les taux de réparations car on suppose que ces temps sont petits par rapport à la période d'exploitation.
Dans ces conditions, le système est assimilé à un processus stochastique markovien homogène.

1.3.1 PROBABILITE D'ETRE DANS UN ETAT

Soit Pj(t) la probabilité que la structure se trouve dans l'état Ej à l'instant t. Soit n le nombre d'états.
Le théorème des probabilités totales conduit à:

$$Pj\ (t + dt) = \sum_{i=1}^{n} Pij\ (t,\ t + dt)\ .\ Pi\ (t)$$

Pij est la probabilité de transition de l'état i vers l'état j dans l'intervalle t, t + dt.
Compte tenu des taux de transitions λij constants, cette relation devient:

$$\frac{d}{d\ t}\ (Pj) = \sum_{i=1}^{n} \lambda ij\ .\ Pi\ (t) \quad \text{lorsque dt tend vers 0.}$$

En étendant cette relation à l'ensemble du système, on obtient l'équation de Chapmann - Komolgorov:

$$\dot{\mathbb{P}}(t) = \mathbb{P}(t) \cdot \Lambda \qquad |1\text{-}1|$$

dont la solution est de la forme:

$$\mathbb{P}[\, t / \mathbb{P}(0)\,] = \mathbb{P}(0) \cdot \exp(\Lambda . t)$$

Pratiquement on fait appel à la transformée de Laplace de l'équation de Chapmann - Komolgorov |1-1|

$$p\,\tilde{\mathbb{P}}(p) - \mathbb{P}(0) = \tilde{\mathbb{P}}(p) . \Lambda \qquad |1\text{-}2|$$

où $\mathbb{P}(0)$ représente les probabilités initiales (t = 0)

$$(\mathbb{I} . p - \Lambda) \cdot \tilde{\mathbb{P}}(p) = \mathbb{P}(0) \qquad |1\text{-}3|$$

ce qui donne la solution suivante en repassant à l'originale:

$$\mathbb{P}(t / \mathbb{P}(0)) = \mathbb{P}(0)\, \mathcal{L}^{-1}\left[(p\mathbb{I} - \Lambda)^{-1}\right] \qquad |1\text{-}4|$$

Il est également possible, à partir de |1-2| de tracer un graphe de transfert. Le noeud source est alors celui qui correspond à l'état initial j tel que Pj(0) = 1, Pi(0) = 0, $\forall\, i \neq j$.

La probabilité d'être dans un état i à l'instant t se détermine alors par application de la règle de Mason.

1.3.2 EVALUATION DE LA SURETE D'UN AUTOMATISME PARTICULIER

Pour illustrer la méthode de calcul, nous évaluons la sûreté d'un automatisme réparable muni d'un dispositif de détection d'erreurs. Cet automatisme a un taux de pannes λ supposé constant. Le système de détection est supposé parfait sur le plan de la sûreté ($\lambda_d = 0$).

Supposons qu'un mécanisme de réaction mette l'automatisme en fonctionnement de sécurité dès qu'une anomalie est signalée.

Nous noterons Pc le taux de couverture du système de détection défini par:

Pc = $\mathcal{P}$ (que la panne soit signalée sachant qu'une panne existe)

Supposons que cet automatisme soit réparable. Nous notons μ et ν les taux de réparation après une panne révélée (μ) ou non révélée (ν).
D'après la convention adoptée, la présence d'une panne non révélée place le système dans un état dangereux. Il est supposé également que le système de détection et de réaction est infaillible.

Nous adoptons la classification des pannes suivante: une panne est dite bénigne si elle ne compromet pas la sécurité. Son existence peut toutefois entraîner un arrêt de la mission.
Une panne est dite maligne si elle ne permet pas d'assurer la sécurité. Dans notre exemple, une défaillance détectée correspond à une panne bénigne.

1. 3.2.1 Calcul des estimateurs de la sûreté

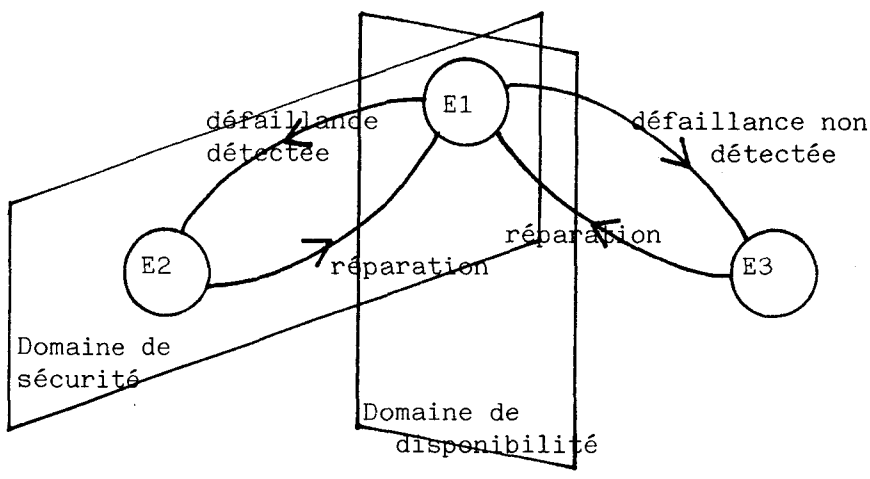
Nous pouvons définir trois états pour la structure:

E1 = automatisme sain

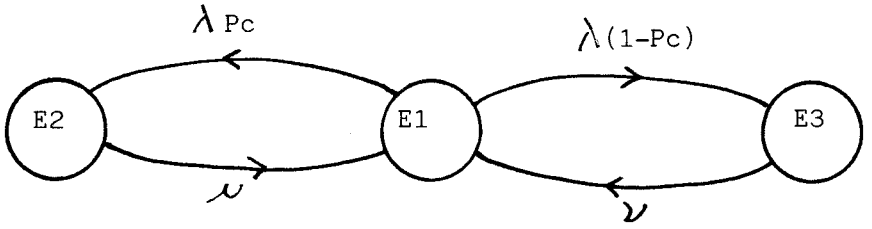
E2 = automatisme mis en sécurité suite à une erreur détectée

E3 = automatisme fonctionnant avec une erreur non détectée. Il est considéré comme dangereux.

Nous obtenons le graphe suivant (fig 1.4a et 1.4b):



- figure 1-4a -



- figure 1-4b -

Nous définissons une fonction $z(t)$ telle que $z(t) = i$ si la structure est dans l'état Ei à l'instant t.

La disponibilité s'écrit alors:

$$A(t) = \mathcal{P}\{ z(t) = 1 ; \forall z(\tau), \tau \in [0, t[\}$$

La sécurité s'écrit:

$$S(t) = \mathcal{P}\{ z(t) = 1,2 ; \forall z(\tau), \tau \in [0, t[\}$$

Du graphe de la fig. 1.4b, nous tirons les équations suivantes:

$$\dot{P}_1 = -\lambda P_1 + \mu P_2 + \nu P_3$$

$$\dot{P}_2 = \lambda P_c P_1 - \mu P_2$$

$$\dot{P}_3 = \lambda (1 - P_c) P_1 - \nu P_3$$

Prenons la transformée de Laplace du système d'équations:

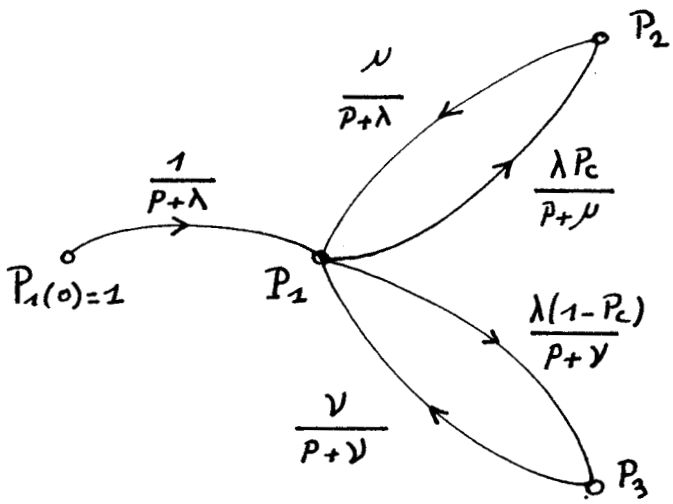
$$(p + \lambda)\, \tilde{P}1 = \mu\, \tilde{P}2 + \nu \tilde{P}3 + P1(0)$$

$$(p + \mu)\, \tilde{P}2 = \lambda P_c \tilde{P}1 + P2(0)$$

$$(p + \nu)\, \tilde{P}3 = \lambda\, (1 - P_c)\, \tilde{P}1 + P3(0)$$

Supposons que la structure est neuve ou réparée à l'instant initial. Nous avons donc P1(0) = 1, P2(0) = 0, P3(0) = 0.

Nous en tirons le graphe de fluence (Fig 1.5):



- figure 1-5 -

Par la règle de Mason, nous obtenons:

$$A(p) = \frac{\frac{1}{p+\lambda}}{1 - \frac{\mu\lambda\, Pc}{(p+\lambda)(p+\mu)} - \frac{\nu\lambda\,(1-Pc)}{(p+\lambda)(p+\nu)}} \qquad |1\text{-}5|$$

si, $\mu = \nu$, on obtient:

$$A(p) = \frac{p+\mu}{(p+\lambda)(p+\mu) - \mu\lambda}$$

$$\boxed{A(t) = \frac{\mu}{\lambda+\mu}\left[1 + \frac{\lambda}{\mu} e^{-(\lambda+\mu)t}\right]} \qquad |1\text{-}6|$$

La disponibilité asymptotique est:

$$SA = \lim_{t\to\infty} A(t) = \lim_{p\to 0} p\, A(p) = \frac{\mu}{\lambda+\mu} \qquad |1\text{-}7|$$

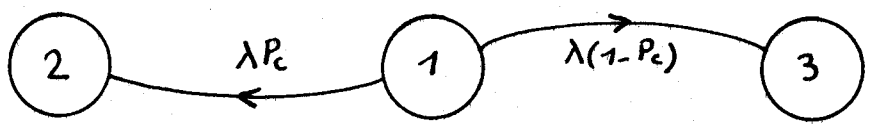
Si $\mu \neq \nu$ la relation |1-5| conduit à:

$$A(p) = \frac{p^2 + (\nu + \mu)p + \nu\mu}{p\left[p^2 + (\lambda + \mu + \nu)p + \mu\nu + \nu\lambda P_c + \mu\lambda(1-P_c)\right]} \qquad |1\text{-}8|$$

qui donne la disponibilité asymptotique

$$SA = \frac{\mu\nu}{\mu\nu + \nu\lambda P_c + \mu\lambda(1-P_c)} \qquad |1\text{-}9|$$

La fiabilité est , ici, la probabilité pour que l'automatisme soit sain dans l'intervalle | 0,t |. Nous rendons donc absorbant les états E2 et E3 puisque la mission n'est plus remplie pendant les temps de réparation.
Le graphe se réduit alors à (Fig 1.6):



- figure 1-6 -

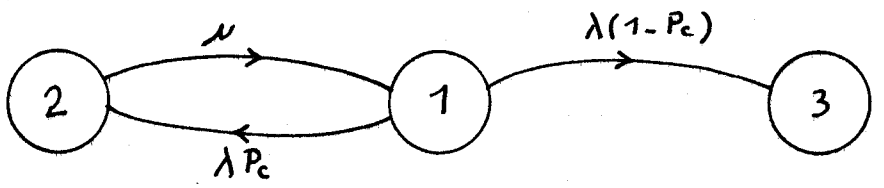
et $R(t) = \mathcal{P}[z(\tau) = 1, \quad \forall \tau \in [0,t]]$

$P1 = \lambda P1$

$(p + \lambda) P1 = P1(0)$

$$R(t) = e^{-\lambda t} \qquad |1\text{-}10|$$

Pour la sécurité, l'état E2 est acceptable. E3 est donc le seul état absorbant. Le graphe est alors (Fig 1.7):

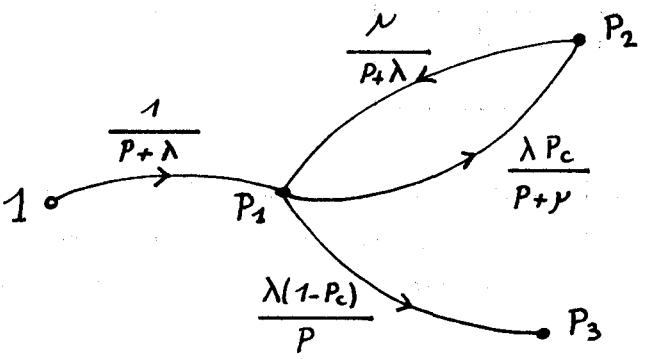


- figure 1-7 -

et $S(t) = \mathcal{P}\{z(\tau) \neq 3, \forall \tau \in [0,t]\}$

$$\left|\begin{array}{l} \dot{P}1 = -\lambda P1 + \mu P2 \\ \dot{P}2 = \lambda Pc\, P1 - \mu P2 \\ \dot{P}3 = \lambda (1-Pc)\, P1 \end{array}\right| \Longrightarrow \left|\begin{array}{l} (p+\lambda)\, P1 = \mu P2 + 1 \\ (p+\mu)\, P2 = \lambda Pc\, P1 \\ p\ \ P3 = \lambda (1-Pc) P1 \end{array}\right|$$

dont le graphe de transfert est (Fig 1.8):



- figure 1-8 -

$$S(t) = 1 - P3(t)$$

$$P3(p) = \frac{\lambda (1 - Pc)(p + \mu)}{p\left[(p+\lambda)(p+\mu) - \mu\lambda Pc\right]}$$

$$P3(p) = \frac{1}{p} - \frac{p + \lambda Pc + \mu}{(p-s1)(p-S2)}$$

$$\text{avec } S1,\ S2 = \frac{-(\lambda + \mu) \pm \sqrt{(\lambda + \mu)^2 - 4\lambda\mu|1-Pc|}}{2}$$

$$\text{ou encore } S1, S2 = \frac{-(\lambda + \mu) \pm \sqrt{(\mu - \lambda)^2 + 4Pc\mu\lambda}}{2}$$

$$P3(t) = 1 - \frac{S1 + \mu + \lambda Pc}{S1-S2} e^{S1t} - \frac{S2 + \mu + \lambda Pc}{S2-S1} e^{S2t}$$

$$S(t) = \frac{S1 + \mu + \lambda Pc}{S1-S2} e^{S1t} + \frac{S2 + \mu + \lambda Pc}{S2-S1} e^{S2t}$$

3.2.2 Influence du mécanisme de détection de défaut sur la sûreté

L'évaluation faite est très approximative dans la mesure où nous supposons que la probabilité de panne du système de détection est nulle. Une étude plus complète est faite au chapitre 4. Néanmoins, cet exemple permet de dégager des tendances. Il va nous permettre également de justifier le cadre dans lequel nous plaçons le reste de l'étude.

Nous allons comparer deux architectures extrêmes. Pour la première, nous supprimons le système de détection; ce qui correspond à Pc = 0. Pour la deuxième, nous supposons Pc = 1.
Le tableau de la figure 1-9 regroupe les résultats obtenus.

	Pc = 0	Pc = 1
Fiabilité	$R(t) = e^{-\lambda t}$ $MTFF = \frac{1}{\lambda}$	$R(t) = e^{-\lambda t}$ $MTFF = \frac{1}{\lambda}$
Sécurité	$S(t) = e^{-\lambda t}$ $MTFMF = \frac{1}{\lambda}$	$S(t) = 1$ $MTFMF \longrightarrow \infty$
Disponibilité	$A(t) = \frac{\nu}{\lambda+\nu}\left[1+\frac{\lambda}{\nu}e^{-(\lambda+\nu)t}\right]$ $SA = \frac{\nu}{\lambda+\nu}$	$A(t) = \frac{\mu}{\lambda+\mu}\left[1+\frac{\lambda}{\mu}e^{-(\lambda+\mu)t}\right]$ $SA = \frac{\mu}{\lambda+\mu}$

- figure 1-9 -

Les résultats, très idéalisés sur la fiabilité et la sécurité, mettent en évidence les limites de notre hypothèse selon laquelle le taux de panne du dispositif détecteur de panne est nul.
Nous pouvons admettre néanmoins, une amélioration de la sécurité.

Dans l'hypothèse des taux de réparation identiques après défaillance, qu'elle soit ou non détectée ($\mu = \nu$), la disponibilité n'est pas modifiée. Par contre, si nous supposons que le système de détection est capable d'aider à localiser les défauts, alors $\mu < \nu$ et la disponibilité est améliorée.

Il ressort de cette étude que la disponibilité est en fait peu dépendante du taux de couverture. Elle est beaucoup plus sensible à l'efficacité de la maintenance. Il est possible d'améliorer la disponibilité par la mise en oeuvre de mécanismes de localisation des défauts en ligne ou hors ligne.
Mais le temps de réparation dépend aussi de critères socio-économiques: disponibilité de l'équipe de maintenance, gestion des stocks de pièces de rechange...

3.2.3 Influence des taux de réparation

Dans la pratique, les temps de réparation sont beaucoup plus faibles que les temps entre défaillances. Avec un rapport $\frac{\lambda}{\nu} = 10^{-2}$, qui est très accessible même pour les automatismes de grande taille, la perte de production en régime stationnaire est de l'ordre de 1% dans notre exemple.
Elle tombe à 1‰ si $\frac{\lambda}{\nu} = 10^{-3}$

Il est clair que la disponibilité, en temps que critère de choix d'architecture, passe rapidement au second plan dans ces conditions.
Voyons maintenant l'influence de $\frac{\lambda}{\mu}$ sur la sécurité.

En posant $\alpha = \frac{\lambda}{\mu}$ on peut écrire |1-11|

$$S(t) = \frac{1}{2}(1+K)\, e^{-\lambda \frac{1+\alpha}{2\alpha}(1+R)t} + \frac{1}{2}(1-K)\, e^{-\lambda \frac{1+\alpha}{2\alpha}(1-R)t} \qquad |1\text{-}12|$$

avec $R = \sqrt{1 - 4\frac{\alpha}{(1+\alpha)^2}(1-P_c)}$

et $K = \frac{\alpha - 2\alpha P_c - 1}{(1+\alpha)R}$

Ce qui donne $\lim_{\alpha \to 0} R = 1 - 2\alpha(1-P_c)$

et $\lim_{\alpha \to 0} K = -1$

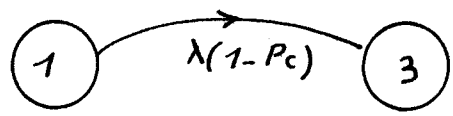
Dans ces conditions, |1.12| tend vers

$$S(t) = e^{-(1-Pc)t} \quad |1.13|$$

La relation |1.13| correspond à la sécurité d'un automatisme réparable, avec détection de défaillance, pour lequel le rapport du temps moyen de réparation au temps moyen entre pannes tend vers 0.

Le graphe correspondant est donné figure 1-10.

Sécurité E1	sain ou en réparation
Non sécurité E3	panne non révélable par le mécanisme de détection



- figure 1-10 -

Le tableau ci-après (Fig. 1-11) montre l'évolution de la sécurité en fonction de $\alpha = \frac{\lambda}{\mu}$ pour le système réparable défini figure 1-8. Dans ce tableau, tr représente le MTFMF calculé à partir de la relation |1.13|. Les valeurs reportées sont donc indépendantes du rapport λ/μ .

S(tr) correspond aux valeurs de la sécurité, calculées à partir de la relation |1.12|, à l'instant tr défini ci-dessus.

La dernière colonne du tableau donne le pourcentage d'erreur introduit en utilisant la relation approchée |1.13|.

Sur cet exemple, nous voyons que l'approximation peut être jugée acceptable dès que $\lambda/\mu \leqslant 10^{-2}$.

$\lambda = 10^{-4}$	λ/μ	Pc	tr	S(tr)	%
	10^{-1}	0	10^4	0,1887	50%
		0,5	2.10^4	0,2143	40%
		0,9	10^5	0,2310	37%
	10^{-2}	0	10^4	0,3496	5%
		0,5	2.10^4	0,3514	5%
		0,9	10^5	0,3529	5%
	10^{-3}	0	10^4	0,3660	5‰
		0,5	2.10^4	0,3662	5‰
		0,9	10^5	0,3664	5‰
	10^{-4}	0	10^4	0,3677	
		0,5	2.10^4	0,3677	
		0,9	10^5	0,3677	
	0 système approché	0	10^4	0,3679	
		0,5	2.10^4	0,3679	
		0,9	10^5	0,3679	

- figure 1-11 -

1.3.2.4 Influence du temps de latence

Dans notre exemple, nous avons admis que toute faute non détectée est non tolérée. Une telle défaillance rend donc l'automatisme dangereux. Nous n'avons pas envisagé de dégradation possible de la structure à partir de l'état E3 correspondant.

En fait, nous avons admis que la probabilité d'une deuxième défaillance avant la catastrophe est négligeable. En réalité, entre le moment où apparaît une panne détectable et le moment où l'automatisme passe en fonctionnement de sécurité, il s'écoule un certain temps appelé temps de latence (t_L).
La probabilité d'apparition d'une deuxième faute pendant le temps de latence n'est pas nulle.
L'état de la structure ne peut plus être représenté par un graphe Markovien. Le taux de transition de l'état défaillant E2 vers l'état conduisant à la réparation est en effet fonction du temps.
Nous pouvons toutefois faire un calcul de probabilité direct.

Notons t_L le temps de latence supposé constant.
Soit P21 = $\mathcal{P}$ {une deuxième défaillance se produise pendant le temps de latence d'une défaillance détectable}
Si ces défaillances ne sont pas corrélées, nous pouvons écrire:

$$P21 = \mathcal{P}\{1 \text{ défaillance détectable à l'instant } t1\} \times \mathcal{P}\{1 \text{ défaillance dans l'intervalle } t1, t1+t_L\} \qquad |1\text{-}14|$$

A partir de la structure neuve ou complètement réparée, nous avons:

$$\mathcal{P}\{1 \text{ panne détectable à } t1\} = Pc\,(1 - e^{-\lambda t1}) \qquad |1\text{-}15|$$

Supposons que la première panne ne modifie pas le taux de panne ni le taux de couverture Pc.

$$\text{Alors } \mathcal{P}\{1 \text{ défaillance dans } t1, t \text{ avec } t1 \leqslant t \leqslant t1+t_L\} = 1 - e^{-\lambda(t-t1)} \qquad |1\text{-}16|$$

A l'instant $t = t1 + t_L$, la relation |1-14| donne en tenant compte de |1-15| et |1-16|:

$$P21 = Pc\,(1 - e^{-\lambda t1})\,(1 - e^{-\lambda t_L}) \qquad |1\text{-}17|$$

Le temps de latence est toujours petit par rapport au temps entre pannes; donc λt_L est petit.

Le développement limité de |1-17| au voisinage de $\lambda t_L = 0$ pour t quelconque, conduit à:

$$P21 \# Pc \cdot \lambda t_L (1 - e^{-\lambda t}) \qquad |1\text{-}18|$$

$$P21 \# \lambda t_L \, P2 \qquad |1\text{-}19|$$

Soient

P22 = $\mathcal{P}$ { 1 panne détectable après une panne détectable latente }

P23 = $\mathcal{P}$ { 1 panne non détectable après une panne détectable latente }

Nous trouvons par un calcul semblable:

$$P22 = \lambda t_L \, Pc \, P2 \qquad |1\text{-}20|$$

$$\text{et} \quad P23 = \lambda t_L (1 - Pc) \, P2 \qquad |1\text{-}21|$$

Nous constatons que l'évaluation de la sûreté faite au paragraphe 1.3.2.1 considère implicitement le terme λt_L suffisamment petit pour que l'on puisse admettre la probabilité de panne double négligeable devant la probabilité de panne simple.
Il est clair que la probabilité d'avoir simultanément plus de deux pannes détectables non révélées est encore plus faible.

1.3.3 HYPOTHESES RETENUES POUR L'EVALUATION DE LA SECURITE DES SYSTEMES REPARABLES

Nous reprenons les hypothèses simplificatrices adoptées dans |LAP.75 §3.5.1 et annexe 7|.

1.3.3.1 Enoncé des hypothèses

H1) La probabilité que la structure reste dans un état comportant une réparation est négligeable devant celle de rester dans un état ne comportant pas de réparation.

H2) La probabilité que la défaillance d'un élément de la structure survienne pendant une réparation est négligeable.
Comme pour l'influence du temps de latence, l'hypothèse H2 implique que $\frac{\lambda}{\mu} = \lambda . t_{réparation}$ est suffisamment faible.

Ceci nous amène à considérer une troisième hypothèse.

H3) La probabilité qu'une deuxième défaillance affecte la structure pendant le temps de latence d'une défaillance détectable est négligeable.
Dans cette hypothèse, le temps de latence correspond à la durée qui sépare l'apparition de la défaillance, de l'instant où la réaction est effective (reconfiguration, mise en sécurité...).

Il résulte de ces hypothèses que toute apparition d'une panne détectable maintient la structure dans l'état qu'elle occupait précédemment comme nous pouvons le constater sur la figure 1.8.
Pour l'évaluation de la fiabilité qui correspond à la probabilité de fonctionnement dans l'intervalle (0,t), les états correspondants à une interruption de la mission sont rendus absorbants (en supprimant les réparations notamment).
Pour certaines architectures, et notamment lorsque des redondances sont mises en place, il existe des états, contenant des réparations, pour lesquels la mission est maintenue. Nous appliquons les hypothèses précédentes à ces états particuliers.
Avec ces hypothèses, l'existence de pannes multiples se réduit aux fautes non détectables.

3.3.2 Remarques sur le choix de la structure d'un automatisme réparable

Le choix d'une architecture résulte d'un compromis coût / performance.
De plus, il n'y a pas indépendance entre les diverses composantes de la sûreté.
Le compromis sécurité / fiabilité est le plus évident. Toute introduction de matériel (ou de logiciel) supplémentaire destiné à améliorer la sécurité entraîne une augmentation de la fréquence des pannes.
Cette constatation est souvent l'argument qui est avancé pour rejeter l'introduction d'un dispositif de détection / réaction au profit d'un système redondant plus complexe. A notre point de vue, cette façon d'appréhender l'amélioration de la sûreté de fonctionnement montre que le problème est en général mal posé, en ce qui concerne les automatismes.

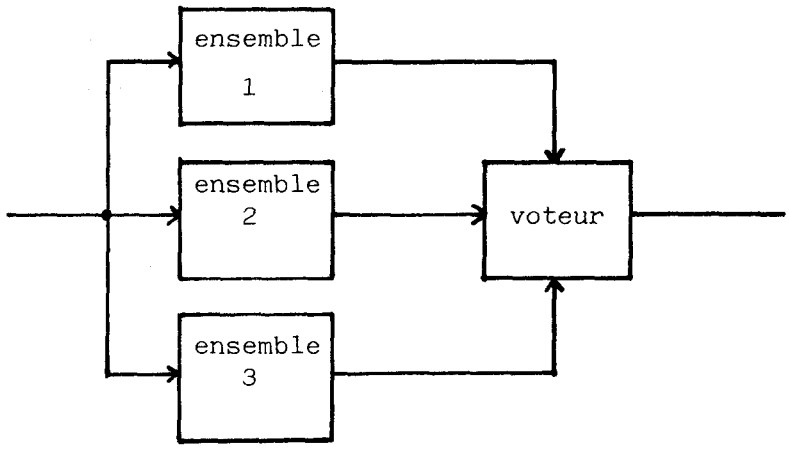
Les paramètres de la sûreté sont relatifs à l'aptitude du système à remplir la mission,plus qu'à l'état de sa structure.
Ceci est particulièrement vrai pour la fiabilité, puisque:
$R(t) = \mathcal{P}\{$ de fonctionnement à tout instant $\tau \in (0,t)\}$
est en général interprété comme probabilité de remplir la mission.
Ceci privilégie abusivement les structures redondantes au moment du choix.

Nous proposons un estimateur que nous appelons <u>la fiabilité intrinsèque</u> (inherent reliability)relatif à l'état de la structure que nous définissons comme suit:

$IR(t) = \mathcal{P}\{$que la structure ne nécessite pas de réparation à tout instant $\tau \in (0,t)\}$.

Ce paramètre diffère de la fiabilité par le fait que toute défaillance nécessite une réparation, même si elle est tolérée ou acceptée. Cette condition est indispensable pour bénéficier pleinement des avantages de la structure réparable. Allonger les temps de réparation d'une panne tolérée, c'est prendre le risque de pannes multiples.
La fiabilité intrinsèque conduit au temps moyen entre interventions du service maintenance. Il permet une meilleure estimation des coûts d'entretien. C'est donc un estimateur important au plan économique.

Par exemple, une structure <u>réparable</u> redondante massive d'ordre3 (Fig 1-12),à vote majoritaire,a une fiabilité qui tend vers 1 (MTFF = ∞) avec un voteur idéal (taux de pannes nul!).
Par contre, $IR(t) = e^{-3\lambda t}$ si λ est le taux de pannes d'un seul ensemble.



- figure 1-12 -

Nous constatons que le rapport des taux de pannes au taux de réparations λ/μ est tel que la disponibilité d'un automatisme simple est souvent suffisante. Par contre, la sécurité obtenue avec cet automatisme non tolérant peut être jugée insuffisante. Il faut alors agir pour améliorer ce paramètre.

Pour choisir une architecture, il faut, en fait, se poser la question: est-ce que la continuité de mission, totale ou partielle, est requise pour la sécurité?
Si la réponse à cette question est affirmative, l'introduction de redondances au niveau des éléments qui risquent de compromettre la sécurité est, par définition, indispensable.

Dans ce chapitre, nous avons rappelé ce qu'est la sûreté de fonctionnement. Nous avons vu comment cette sûreté peut être évaluée.
Dans les chapitres suivants, nous envisageons les moyens d'améliorer la sûreté de fonctionnement et notamment la sécurité.
Nous nous interessons plus particulièrement aux automatismes à évolution séquentielle qui correspondent à une classe d'applications très répandue. Actuellement, la partie commande de ces systèmes est pratiquement toujours confiée à des microcalculateurs industriels ou à des automates programmables industriels (A.P.I.).

Nous envisageons donc, dans un premier temps, les implications de la conception des logiciels sur la sûreté.

CHAPITRE II

INFLUENCE DE LA SYNTHESE DES DISPOSITIFS DE COMMANDE DES AUTOMATISMES A EVOLUTION SEQUENTIELLE

La réalisation d'un automatisme nécessite la mise en place d'une partie commande dont la synthèse correspond à l'opération qui consiste à passer des spécifications du cahier des charges à une réalisation. De plus en plus fréquemment, cette synthèse conduit à l'écriture d'un programme.

Il y a quelques années, les spécifications du cahier des charges étaient traduites sous forme algébrique ou tabulaire (tables de fluence, expressions régulières...). Actuellement, des méthodes plus naturelles, telle que le grafcet ou les Réseaux de Pétri permettent la modélisation des spécifications fonctionnelles. Dans la suite de l'étude, nous adoptons la représentation par le grafcet.
Dans ce chapitre, nous allons introduire des notions développées à partir de |ZAH-80| qui permettront de répondre à la question suivante:

> l'utilisation du grafcet a-t-elle une influence sur la sûreté?

Ces outils mathématiques, utilisés dans le chapitre suivant, permettent de quantifier la sûreté de fonctionnement obtenue pour certains types de défaillances.

REPRESENTATION FORMELLE DES MACHINES SEQUENTIELLES

Dans ce paragraphe, nous rappelons un ensemble de définitions formelles extraites de |ZAH-80|. Toutefois, nous avons introduit dans la représentation des machines booléennes par les équations de récurrence, les termes qui dépendent des variables internes (y) prises sous forme niée (§ 2.1.3). Cette convention vise à permettre (§2.2) la représentation par des équations de récurrence, de grafcets dans lesquels l'état actif ou inactif d'une étape est pris en compte dans une réceptivité.

2.1.1 SEQUENCES

Définition:

Soit A un ensemble non vide appelé alphabet.

Pour tout entier $n \geqslant 1$, nous désignons par N l'ensemble ordonné des entiers k tels que $1 \leqslant k \leqslant n$.

La fonction $x: N \rightarrow A$ qui associe à chaque élément de N un élément de A est une séquence de longueur n sur A notée x(1,n).

Exemple: $A = \{a,b\}$; $n = 5$

Le tableau de la figure 2.1 représente une séquence de longueur 5 sur A. x(1,5) = b a a b a

k	x(k)
1	b
2	a
3	a
4	b
5	a

Fig. 2.1

Produit cartésien de séquences:

Soient A et B deux alphabets et x(1,n), y(1,n) deux séquences de même longueur n respectivement sur A et B.

Nous appelons produit cartésien de x par y, la séquence des couples (x(1), y(1))(x(2), y(2)) ... (x(n), y(n)) sur l'alphabet A * B. Cette séquence est notée x * y(1,n).

Concaténation de séquences:

Soient deux séquences x(1,n), y(1,m) sur un même alphabet A.

La séquence x(1) ... x(n), y(1) ... y(m) notée x y(1, n+m), de longueur n+m sur A est obtenue par concaténation des séquences x(1,n) et y(1,m).

Notation: en général, lorsque la longueur de la séquence n'est pas fondamentale, on notera x la séquence x(1,n).

2.1.2 MACHINES BOOLEENNES

2.1.2.1 Définition

Nous notons X^+ et Y^+ l'ensemble des séquences de longueur quelconque sur les alphabets X et Y.
Une machine M établit une correspondance $M: X^+ \longrightarrow Y^+$ entre séquences d'entrée $x(1,n) \in X^+$ et séquence de sortie $y(1,n) \in Y^+$.
La machine conserve les longueurs.

Machine à entrées et sorties multiples:
Si la machine a p entrées et q sorties, les séquences d'entrée et de sortie sont définies respectivement sur les alphabets
$X1 * \ldots * Xp$ et $Y1 * \ldots * Yq$.
Si $X1 = \ldots Xp = Y1 = \ldots Yp = \{0,1\}$, la machine est dite binaire. L'alphabet obtenu par concaténation de p variables binaires est noté Bp.

2.1.2.2 Machine combinatoire

Une machine $M: X^+ \longrightarrow Y^+$ est dite combinatoire si:
- $\forall x \in X^+$ alors, $M(x) = 0$
- $\forall x \in X^+$ et $\forall x' \in X^+$

  $M(x\,x') = M(x)\,M(x')$

2.1.2.3 Machine séquentielle

Une machine $M: X^+ \longrightarrow Y^+$ est dite séquentielle s'il existe une correspondance $f: Y * X \longrightarrow Y$ telle que:
- $\forall y * x \in Y * X$, alors $f(y * x) \neq \emptyset$
- $\forall x \in X \Rightarrow M(x) == Y$
- $\forall x(1,n) \in X^+$, $\forall y(1,n) \in Y^+$, $n > 1$

  $y \in M(x) \Leftrightarrow y(k+1) \in f(y(k) * x(k))$ |2-1|

  pour $k = 1, \ldots, n-1$

La machine séquentielle est complètement spécifiée au sens de |ZAH-80| si pour tout couple $y * x$ $Y * X$, il existe une seule image par f dans Y.
Dans ces conditions, la relation |2-1| s'écrit:

$\forall x(1,n) \in X^+ , \forall y(1,n) \in Y^+$

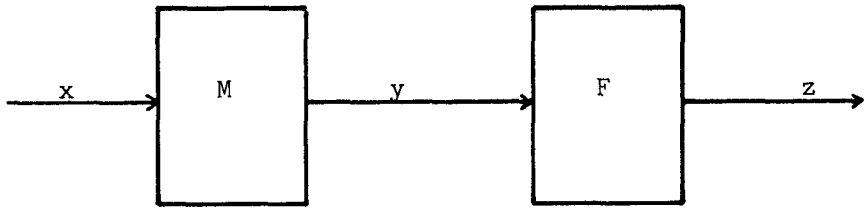
$y \in M(x) \Leftrightarrow y(k+1) = f(y(k) * x(k))$ |2-2|

pour k = 1, ...n-1

Ceci traduit le fait que la connaissance de l'élément $y(k) * x(k)$ détermine sans ambiguité la valeur du successeur $y(k+1)$.

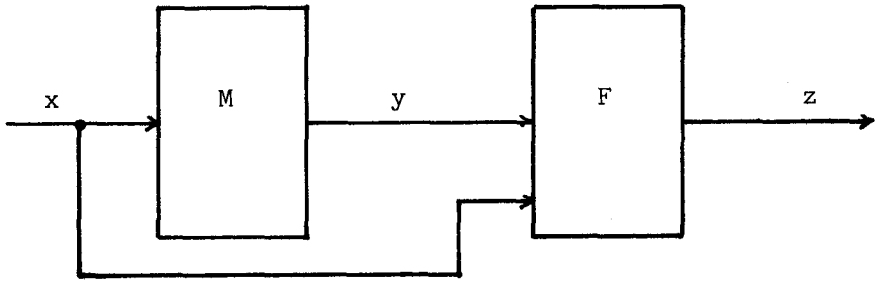
Les automatismes sont couramment présentés sous forme d'une machine composée regroupant une machine séquentielle et une machine combinatoire. Deux formes sont utilisées:

La machine de MOORE (fig. 2.2)



- figure 2.2 -

La machine de MEALY (fig. 2.3)



- figure 2.3 -

x forme l'alphabet primaire (entrée), y l'alphabet secondaire (interne), z celui de sortie.

Ces machines sont complètement spécifiées si les machines séquentielles et combinatoires qui les forment le sont également.
On généralise immédiatement l'ensemble de ces définitions aux machines à entrées / sorties multiples en remplaçant:

x par x(1) * x(2) ... * x(p) et
y par y(1) * y(2) ... * y(q)

Les méthodes de synthèse font actuellement largement appel aux machines binaires. Nous étudierons donc plus spécialement ce type de machines.

2.1.3 EQUATIONS DE RECURRENCE DES MACHINES BINAIRES

2.1.3.1 Equations de recurrence des machines séquentielles |ZAH-80|

Soit M (x(1), ... x(p)) (y(1), ... y(q)) une machine séquentielle binaire multivariable. Un état total de cette machine est un couple x * y, élément de l'alphabet Bp * Bq.
L'état interne est représenté par y à valeur dans Bq.
La correspondance M est définie par la relation |2-1|:

$\forall$ k, y(k+1) $\in$ f(y(k) * x(k))

Toute réalisation est une machine complètement spécifiée pour laquelle nous avons |2-2|, c'est-à-dire:

$\forall$ k y(k+1) = f (y(k) * x(k))

Pour chaque composante yi de y, il existe une fonction booléenne fi telle que:

yi(k+1) = fi(y(k) * x(k))

Une machine séquentielle binaire complètement spécifiée peut donc être representée par un système d'équations de récurrence booléennes d'ordre 1.

y1(k+1) = f1 (y(k) * x(k)) |2-3|

.
.
.

yq(k+1) = fq (y(k) * x(k))

2.1.3.2 Equations de récurrence des machines combinatoires

Il est clair que le fonctionnement d'une machine combinatoire binaire F(x1 ... xp) (y1 ... yq) complètement spécifiée est défini par un système d'équations de récurrence d'ordre 0.

$$y1(k) = f1(\ x1(k)...xp(k)\ ) \qquad |2\text{-}4|$$

$$\vdots$$

$$yq(k) = fq(\ x1(k)...xq(k)\ )$$

2.1.3.3 Représentation matricielle d'une machine de Mealy

On a composition d'une machine séquentielle $M1(y_* x)$ et d'une machine combinatoire $M2(z_* y_* x)$. Les équations de récurrence sont de la forme:

pour M1 $\quad y1(k+1) = f1(\ y(k)\ {}_*\ x(k)\ )$

$$\vdots$$

$$yq(k+1) = fq(\ y(k)\ {}_*\ x(k)\ )$$

pour M2 $\quad z1(k) = g1(\ y(k)\ {}_*\ x(k)\ )$

$$\vdots$$

$$zr(k) = gr(\ y(k)\ *\ x(k)\ )$$

On déduit immédiatement les équations de récurrence d'une machine de MOORE en remarquant que M2 est une machine combinatoire sur l'alphabet binaire $(z_* y)$.

En utilisant les propriétés du développement de Shannon, nous pouvons écrire:

$$y1(k+1) = f^1_{11}\ (\overset{q}{\underset{i=2}{X}}(yi(k))^*\ x(k)).y1(k) + \ldots + f^1_{1q}(\overset{q-1}{\underset{i=1}{X}}(yi(k))^*x(k)).yq(k)$$

$$+ f^2_{12}(\overset{k}{\underset{i=2}{X}}(\ yi(k))^*\ x(k)).\overline{y1(k)} + \ldots + f^2_{1q}(\overset{q-1}{\underset{i=1}{X}}(yi(k))^*x(k)).\overline{yq(k)} \qquad |2\text{-}5|$$

Les symboles * et X représentent le produit cartésien.

Les symboles de multiplication et d'addition correspondent évidemment aux opérateurs ET et OU.
Les machines séquentielles et combinatoires se mettent alors sous forme de matrices de fonctions booléennes.

A chacune de ces fonctions peut être associée une expression booléenne.

$$M1 \rightarrow \left| y(k+1) \right| = \mathbb{F}1 . \left| y(k) \right| + \mathbb{F}2 . \left| \overline{y(k)} \right| \qquad |2\text{-}6|$$

$$M2 \rightarrow \left| z(k) \right| = \mathbb{G}1 . \left| y(k) \right| + \mathbb{G}2 . \left| y(k) \right| \qquad |2\text{-}7|$$

$\mathbb{F}1$, $\mathbb{F}2$, $\mathbb{G}1$ et $\mathbb{G}2$ sont des matrices de fonctions booléennes dont les termes

f^1_{ij}, f^2_{ij}, g^1_{ij} et g^2_{ij} sont définis sur B_{p+q-1} à partir des éléments de l'alphabet:

$$\bigtimes_{\substack{h=1 \\ h\neq i}}^{q} (yh(k))^* x(k).$$

Les expressions en f^1_{ii} correspondent aux conditions de maintien de la variable interne yi.

Exemple:

Soit une machine séquentielle définie par les équations de récurrence suivantes:

$$y1(k+1) = 0$$

$$y2(k+1) = x1(k)\, y1(k) + \overline{x2}(k)\, y2(k)$$

$$y3(k+1) = x2(k)\, y4(k)\, y2(k)$$

$$y4(k+1) = \overline{x1}(k)\, \overline{y3}(k)\, y2(k) + y3(k)$$

Par application du théorème de Shannon, on peut écrire notamment:

$$y4(k+1) = 0\, y1(k) + \overline{x1}(k)\, \overline{y3}(k) + 1\, y3(k) + 0\, y4(k) + 0\, \overline{y1}(k) + 0\, \overline{y2}(k) + \overline{x1}(k)\, y2(k)\, \overline{y3}(k) + 0\, \overline{y4}(k)$$

On obtient:

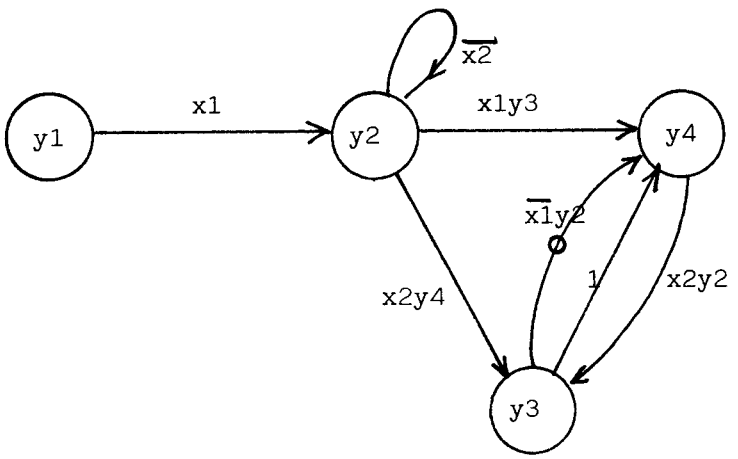
$$\mathbb{F}1 = \begin{vmatrix} 0 & 0 & 0 & 0 \\ x1 & \overline{x2} & 0 & 0 \\ 0 & x2y4 & 0 & x2y2 \\ 0 & \overline{x1}\overline{y3} & 1 & 0 \end{vmatrix} \qquad \mathbb{F}2 = \begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \overline{x1}y2 & 0 \end{vmatrix}$$

Le système d'équations de récurrence représenté par $\mathbb{F}_1$ peut être mis sous forme d'un graphe dont les arcs sont étiquetés par f^1_{ij} (fig. 2.4a).

Le système d'équations représenté par $\mathbb{F}_2$ peut être mis sous forme d'un graphe utilisant des arcs inhibiteurs (fig. 2.4b).

- figure 2.4 -

La superposition de ces deux graphes représente la machine séquentielle.
Dans notre exemple, nous obtenons le graphe de la figure 2.5.



- figure 2.5 -

Remarque:
La représentation utilisée peut conduire à une confusion avec un graphe d'état. A priori, plusieurs variables internes peuvent être simultanément à 1.

2.1.3.4 Simplification des expressions de récurrence pour des machines binaires non linéaires.

Machine binaire linéaire: |ZAH-80|

Une machine est dite B. linéaire si tous les termes des matrices $\mathbb{F}1$ et $\mathbb{F}2$ sont indépendants de l'état interne, donc des yi.

Proposition:

Dans les matrices de récurrence d'une machine non B.linéaire, il existe des expressions redondantes dont la suppression est sans effet sur le résultat obtenu.

Soit une équation de récurrence:

$$yi(k+1) = f^1_{i1}\, y1(k)+\ldots+f^1_{iq}\, yq(k)+f^2_{i1}\, \overline{y1(k)}+\ldots+f^2_{iq}\, \overline{yq(k)} \qquad |2\text{-}8|$$

relative à une machine M1 non B-linéaire.

Soit deux rangs j et n tels que yi(k+1) soit de la forme:

$$yi(k+1)=F(Y_{jn}*X)+f_{\bar{j}\bar{n}}\overline{y}_j y_n+f_{j\bar{n}}y_j\overline{y}_n+f_{\bar{j}\bar{n}}\overline{y}_j\overline{y}_n+f_{jn}y_jy_n+f_jy_j+f_ny_n+$$
$$f_{\bar{j}}\overline{y}_j+f_{\bar{n}}\overline{y}_n \qquad |2\text{-}9|$$

où Y_{jn} représente l'alphabet interne limité aux éléments de Y autres que y_n et y_j. Les fonctions f_{jn}, f_j, f_n sont définies sur $Y_{jn}*X$. Le développement de Shannon donne:

$$yi(k+1)=F(X*Y_{jn})+(f_{jn}y_j+f_{j\bar{n}}\overline{y}_j+f_n)y_n+(f_{jn}y_n+f_{j\bar{n}}\overline{y}_n+f_j)y_j+$$
$$(f_{\bar{j}n}y_j+f_{\bar{j}\bar{n}}\overline{y}_j+f_{\bar{n}})\overline{y}_n+(f_{\bar{j}n}y_n+f_{\bar{j}\bar{n}}\overline{y}_n+f_{\bar{j}})\overline{y}_j \qquad |2\text{-}10|$$

Le développement d'une telle expression montre qu'il existe deux termes produits tels que $f_{jn}y_jy_n$.

Le résultat obtenu est inchangé en enlevant l'expression

$$f_{jn}y_j \text{ OU } f_{jn}y_n \text{ de } \mathbb{F}1$$

qui constitue ce que nous appelons un terme redondant.

Il existe dans l'expression de yi(k+1) quatre de ces termes pris dans $\mathbb{F}1$ et $\mathbb{F}2$.

Les termes redondants doivent rester représentés une fois.

Exemple:

Dans l'exemple précédent, nous pouvons choisir:

$$\mathbb{F}_1 = \begin{vmatrix} 0 & 0 & 0 & 0 \\ x1 & \overline{x2} & 0 & 0 \\ 0 & x2y4 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{vmatrix} \qquad \mathbb{F}_2 = \begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \overline{x1}y2 & 0 \end{vmatrix}$$

qui donne le graphe de la figure 2.6

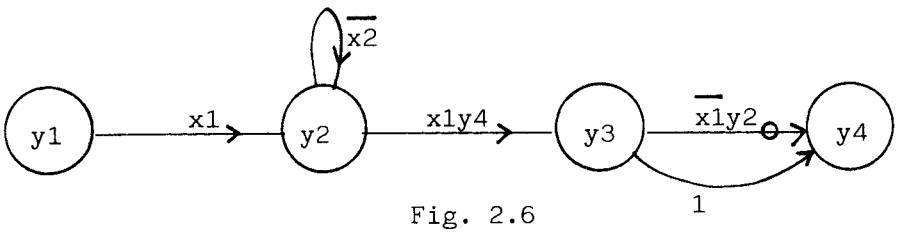


Fig. 2.6

ou

$$\mathbb{F}_1 = \begin{vmatrix} 0 & 0 & 0 & 0 \\ x1 & \overline{x2} & 0 & 0 \\ 0 & x2y4 & 0 & 0 \\ 0 & \overline{x1}\overline{y3} & 1 & 0 \end{vmatrix} \qquad \mathbb{F}_2 = \begin{vmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{vmatrix}$$

qui donne un graphe sans arc inhibiteur (fig 2.7)

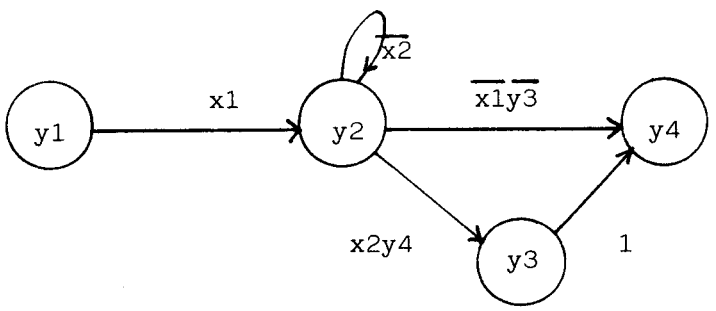


Fig. 2.7

Un traitement de même nature permet d'éliminer les termes redondants des matrices $\mathbb{G}_1$ et $\mathbb{G}_2$.

Définition:

Nous dirons qu'un système est stable pour un état secondaire donné, défini par un élément de l'alphabet secondaire y, s'il est incapable de quitter cet état sans modification d'au moins une entrée.

Comme dans |ZAH-80|, nous nous limiterons dans la suite de l'exposé aux machines totalement stables pour l'état secondaire
$y1 * \ldots * yp = 0 * 0 \ldots * 0$.
Par totalement stable, il faut entendre, incapable de quitter cet état secondaire, quelque soit l'état des entrées.

Proposition:

Pour de telles machines, il est toujours possible de choisir les termes redondants de façon à rendre la matrice $\mathbb{F}2$ identiquement nulle.

Cette propriété est évidente pour une machine B-linéaire.
Soit la machine non B-linéaire, totalement stable pour l'état secondaire nul, envisagée ci-dessus.La propriété de stabilité de cette machine permet de tirer de |2.9| les égalités suivantes:

$$f_{\bar{n}} = f_{\bar{j}} = f_{\bar{j}\bar{n}} = 0.$$

Tout terme non nul de $\mathbb{F}2$ tel que $f_{\bar{j}n} y_n$ est redondant avec le terme correspondant (ici $f_{\bar{j}n} \bar{y}_j$) de $\mathbb{F}1$. La propriété étudiée découle de cette constation.

2.1.3.5 Machines réceptives

Une machine est réceptive si les équations de récurrence sont telles que:

$$\forall i \qquad fii + \overline{\sum_{j \neq i} fji} = fii \qquad |2\text{-}12|$$

Cette relation peut s'écrire:

$$\overline{\sum_{j \neq i} fji} \leqslant fii \qquad |2\text{-}13|$$

En général, nous avons $\overline{\sum_{j=i} fji} = fii$

Proposition:

Une machine réceptive est complètement spécifiée.

Cette proposition évidente se démontre ainsi:

$$\forall i \quad \sum_{j=1}^{q} fji = fii + \sum_{\substack{j=1 \\ j \neq i}}^{q} fji$$

Si la machine est réceptive, nous avons, en tenant compte de |2-12|:

$$fii + \overline{\sum_{j \neq i} fji} + \sum_{j \neq i} fji = 1$$

La réciproque est tout aussi évidente.

Remarque:

La relation |2-12| doit être appliquée à partir du système d'équations de récurrence contenant tous les termes redondants.

Exemple:

Soit le système d'équations de récurrence

$y1(k+1) = x2(k)\ y3(k) + \overline{x1(k)\ y2(k)}\ y1$

$y2(k+1) = x2(k)\ y3(k)$

$y3(k+1) = x1(k)\ y1(k)\ y2(k) + \overline{x2(k)}\ y3(k)$

En conservant les termes redondants, nous avons la matrice:

$$\begin{vmatrix} \overline{x1y2} & 0 & x2 \\ 0 & 0 & x2 \\ x1y2 & x1y1 & \overline{x2} \end{vmatrix}$$

qui donne

$f22 = 0$ et $f22 + \overline{\sum_{j \neq 2} fj2} = \overline{f32} = \overline{x1y1} \neq f22$

Le système n'est pas réceptif.

L'une des matrices obtenues après élimination des termes redondants est:

$$\begin{vmatrix} \overline{x1y2} & 0 & x2 \\ 0 & 0 & x2 \\ x1y2 & 0 & \overline{x2} \end{vmatrix}$$

si nous tirons $f22 = 0$; $\overline{\sum_{j\neq 2} fj2 = 0}$.

Le système semble réceptif, ce qui est faux.

2.2 REPRESENTATION D'UN GRAFCET PAR LES EQUATIONS DE RECURRENCE

A l'étape i, il est possible d'associer une variable booléenne yi. L'ensemble de ces variables forme l'alphabet interne. Le grafcet est alors un graphe booléen dont nous savons qu'il est réceptif.

2.2.1 DETERMINATION DES EQUATIONS DE RECURRENCE

Pour obtenir un système d'équations de récurrence à partir d'un grafcet, il suffit:

a) d'affecter à chaque étape i une variable secondaire yi

b) pour chaque étape, de faire $yi(k+1) = \sum_{j=1}^{n} Rij\ yj$

Rij est le produit de la réceptivité associée à la transition placée sur l'arc yj, yi par chacune des variables associées aux étapes antécédentes autres que j.

yj est l'une des étapes antécédentes à cette transition.

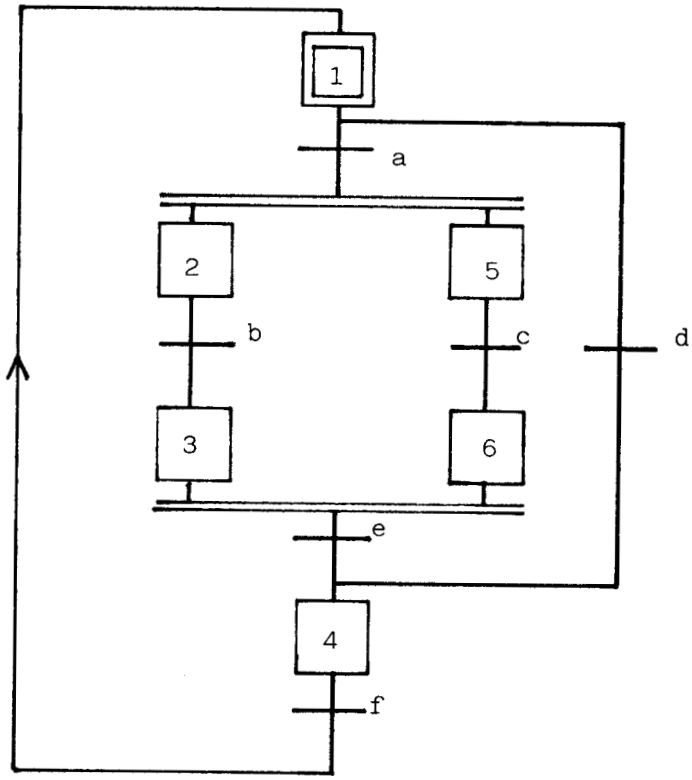
c) transformer les éléments diagonaux, de façon à rendre le système booléen réceptif en posant:

$$\forall i \quad fii = Rii + \overline{\sum_{i\neq j} Rji} \qquad |2\text{-}14|$$

d) éliminer éventuellement les termes redondants.

Exemple:

Soit un grafcet défini figure 2.8:



- figure 2.8 -

Par application de l'étape b, nous avons le système d'équations suivant:

$$y1(k+1) = f.y4(k)$$
$$y2(k+1) = a.y1(k)$$
$$y3(k+1) = b.y2(k)$$
$$y4(k+1) = d.y1(k) + e.y6(k).y3(k) + e.y3(k).y6(k)$$
$$y5(k+1) = a.y1(k)$$
$$y6(k+1) = c.y5(k)$$

ce qui correspond à la matrice ci-dessous:

$$\begin{vmatrix} 0 & 0 & 0 & f & 0 & 0 \\ a & 0 & 0 & 0 & 0 & 0 \\ 0 & b & 0 & 0 & 0 & 0 \\ d & 0 & e\ y6 & 0 & 0 & e\ y3 \\ a & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c & 0 \end{vmatrix}$$

Nous calculons alors les éléments diagonaux par application de |2-14|.

Par exemple: $f33 = R33 + \overline{\sum Rj3}$

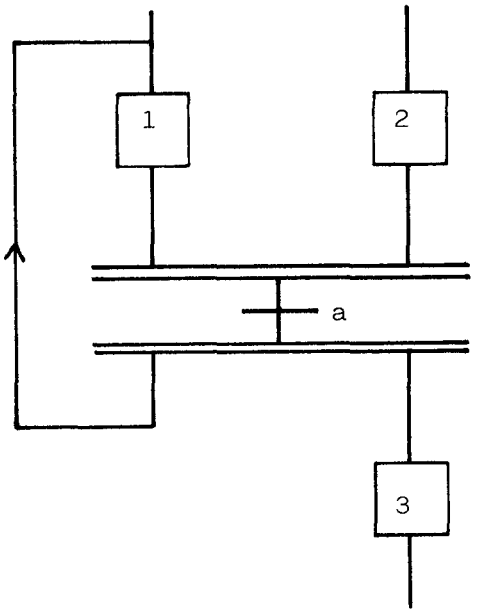
$= 0 + \overline{e\ y6}$

Ce traitement nous donne la matrice suivante dans laquelle l'un des deux termes redondants (cerclés dans le tableau), peut être remplacé par 0.

$$\begin{vmatrix} \overline{a + d} & 0 & 0 & f & 0 & 0 \\ a & \overline{b} & 0 & 0 & 0 & 0 \\ 0 & b & \overline{e\ y6} & 0 & 0 & 0 \\ d & 0 & (e\ y6) & \overline{f} & 0 & (e\ y3) \\ a & 0 & 0 & 0 & \overline{c} & 0 \\ 0 & 0 & 0 & 0 & c & \overline{e\ y3} \end{vmatrix}$$

Autre exemple:

Soit une partie de grafcet représenté figure 2.9:



- figure 2.9 -

Par application du pas b de l'algorithme, on trouve l'élément de tableau suivant:

a y2	0	0	. . .
0	0	0	. . .
a y2	a y1	0	. . .

D'après |2 -14|:

$$f11 = R11 + \overline{\sum_{j\neq 1} R1j}$$

Nous avons R11 = a y2 et R13 = a y2. La relation précédente s'écrit:

$$f11 = a\ y2 + \overline{(a\ y2 + \sum_{\substack{j\neq 1 \\ j\neq 3}} R1j}$$

$$f11 = a\ y2 + \overline{a\ y2} \cdot \overline{\left[\sum_{\substack{j\neq 1 \\ j\neq 3}} R1j \right]}$$

$$f11 = a\ y2 + \overline{\sum_{\substack{j\neq 1 \\ j\neq 3}} R1j}$$

Le terme a y2 disparaît dans la mesure où il étiquette un arc qui ne désactive pas y1.

Remarques:

- Le système d'équations obtenu constitue une interprétation algébrique du grafcet. Comme dans | AFC-82|, on peut admettre que ces équations s'obtiennent directement par simple lecture du graphe.
- Un même grafcet peut se représenter selon différents systèmes d'équations équivalentes selon les termes redondants conservés.

2.2.2 ETATS INTERNES D'UNE MACHINE SPECIFIEE PAR UN GRAFCET

L'état interne d'une machine spécifiée par un grafcet est donné par l'état actif ou inactif de l'ensemble de ses étapes. Associer à chaque étape un booléen prenant la valeur "1" si et seulement si l'étape est active, c'est choisir un codage des états internes.

Etat interne stable:

Un état interne est dit stable s'il ne peut être modifié que par modification d'au moins une entrée. Un état interne stable est caractérisé par:

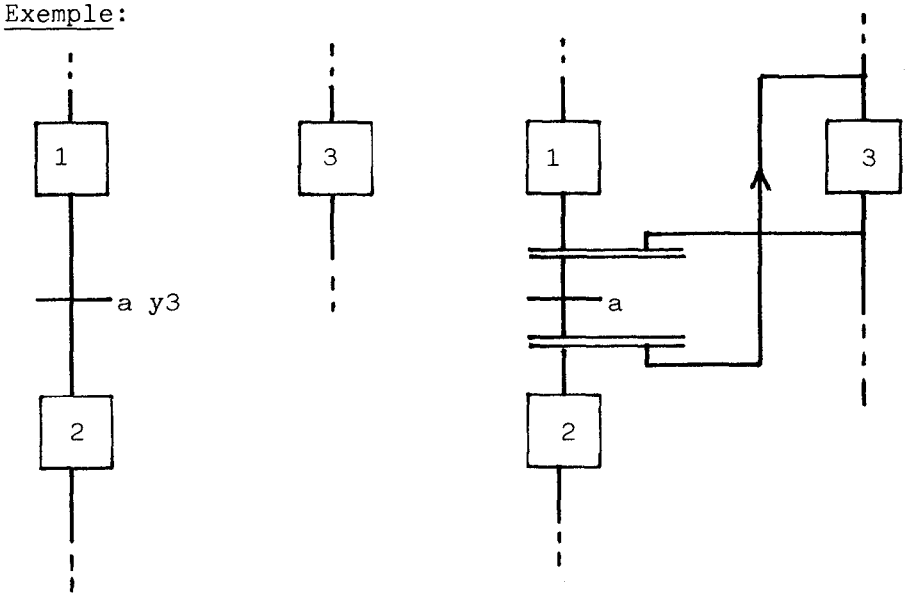
$$Y_{(k+1)} = Y_{(k)} \qquad |2\text{-}15|$$

Un système est dans un état interne totalement stable si la relation |2-15| est vérifiée pour toutes les combinaisons de valeur des p entrées. Ce système est alors bloqué, ce qui est rarement recherché, sauf peut être pour placer le système dans un fonctionnement de sécurité.
Par contre, un système peut être dans un état totalement instable (cas d'un événement toujours vrai).

Remarques:

- Il est admis dans la norme grafcet de prendre en compte l'état des étapes dans les réceptivités. Cet état peut apparaître sous forme affirmée ou niée. Cette convention permet un allégement du graphisme (figure 2.10).

Exemple:



- figure 2.10 -

Il est clair que dans le cas général, la matrice $\mathbb{F}_2$ n'est pas identiquement nulle. Toutefois, si on exclut la possibilité d'utiliser des transitions sources (non précédées d'étapes), le grafcet est totalement stable pour l'état secondaire nul (aucune étape active). Dans ces conditions, la matrice $\mathbb{F}_2$ peut toujours être rendue identiquement nulle.

- La même démarche peut s'appliquer aux matrices $\mathbb{G}_1$ et $\mathbb{G}_2$ relatives à la machine combinatoire.

En isolant le combinatoire général indépendant de l'état secondaire, nous limitons la représentation du grafcet à trois matrices:
une matrice $\mathbb{F}$ pour la machine séquentielle,
une matrice $\mathbb{G}$ pour le combinatoire local
une matrice ligne $\mathbb{H}$ pour le combinatoire général.

2.3 INFLUENCE DE LA SYNTHESE DES PARTIES COMMANDES SUR LA SURETE DE FONCTIONNEMENT

Le cahier des charges d'un automatisme à évolution séquentielle définit l'ordonnancement des actions et des tâches en fonction de l'arrivée d'événements normaux internes (C.R. des P.O.) ou externes (événements de conduite).
La prise en compte des différents modes de marches et notamment du fonctionnement en sécurité, l'introduction de contraintes technologiques, conduisent par affinements successifs vers la répartition et la spécification des fonctions de commande afférentes à différents sous-systèmes de l'automatisme. Ces fonctions de commande sont généralement réparties entre un, voire plusieurs sous-systèmes ayant une fonction exclusive de P.C. et des préactionneurs qui ajoutent à cette fonction, un rôle d'adaptation aux actionneurs. Dans cette catégorie, nous trouvons les contacteurs, distributeurs, régulateur, onduleurs...
Sans entrer plus avant dans l'aspect méthodologique qui ne constitue pas la préoccupation essentielle de notre travail, nous pouvons admettre que l'étude de l'automatisme conduit à la définition des spécifications relatives à chaque sous P.C. dont le comportement est lié à des séquences d'événements normaux. Ces sous-ensembles sont incomplètement spécifiés dans la mesure où, pour certaines valeurs $x(k) * y(k)$ réputées inaccessibles en l'absence de défaillance, toute valeur de Y est acceptable pour $y(k+1)$.
Avec les méthodes de synthèse de type Hoffman, l'objectif visé est la limitation du nombre de composants. Deux axes complémentaires sont empruntés.

Dans un premier temps, on cherche à diminuer le nombre d'états internes et, du même coup, le nombre de variables secondaires constituant l'alphabet Y. Ce résultat est obtenu en exploitant les possibilités de choix qui résultent de la non définition de la machine pour les événements anormaux.

Dans un deuxième temps, après avoir choisi un codage des états internes (constitution de l'alphabet Y), les fonctions combinatoires déduites de ces choix successifs sont simplifiées. Là encore, les simplifications sont menées en exploitant au mieux le fait que les fonctions obtenues sont, généralement, incomplètement définies.

L'inconvénient de cette approche réside dans l'absence de prévision de l'évolution obtenue en cas d'apparition d'un événement anormal, même prévisible, liée à une défaillance.

L'utilisation d'outils tels que le grafcet ou les Réseaux de Pétri pour la modélisation du cahier des charges, rend naturellement les parties commandes ainsi spécifiées, réceptives et sensibles aux seuls événements normaux. Si cette propriété est respectée dans la phase de synthèse et d'implantation, la commande obtenue sera indifférente à tout événement anormal correspondant à une erreur de C.R.. Cette propriété est incontestablement bénéfique pour la sécurité, dans la mesure où elle évite la propagation de l'erreur par la P.C.. Si l'on admet que l'utilisation de ces outils de description permet également de limiter les risques d'erreur de conception par une représentation plus claire des spécifications, il est certain que ces méthodes naturelles de transcription du cahier des charges conduisent, pour toutes ces raisons, à une amélioration de la sécurité des automatismes.

Si le risque de propagation de l'erreur de C.R. est limité par l'utilisation des machines réceptives, il est clair que la persistance de cette erreur doit conduire, à terme, à une prise en compte de l'information erronée. Pour éviter réellement cette propagation, il faut alors supposer l'existence d'un mécanisme de détection de l'anomalie, éventuellement représenté par un opérateur, ou la disparition naturelle de l'erreur. Cette deuxième alternative constitue l'hypothèse de base à l'étude du chapitre suivant.

La propagation par la P.C. des erreurs de C.R. ne constitue pas la seule limitation à la sécurité des automatismes.
Dans la suite de notre étude, nous analysons les différentes sources d'erreurs qui peuvent affecter les automatismes. Des propositions, visant à en limiter les effets néfastes sur la sûreté de fonctionnement sont proposées et discutées.

CHAPITRE III

INFLUENCE DES PANNES NON CONSISTANTES

Parmi les pannes les plus souvent prises en compte par les automaticiens ouverts aux problèmes de la sécurité, se trouvent celles qui sont consécutives à une défaillance de capteur.
Un tel défaut est à l'origine d'un compte-rendu défectueux qui peut conduire la partie commande à générer des sorties érronées.
Nous assistons ici au phénomène de propagation des erreurs.
Pour éviter cette propagation, différentes solutions sont envisageables. Dans ce chapitre, nous allons essayer de limiter le risque face à une panne non consistante.

Nous disons qu'une panne est consistante, si elle ne peut être éliminée que par une réparation.
Les pannes non consistantes regroupent alors les pannes fugitives, intermittentes, etc...
Elles peuvent être dues à des défaillances particulières: susceptibilité électromagnétique, mauvaises connexions électriques..., ou à des incidents mécaniques: manipulation accidentelle d'un capteur par un opérateur ou par la chute d'une pièce...
Pour évaluer l'aptitude d'un grafcet à ignorer ces pannes non consistantes, nous proposons de calculer pour chaque graphe un coefficient estimateur. Ceci nous amène à préciser les notions de réceptivité et de sensibilité. A la fin de ce chapitre, deux règles sont proposées; en les utilisant, le concepteur doit tendre naturellement vers un graphe optimum vis-à-vis du comportement relatif aux pannes non consistantes.

3.1 RECEPTIVITE ET SENSIBILITE A UNE ENTREE

L'état interne d'une machine séquentielle à l'instant k est défini par l'ensemble des valeurs y(k).

Réceptivité:

Un système placé dans un état interne donné est réceptif à une entrée si une modification de cette entrée provoque un changement d'état interne.

Un système est globalement réceptif à une entrée si la définition précédente s'applique quelque soit l'état interne du système.

Un système est généralement globalement réceptif à une entrée de forçage type:" mise aux conditions initiales".

Sensibilité:

Un système, placé dans un état interne donné, est dit sensible à une entrée si toute modification de cette entrée entraîne la variation d'au moins une sortie sans modification de l'état interne.

Un système est globalement sensible à une entrée si la définition précédente s'applique quelque soit l'état interne du système.

La sensibilité caractérise la machine combinatoire; la réceptivité est relative à la partie séquentielle du système.

3.2 EXPRESSION ALGEBRIQUE DE LA RECEPTIVITE ET DE LA SENSIBILITE

3.2.1 DEFINITION D'UNE FONCTION DERIVEE

Soient f(x1 ... xn), une expression booléenne définie sur B^n et deux instants k, k+1.

Nous appelons nombre dérivé de f par rapport à la variable xi le booléen défini par:

$$\frac{\partial f}{\partial xi} = f(x1(k), \ldots, xi(k), \ldots, xn(k)) \oplus f(x1(k), \ldots, xi(k+1), \ldots, xn(k)) \qquad |3\text{-}1|$$

L'opérateur $\oplus$ est le dilemme (OU exclusif).

Il est clair que ce nombre prend la valeur 1, si et seulement si, pour toutes variables autres que xi maintenues constantes, la modification de xi, entre les instants k et k+1, entraîne une modification de f.

Proposition:
Il existe une fonction booléenne, définie sur B^{n-1}, qui donne à chaque instant k la valeur du nombre dérivé. Cette fonction est appelée fonction dérivée par rapport à xi.

Soit f une fonction des n variables xi; i = 1, ..., n.
Cette fonction s'écrit:
$f = f1\ xi + f2\ \overline{xi}$
f1,f2 fonction de n-1 variables xj; $j \neq i$, j = 1 ...n.

La relation |3-1| s'exprime alors:

$$\frac{\partial f}{\partial xi} = [\ f1(k)\ xi(k)+f2(k)\ \overline{xi(k)}\ ] \oplus [\ f1(k)\ xi(k+1)+f2(k)\ \overline{xi(k+1)}\ ] \qquad |3\text{-}2|$$

Après développement, on montre que

$$\frac{\partial f}{\partial xi} = [\ f1(k) \oplus f2(k)\ ] \ .\ [\ xi(k) \oplus xi(k+1)\ ] \qquad |3\text{-}3|$$

La fonction f1 ⊕ f2 définie sur les n-1 variables autres que xi prend la valeur du nombre dérivé lorsque xi(k) ⊕ xi(k+1) = 1.
Cette fonction notée $\frac{\partial f}{\partial xi}$ sera appelée fonction dérivée de f par rapport à xi.
Pour obtenir la fonction dérivée $\frac{\partial f}{\partial xi}$, il suffit de déterminer la fonction f1 obtenue en remplaçant xi par 0 dans f. Puis, de façon semblable, d'élaborer la fonction f2 en donnant à xi la valeur 1.
La fonction dérivée est alors obtenue en faisant:

$$\frac{\partial f}{\partial xi} = f1 \oplus f2 \qquad |3\text{-}4|$$

Dans la suite de l'exposé, la relation |3-3| sera écrite selon la forme ci-dessus |3-4|. Le terme xi(k) ⊕ xi(k+1) étant implicite.

Exemples:
- $f = a + b$; $f2 = b$, $f1 = 1$

$$\frac{\partial f}{\partial a} = \overline{b}$$

- $f = ab$; $f2 = 0$, $f1 = b$

$$\frac{\partial f}{\partial a} = b$$

- $f = \overline{ab}$; $f2 = 1$, $f1 = \overline{b}$

$$\frac{\partial f}{\partial a} = b$$

3.2.2 RECEPTIVITE ET SENSIBILITE PAR RAPPORT A UNE VARIABLE

Soit une machine séquentielle dont l'état interne, stable à l'instant k, est défini par les équations de récurrence:

$$Y(k+1) = F(k)\ Y(k) \qquad |3\text{-}5|$$

L'état interne étant stable, nous avons par hypothèse:

$$Y(k+1) = Y(k) \qquad |3\text{-}6|$$

Supposons qu'à l'instant k+1 la variable xi et elle seule change de valeur.

Alors, $Y(k+2) = F(k+1)\ Y(k+1)$

Si le système était réceptif à xi, alors:

$$Y(k+2) \neq Y(k+1)$$
$$\neq Y(k)$$

L'entrée xi est supposée maintenue à la valeur xi(k+1) jusqu'à l'instant k+2, ce qui entraîne x(k+2) = x(k+1)

On a donc:

$$Y(k+2) \oplus Y(k+1) = F(k+1)\ Y(k+1) \oplus F(k)\ Y(k) \qquad |3\text{-}7|$$

qui, d'après |3-6|, peut s'écrire:

$$Y(k+2) \oplus Y(k+1) = (F(k+1) \oplus F(k)) \cdot Y(k) \qquad |3\text{-}8|$$

qui donne:

$$(xi(k+2) \oplus xi(k))(Y(k+2) \oplus Y(k)) = (xi(k+1) \oplus xi(k))(F(k+1) \oplus F(k))\ Y(k)$$

qui s'écrit avec les conventions précédentes:

$$\frac{\partial}{\partial xi}\left(\mathbb{Y}\right)(k+2) = \left(\frac{\partial}{\partial xi}\mathbb{F}\right)(k) \cdot \mathbb{Y}(k) \qquad |3\text{-}9|$$

Le terme $\mathbb{Y}(k+2)$ indique simplement le passage par l'état instable correspondant à la nouvelle valeur de xi avant changement d'état interne.

Pour un système défini par un grafcet, nous élaborons les matrices de réceptivité et de sensibilité par rapport à une variable xi. Ces matrices de fonctions booléennes sont notées:

$\frac{\partial \mathbb{F}}{\partial xi}$ pour la réceptivité

$\frac{\partial \mathbb{G}}{\partial xi}$ pour la sensibilité relative au combinatoire local

$\frac{\partial \mathbb{H}}{\partial xi}$ pour la sensibilité liée au combinatoire général.

3.2.2.1 Réceptivité par rapport à une variable

Une machine est dite réceptive par rapport à la variable xi pour un état total donné si:

$$\left(\frac{\partial \mathbb{F}}{\partial xi}\right)(k) \cdot \mathbb{Y}(k) \neq 0 \qquad |3\text{-}10|$$

Un système est dit potentiellement réceptif à xi pour un état interne donné s'il existe au moins une combinaison de valeurs des p-1 entrées autres que xi pour laquelle le système est réceptif.
Un système est inconditionnellement réceptif à xi pour un état interne donné si la relation |3-10| est vérifiée pour toutes valeurs des p-1 autres entrées que xi.

3.2.2.2 Sensibilité par rapport à une variable

- Une machine est sensible par rapport à la variable xi si pour un état total donné

$$\left(\frac{\partial \mathbb{G}}{\partial xi}\right)(k)\ \mathbb{Y}(k) \neq 0 \quad \text{et / ou} \quad \left(\frac{\partial \mathbb{H}}{\partial xi}\right) k \neq 0 \qquad |3\text{-}11|$$

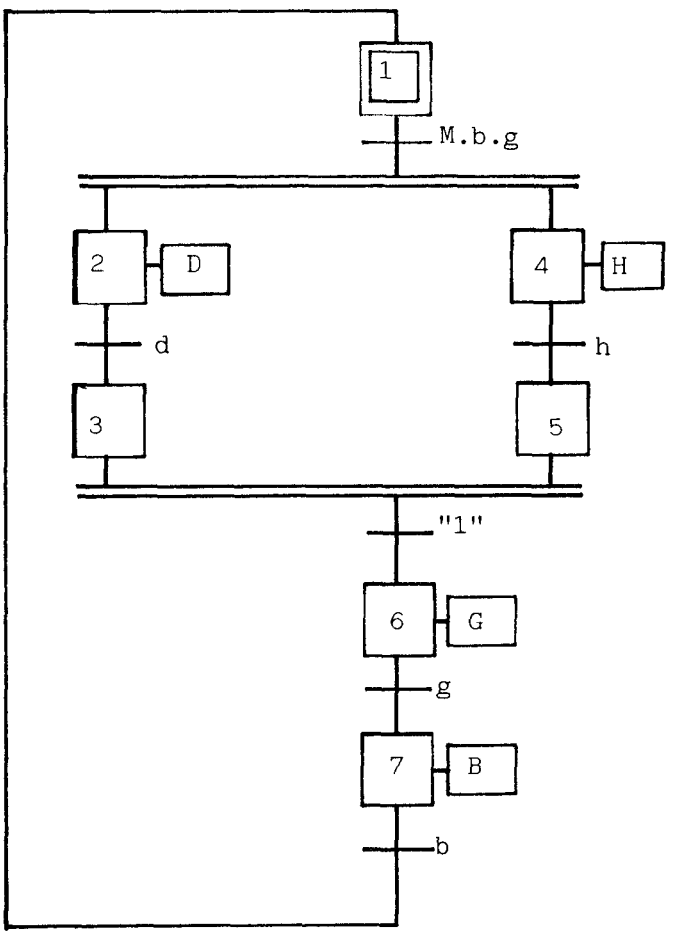
- Comme pour la réceptivité, on dira qu'un système est potentiellement sensible à une variable xi pour un état interne donné s'il existe au moins une combinaison de valeurs des p-1 entrées autres que xi qui vérifie les équations |3-11|.

- Un système est inconditionnellement sensible à xi pour un état interne donné si les relations |3-11| sont vérifiées pour toutes valeurs des entrées autres que xi.

Remarque:
Un système inconditionnellement sensible par le combinatoire général est globalement sensible au sens du paragraphe 1.

3.2.2.3 Exemples

- Soit le grafcet représenté figure 3.1 |BLA-80|



- figure 3.1 -

Les matrices $\mathbb{F}$, $\mathbb{G}$, $\mathbb{H}$ sont les suivantes:

$$\mathbb{F} = \left|\begin{array}{ccccccc}
\overline{Mbd} & 0 & 0 & 0 & 0 & 0 & b \\
Mbd & \overline{d} & 0 & 0 & 0 & 0 & 0 \\
0 & d & \overline{y5} & 0 & 0 & 0 & 0 \\
Mbd & 0 & 0 & \overline{h} & 0 & 0 & 0 \\
0 & 0 & 0 & h & \overline{y3} & 0 & 0 \\
0 & 0 & y5 & 0 & (0) & \overline{g} & 0 \\
0 & 0 & 0 & 0 & 0 & g & \overline{b}
\end{array}\right|$$

(0) terme redondant éliminé

$$\begin{pmatrix} H \\ B \\ D \\ G \end{pmatrix} = \mathbb{G} \begin{pmatrix} y1 \\ \vdots \\ y7 \end{pmatrix}$$

$$\mathbb{G} = \left|\begin{array}{ccccccc}
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0
\end{array}\right|$$

$$\mathbb{H} = | \; 0 \; |$$

L'étude du comportement vis-à-vis de d conduit à écrire:

$$\frac{\partial \mathbb{F}}{\partial d} = \left|\begin{array}{ccccccc}
Mb & 0 & 0 & 0 & 0 & 0 & 0 \\
Mb & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 \\
Mb & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}\right| \qquad \frac{\partial \mathbb{G}}{\partial d} = 0$$

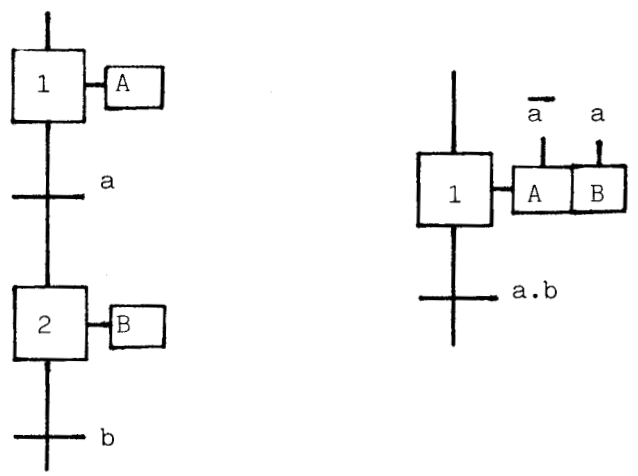
Ce système est inconditionnellement réceptif à d si l'étape 2 est active (y2 = 1).

Il est potentiellement réceptif à d si y1 = 1 (étape 1 active).

Il devient réceptif si en plus de y1 = 1, on a Mb = 1.

Ces résultats sont évidents sur un exemple aussi simple.

- Soit deux parties de grafcet équivalentes dans la mesure où l'état du capteur a est maintenu à la fin de l'action A (Fig 3.2).



- figure 3.2 -

Pour le premier graphe, nous avons:

$$\mathbb{F} = \begin{vmatrix} \bar{a} & 0 & - \\ a & \bar{b} & - \\ - & - & - \end{vmatrix}$$

$$\mathbb{G} = \begin{vmatrix} 1 & 0 & - \\ 0 & 1 & - \\ - & - & - \end{vmatrix} \qquad \text{si} \quad \begin{pmatrix} A \\ B \end{pmatrix} = \mathbb{G} \begin{pmatrix} y1 \\ y2 \\ \vdots \end{pmatrix}$$

$$\frac{\partial \mathbb{F}}{\partial a} = \begin{vmatrix} 1 & 0 & - \\ 1 & 0 & - \\ - & - & - \end{vmatrix} \qquad \frac{\partial \mathbb{G}}{\partial a} = 0$$

Ce système est inconditionnellement réceptif à a si y1 est actif. Il est non sensible par rapport à a.

Pour le deuxième graphe:

$$F = \begin{vmatrix} \overline{ab} & 0 & - \\ 0 & ab & - \\ - & - & - \end{vmatrix}$$

$$G = \begin{vmatrix} \overline{a} & - & - \\ a & - & - \\ - & - & - \end{vmatrix}$$

$$\frac{\partial F}{\partial a} = \begin{vmatrix} b & 0 & - \\ 0 & b & - \\ - & - & - \end{vmatrix}$$

$$\frac{\partial G}{\partial a} = \begin{vmatrix} 1 & - & - \\ 1 & - & - \\ - & - & - \end{vmatrix}$$

Le système est potentiellement réceptif à la variable a si y1 = 1. Il devient réceptif si b = 1. Il est inconditionnellement sensible à a si y1 est actif.

Ce dernier exemple illustre que différents grafcet peuvent représenter le fonctionnement d'une même machine.
Dans le premier graphe, il est clair que toute panne fugitive sur a, en dehors du temps de réalisation de A, est sans effet.
Dans le deuxième grafcet, par contre, une panne fugitive sur a pendant la réalisation de l'action B provoque une anomalie par propagation de l'erreur. Il existe donc des grafcet plus critiques que d'autres vis-à-vis des pannes fugitives.
Nous nous proposons de mettre en place un paramètre qui permette d'estimer la "susceptibilité" d'un grafcet par rapport à une panne fugitive sur une entrée. Ce paramètre permet de choisir un grafcet optimal vis-à-vis de la sécurité.

3.3 ESTIMATION DU COMPORTEMENT EN PRESENCE D'UNE PANNE NON CONSISTANTE

Soit un système observé pendant une durée T. Soit ti $\leqslant$ T les durées cumulées pendant lesquelles le système est réceptif ou sensible à xi durant la période d'observation.
Supposons que pendant cette période T, il existe une faute non consistante de durée tf sur xi.
Si on suppose que les deux événements ne sont pas corrélés, la probabilité pour qu'il existe une date $t \in |0,T|$ où nous ayons à la fois le système réceptif ou sensible à xi et la faute présente est:

$$\mathcal{P}_{f(t)} = \frac{tf}{T} \cdot \frac{ti}{T}$$

$\mathcal{P}_{f(t)}$ représente la probabilité pour qu'une panne non consistante, de durée tf sur xi, soit propagée par la commande.

Remarque:
Un filtrage des entrées élimine les fautes fugitives, contenues dans les comptes rendus, qui ont une durée faible. Dans les automates programmables, ce filtrage est de quelques ms.
Pour diminuer les risques de propagation de ce genre d'erreurs, il faut agir sur le rapport $\frac{ti}{T}$. Le meilleur grafcet, selon ce critère, est donc celui qui minimise ce rapport.
S'il est théoriquement possible de déterminer ce rapport $\frac{ti}{T}$, cela est pratiquement irréaliste dans la phase de conception de l'automatisme. Nous proposons un paramètre d'estimation. Il permet de comparer les comportements de différents grafcets satisfaisant un même cahier des charges, par rapport aux fautes non consistantes sur une entrée.

3.3.1 ETAT INTERNE ATTEIGNABLE

L'état interne d'une machine séquentielle peut être représenté par l'état d'activation du grafcet. En fonctionnement normal (sans défaillance), il existe un certain nombre d'états internes qu'il est possible d'atteindre à partir de l'état initial du grafcet. Nous supposons connue la liste de ces états atteignables.

Ces états dépendent bien sûr de la structure du graphe, mais aussi des exclusions tant technologiques que logiques, au niveau des événements.

Pour établir cette liste d'états atteignables, nous ne faisons aucune hypothèse sur les temps de réalisation, ni sur les occurences possibles d'événements.

Il est clair dans ces conditions que certains états atteignables ne seront jamais atteints en fonctionnement.

3.2 CONVENTIONS PERMETTANT DE DEFINIR UN ESTIMATEUR

Cet estimateur devrait, idéalement, être une mesure de $\frac{ti}{T}$ puisque en minimisant ce rapport, nous réduisons la probabilité de propagation de l'erreur consécutive à une panne non consistante sur xi.

En fait, cette approche est utopique. Les problèmes à résoudre sont essentiellement les suivants:

- Définir un cycle pour en connaître sa durée et les états réellement atteints. Ceci n'est possible qu'avec des machines à cycle fixe.
- Les variables de commande, qui évoluent en principe de façon aléatoire par rapport au cycle en cours, interviennent dans les réceptivités et les sensibilités. Ces grandeurs modifient la valeur de ti de façon imprévisible puisqu'elles sont indépendantes de la partie opérative.

Nous adoptons donc les deux conventions suivantes:

- Proposition 1:

Nous considérons un cycle fictif tel que tout état atteignable est atteint une fois et une seule.

- Proposition 2:

Nous admettons que tous les états atteignables, non inconditionnellement instables, ont une même durée

- Proposition 3:

Si le système est potentiellement réceptif (resp. sensible) à xi pour un état donné, nous le considérons comme inconditionnellement réceptif (resp. sensible) à cette variable.

3.3.3 ESTIMATION DE LA RECEPTIVITE ET DE LA SENSIBILITE PAR RAPPORT A UNE ENTREE

Supposons donc la période d'observation formée des n états internes atteignables non inconditionnellement instables. Tous ces états sont supposés avoir une durée identique τ . Dans ces conditions T s'exprime par:

$$T = n\tau \qquad |3\text{-}12|$$

Soit m le nombre d'états internes pour lequel le système est potentiellement réceptif à xi. Dans ces conditions:

$$ti = m\tau \qquad |3\text{-}13|$$

Nous estimons la réceptivité à xi sur le cycle de référence par:

$$\boxed{R(xi) = \frac{m}{n}} \qquad |3\text{-}14|$$

De même, si h est le nombre d'états internes pour lequel le système est potentiellement réceptif, nous estimons la sensibilité par:

$$\boxed{S(xi) = \frac{h}{n}} \qquad |3\text{-}15|$$

Remarque:
Si le système est potentiellement sensible à xi pour tout état interne, c'est-à-dire, si $\frac{\partial\, ff}{\partial\, xi}$ est non nul pour au moins une combinaison de valeurs des entrées autres que xi, alors h = n. Dans ce cas, S(xi) = 1.

3.3.4 ESTIMATEUR DE LA SUSCEPTIBILITE PAR RAPPORT A UNE ENTREE

Nous définissons le facteur de susceptibilité par:

$$C(xi) = \frac{ti}{T} \qquad |3\text{-}16|$$

Les valeurs de ti et T étant définies par application de nos conventions de départ.

Si nous avons m états internes pour lesquels le système est potentiellement réceptif, et h états internes pour lesquels il est potentiellement sensible, il est possible aussi qu'il existe k états internes pour lesquels le système est, à la fois, potentiellement réceptif et sensible.
Dans ces conditions, ti est estimé par:

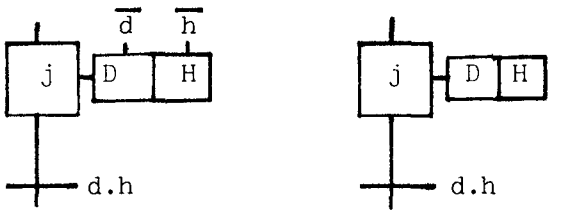
$$ti = (m + h - k)$$

La relation |3-16|, compte tenu de |3-12|, devient:

$$C(xi) = \frac{m + h - k}{n} \qquad |3\text{-}17|$$

Remarque:
L'estimateur global ainsi défini peut s'avérer optimiste à cause de nos conventions. En effet, si le système est potentiellement réceptif et sensible, il est considéré comme inconditionnellement réceptif et sensible. Cette hypothèse justifie |3-17|. Dans la pratique, il est possible que le système soit plus probablement réceptif que sensible (ou vice versa).

Exemple:
Soit l'élément de grafcet ci-dessous (Fig 3.3):



- a - - b -

- figure 3.3 -

Pour tout état interne contenant l'étape j active, le système défini Fig 3.3a est potentiellement réceptif et sensible à d. Par contre, le système est réellement réceptif inconditionnellement à d. En cas de défaillance de d, la propagation de l'erreur est plus probable par l'effet de la sensibilité que par la réceptivité.

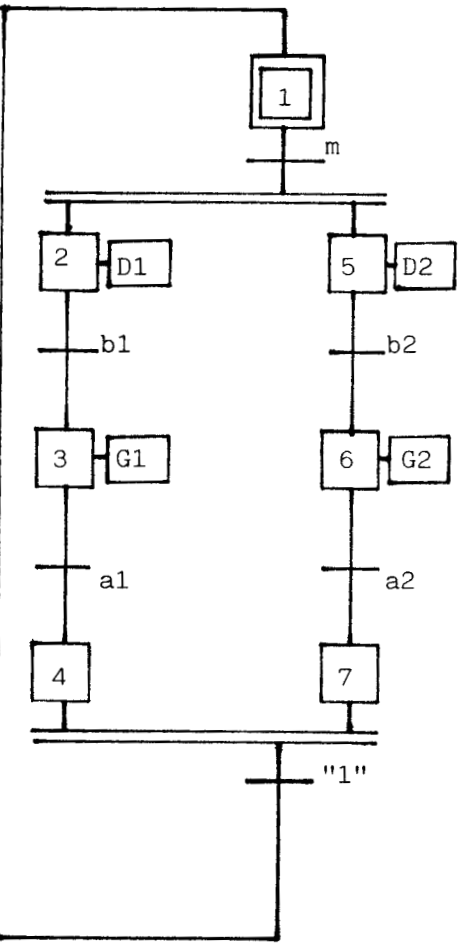
Il ressort de cet exemple que l'estimateur R(xi) + S(xi) peut indiquer une susceptibilité plus importante que ne le laisse supposer C(xi), lorsque C(xi) $<$ R(xi) + S(xi).
Dans le cas de la figure, la susceptibilité C(d) est la même pour les deux graphes, alors que la somme R(d) + S(d) diffère.
Dans la réalité, la représentation a est plus critique que b.
Il va de soi que ces deux représentations sont fonctionnellement identiques, uniquement si les actions D et H peuvent être maintenues (cas de deux vérins en butée, par exemple).

Remarque:
Il n'est pas inutile de rappeler que C(xi) est un estimateur qui permet de guider le concepteur dans le choix d'un grafcet.
Il serait hasardeux de considérer ce coefficient comme une mesure de la probabilité de propagation de l'erreur.

3.3.5 EXEMPLES DE CALCUL

Attention: La norme grafcet impose la notation Xi pour l'étape i. L'état interne s'exprime alors par Xk au lieu de Yk. Il ne saurait y avoir de confusion entre l'entrée j notée xj et l'étape j notée Xj.

Soit deux chariots qui effectuent un aller/ retour sur leur voie propre, chaque fois qu'un opérateur agit sur un bouton poussoir.
Le cahier des charges peut se représenter de différentes façons.
Un exemple est donné par les figures 3.4 a, b, c.
Intuitivement, on se rend compte que le comportement vis-à-vis d'une panne non consistante sur m doit être le même pour les trois graphes de l'exemple.



- figure 3.4a -

$$\mathbb{F} = \begin{vmatrix} \overrightarrow{m} & 0 & 0 & X7 & 0 & 0 & 0 \\ m & \overrightarrow{b1} & 0 & 0 & 0 & 0 & 0 \\ 0 & b1 & \overrightarrow{c1} & 0 & 0 & 0 & 0 \\ 0 & 0 & a1 & \overrightarrow{X7} & 0 & 0 & 0 \\ m & 0 & 0 & 0 & \overrightarrow{b2} & 0 & 0 \\ 0 & 0 & 0 & 0 & b2 & \overrightarrow{a2} & 0 \\ 0 & 0 & 0 & 0 & 0 & a2 & \overrightarrow{X4} \end{vmatrix}$$

En posant $\begin{vmatrix} D1 \\ G1 \\ D2 \\ G2 \end{vmatrix} = \mathbb{G} \cdot \mathbb{X}$

$$\mathbb{G} = \begin{vmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{vmatrix}$$

$$\mathbb{H} = |0|$$

Exprimons les dérivées par rapport à m:

$$\frac{\partial \mathbb{F}}{\partial m} = \begin{vmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & & & & & & \\ 0 & & & & & & \\ 0 & & & 0 & & & \\ 1 & & & & & & \\ 0 & & & & & & \\ 0 & & & & & & \end{vmatrix}$$

Les situations non inconditionnellemnt instables sont:

1; 2,5; 2,6; 2,7; 3,5; 3,6; 3,7; 4,5; 4,6. (4,7 est éliminé).

Soit 9 états internes à considérer.

Seuls les états internes pour lesquels X1 est actif sont à retenir.

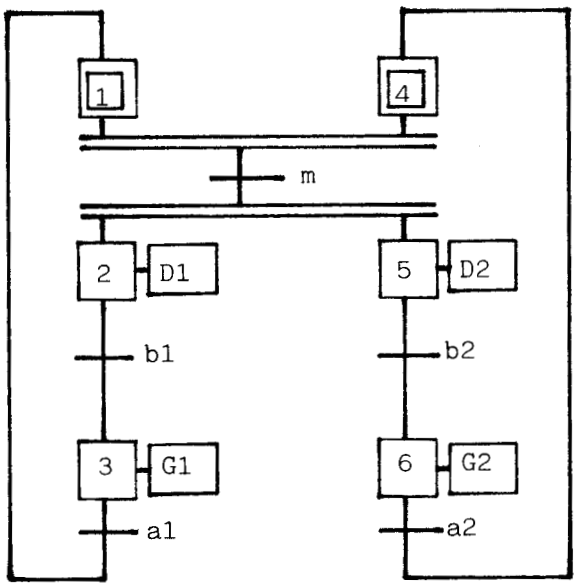
Donc: R(m) = 1/9

S(m) = 0

$$\boxed{C(m) = 1/9}$$

Par contre, pour a, on trouve:

$$\boxed{C(a1) = \frac{3}{9} = \frac{1}{3}}$$



$$F = \begin{vmatrix} \overline{mX4} & \overline{0} & a1 & 0 & 0 & 0 \\ mX4 & \overline{b1} & 0 & 0 & 0 & 0 \\ 0 & b1 & \overline{a1} & 0 & 0 & 0 \\ 0 & 0 & 0 & \overline{mX1} & 0 & a2 \\ mX4 & 0 & 0 & 0 & \overline{b2} & 0 \\ 0 & 0 & 0 & 0 & b2 & \overline{a2} \end{vmatrix}$$

$$G = \begin{vmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix}$$

$$H = |0|$$

Exprimons les dérivées par rapport à m:

$$\frac{\partial G}{\partial m} = 0 \quad ; \quad \frac{\partial F}{\partial m} = \begin{vmatrix} X4 & 0 & 0 & 0 & 0 & 0 \\ X4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & X1 & 0 & 0 \\ X4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

- figure 3.4b -

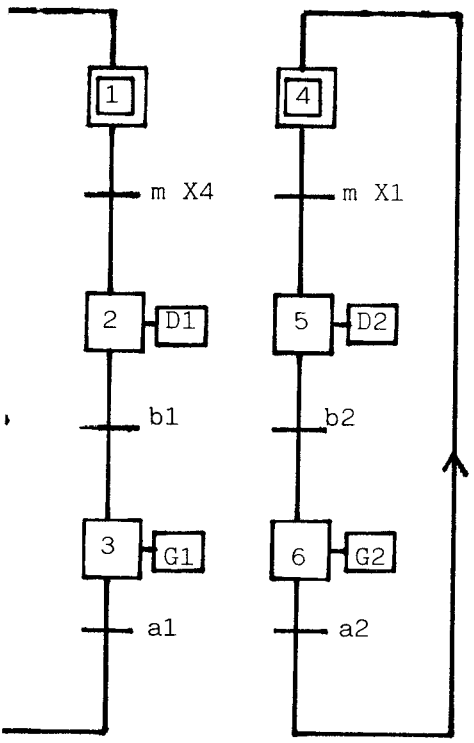
Les situations non inconditionnellement instables sont:
1,4; 1,5; 1,6; 2,5; 2,6; 3,5; 3,6; 4,2; 4,3.
Soit 9 états internes. Seule la solution 1,4 est à retenir.
Exemple sur 1,5: X4 = 0; X1 = X5 = 1

$$\left(\frac{\partial F}{\partial m}\right)(Y) = 0$$

$$\left|\begin{array}{cccccc||c} 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}\right| = 0$$

$$\boxed{C(m) = \frac{1}{9}} \qquad \boxed{C(a1) = \frac{1}{3}}$$

La valeur de C(a1) est évaluée par un calcul analogue.



$$\mathcal{F} = \begin{vmatrix} \overline{mX4} & 0 & a1 & 0 & 0 & 0 \\ mX4 & \overline{b1} & 0 & 0 & 0 & 0 \\ 0 & b1 & \overline{a1} & 0 & 0 & 0 \\ 0 & 0 & 0 & \overline{mX1} & 0 & a2 \\ 0 & 0 & 0 & mX1 & \overline{b2} & 0 \\ 0 & 0 & 0 & 0 & b2 & \overline{a2} \end{vmatrix}$$

$$\mathcal{G} = \begin{vmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{vmatrix}$$

$$\mathcal{H} = |0|$$

Exprimons les dérivées par rapport à m:

$$\frac{\partial \mathcal{F}}{\partial m} = \begin{vmatrix} X4 & 0 & 0 & 0 & 0 & 0 \\ X4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & X1 & 0 & 0 \\ 0 & 0 & 0 & X1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{vmatrix} \qquad \frac{\partial \mathcal{G}}{\partial m} = 0$$

- figure 3.4c -

Les situations non inconditionnellement instables sont:
1,4; 1,5; 1,6; 2,4; 2,5; 2,6; 3,4; 3,5; 3,6.
Les situations sont les mêmes que précédemment.
Seule la solution 1,4 est ici à retenir pour le calcul de C(m).

Nous trouvons:

$$C(m) = \frac{1}{9} \qquad C(a1) = \frac{1}{3}$$

Nous retrouvons bien ici l'identité de comportement de ces trois grafcetsvis-à-vis des fautes non consistantes sur m et a1.
(On trouve C(a1) = C(a2) = C(b1) = C(b2)).

La comparaison entre C(m) et C(a1) met simplement en évidence que le nombre d'états internes atteignables pour lequel le système est réceptif à a1 est supérieur à celui pour lequel il est réceptif à m.

Dans la pratique, la performance dépend évidemment de la vitesse d'exécution des actions mais aussi du temps d'attente imposé par l'opérateur. Ce dernier point illustre bien le caractère aléatoire de m par rapport à la machine. Dans ce cas, l'évaluation de C(m) nécessite une étude sur un grand nombre de cycles. Ceci relève de la statistique.

3.3.6 RECHERCHE DES ETATS INTERNES ATTEIGNABLES

Dans les cas simples, la liste des états internes atteignables est accessible. Il n'en est pas de même lorsque le grafcet est important et qu'il y a beaucoup de parallélisme.
Nous proposons alors de définir la liste des états internes structurellement atteignables qui peuvent être atteints à partir de l'état initial par une séquence de tir de transitions non interprétées. Cette liste qui ne tient pas compte des exclusions liées à l'interprétation, contient la liste des états atteignables.

Proposition:
La liste des états structurellement atteignables, relative à un grafcet de taille finie, est elle-même de dimension finie.
En effet, le grafcet, en tant que graphe booléen, peut présenter au plus, 2^n états, s'il contient n étapes.

Convention:
Dans la pratique, les divergences en OU sont, le plus souvent, rendues exclusives par l'interprétation. Nous considérons alors ces noeuds OU comme exclusifs. Dans le cas contraire, une transformation préalable du graphe est indispensable.
La différence entre les listes des états atteignables d'une part, et des états structurellement atteignables d'autre part, se réduit aux états inconditionnellement instables ou, dont la condition de stabilité est liée uniquement à l'état d'une ou plusieurs étapes.

Elaboration de la liste des états structurellement atteignables:
Comme pour les réseaux de Pétri, il est possible de définir la structure du grafcet par des matrices d'incidence avant et arrière |DAC-76| |THE- |.

Les produits et sommes sont ici remplacés par les opérateurs ET et OU.

La matrice d'incidence avant permet de donner la liste des transitions structurellement franchissables pour un état donné. Ces transitions sont tirées une à une, conduisant chaque fois à un nouvel état interne. La nouvelle valeur obtenue par désactivation des étapes antécédentes, puis activation des étapes subséquentes, est mise dans la liste des états structurellement accessibles, s'il n'y est pas déjà. L'algorithme est repris pour tous les états nouvellement rencontrés. Il se termine,lorsque toutes les transitions tirables à partir du dernier état rencontré, conduisent à des états atteignables connus. La liste ainsi obtenue est alors "filtrée" à partir d'une expression particulière de la matrice .

Chaque fonction fij de $\mathbb{F}$ est ainsi traitée:

-si fij est identiquement égale à 1, elle conserve sa valeur;

- si elle contient des termes réduits, dépendant uniquement de l'activation d'étapes, elle est réduite à la somme logique de ces termes réduits;
- elle est rendue nulle dans tous les autres cas.

A partir de la matrice $\mathbb{F}$ ainsi obtenue, nous effectuons l'opération $\mathbb{Y}(k+1) = \mathbb{F} \cdot \mathbb{Y}(k)$ pour chaque valeur $\mathbb{Y}(k)$ correspondant à la liste des états structurellement atteignables.

Si $\mathbb{Y}(k+1) = \mathbb{Y}(k)$, l'état est conditionnellement stable et il est conservé.

Si $\mathbb{Y}(k+1) \neq \mathbb{Y}(k)$, l'état qui est alors instable quelque soit les entrées, est éliminé.

La liste ainsi filtrée est celle que nous recherchions. Il est certain que le temps d'exécution de cet algorithme peut être très long.

Nous allons tenter de mettre en évidence des principes qui permettraient d'obtenir "naturellement" des grafcets moins susceptibles aux fautes non consistantes.

3.4 MISE EN EVIDENCE DES PRINCIPES QUI REDUISENT LA SUSCEPTIBILITE

Dans le paragraphe précédent, nous avons établi le principe d'un coefficient estimateur de la susceptibilité d'un grafcet vis-à-vis des pannes non consistantes sur une entrée. Nous allons maintenant mettre en évidence un certain nombre de règles dont l'application permet d'obtenir naturellement un grafcet proche de l'optimal pour ce critère. Ces règles sont établies à partir d'exemples.

3.4.1 FAUT-IL PRIVILEGIER L'ASPECT RECEPTIF OU L'ASPECT SENSIBLE ?

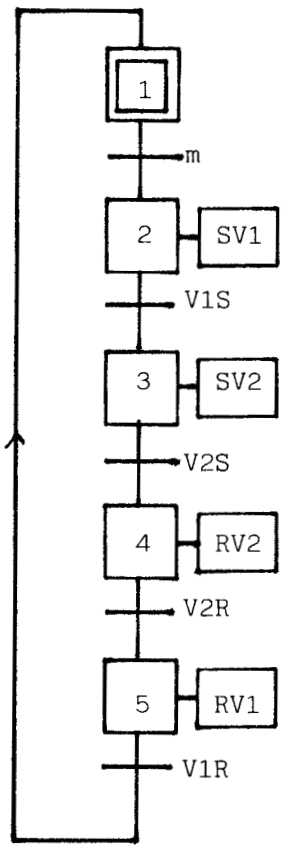
Il est parfois dit que la réceptivité à une faute fugitive est plus gênante que la sensibilité à cette même anomalie. L'idée est qu'à la disparition de l'erreur, en cas de sensibilité, le système reprend l'état normal, alors qu'en cas de réceptivité, la modification de l'état interne est irréversible.
Cette affirmation est parfaitement juste si on admet que la modification d'une sortie pendant un court instant est admissible, parce que filtrée par les actionneurs. En fait, un filtrage des entrées en rapport avec la vitesse d'évolution de la machine aurait dû, dans ce cas, interdire l'exploitation de la faute fugitive. Pour notre part, nous considérons que toute variation erronée d'une sortie, quelqu'en soit la durée, est contraire à la sécurité de l'ensemble. Il reste toutefois au concepteur à juger, pour chaque application, des risques encourus.

Exemples: Ouvrir une vanne de vidange à l'égout d'une cuve contenant un produit toxique donne une pollution moins importante si le temps d'ouverture de la vanne est faible!

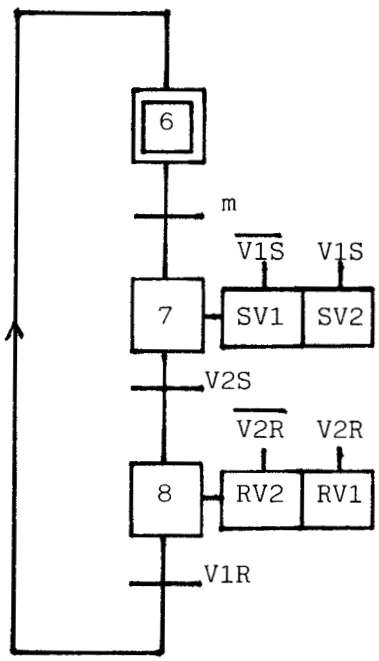
Desserrer le bridage d'une pièce pendant son usinage provoque une catastrophe, même si ce débridage est de courte durée.

L'introduction des spécifications technologiques permet souvent au concepteur de diminuer le nombre d'étapes en rendant le système sensible plutôt que réceptif à certaines variables qui sont maintenues.
Si nous prenons le traditionnel cycle en L de deux vérins, le grafcet qui spécifie le fonctionnement est donné Fig(3.5).

Nous notons SVi (resp. RVi) les actions sortir (resp. rentrer) le vérin i et ViS (resp. ViR) le compte rendu vérin i rentré (resp. sorti).



- a -



- b -

- figure 3.5 -

Nous savons que deux états internes suffisent au cycle en L. Le graphe figure 3.5b donne le même fonctionnement que celui de la figure 3.5a. Toutefois, le premier graphe donne:

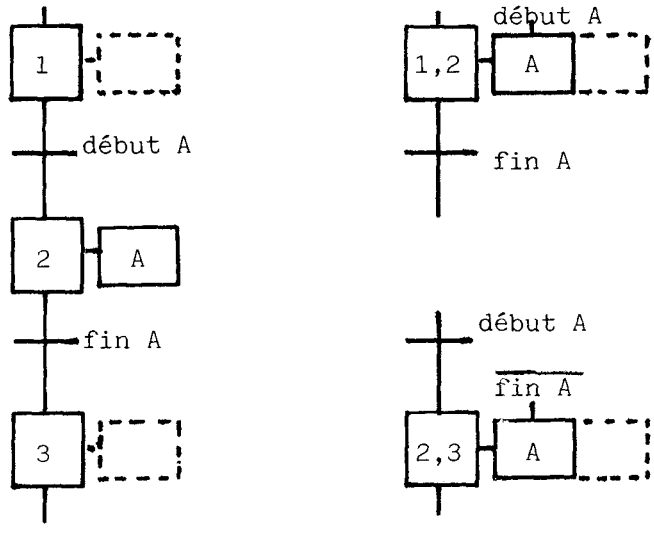
$$C(V1S) = C(V2S) = C(V2R) = C(V1R) = \frac{1}{5}$$

alors que le deuxième graphe donne la valeur $\frac{1}{3}$.

Notre estimateur laisse prévoir une susceptibilité plus forte de ce deuxième graphe vis-à-vis des défauts sur les capteurs.

Ce phénomène est bien vérifié puisque, avec ce groupement d'étapes, une modification de V1S perturbe la sortie de V2 et que V2S influe sur la sortie de V1. Le même phénomène se produit au retour avec V1R et V2R.
En fait, deux transformations génériques sont utilisées pour ce genre de fusionnement.

Soit une action A qui est activée par le compte rendu "début de A" et terminée par modification de "fin de A"(fig 3.6a).

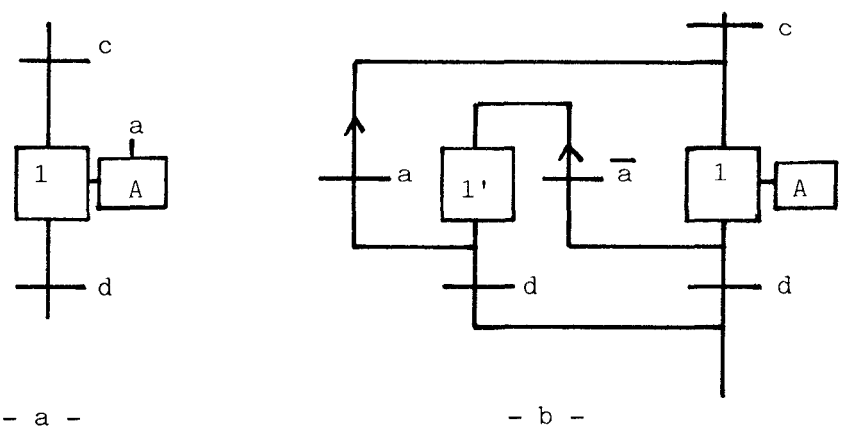


- figure 3.6 -

Si "début de A" est maintenu pendant l'action A, on peut adopter la transformation 1 (fig 3.6a). Par contre, si fin de A est maintenu après la fin de l'action, on peut adopter la transformation 2 . Pratiquement ces deux transformations sont appliquées simultanément.
Dans notre exemple, le fusionnement des étapes 2,3 et 4,5 (fig 3.5a) donne respectivement les étapes 7 et 8 (fig 3.5b).
Nous pouvons généraliser ce résultat.
Si les étapes 1 et 2 de la figure 3.6a sont fusionnables, c'est qu'elles interviennent l'une et l'autre dans x situations atteignables. Après fusionnement, si rien n'est changé par ailleurs, l'étape fusionné 1,2 (resp. 1,3) se trouvera également dans x situations atteignables.

Le fusionnement 1,2 (ou 2 ,3) fait donc passer le nombre de situations atteignables de n à n-x.
Le fusionnement 1,2 diminue la réceptivité qui passe de m à m-x mais, la sensibilité évolue de h à h+x. Donc, C passe de $\frac{m+h}{n}$ à $\frac{m-x+h+x}{n-x}$, la susceptibilité est augmentée.

Remarques:
Comme nous avons pu l'observer dans l'exemple précédent, l'estimateur va diminuer pour toutes les variables. Cela est dû au fait que la période d'observation T est divisée en un plus petit nombre de situations. Chaque état interne atteignable se voit donc attribuer artificiellement une durée plus longue.
Dans la pratique, il est bien évident que le comportement du système est plus critique uniquement pour la variable concernée par le fusionnement.
Ayant remarqué que la transformation qui rend un système plus sensible et moins réceptif est néfaste, on peut alors se poser la question de la représentation du combinatoire local (fig 3.7).



- figure 3.7 -

Une étude locale montre, comme cela est prévisible, que l'on a dans le premier cas: C(a) = S(a) = 1
C(d) = R(d) = 1

Dans le deuxième cas, on retrouve:

C(a) = R(a) = 1

C(d) = R(d) = 1

La première solution est évidemment la meilleure dans ce cas, puisque plus lisible pour une même performance.

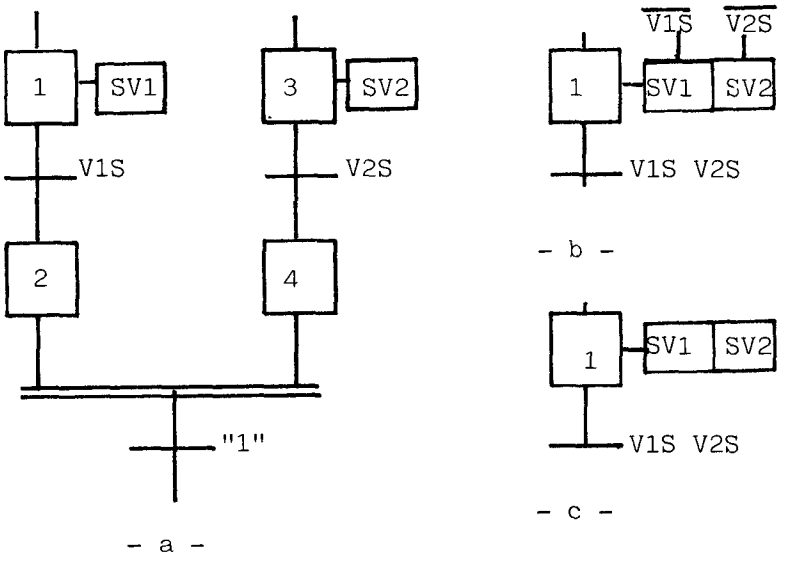
Il est indéniable que le combinatoire local doit apparaître sous forme de sensibilité.

Par contre, nous pouvons tirer de notre étude les règles suivantes:

Règle 1

Toute exploitation des spécifications technologiques visant à réduire le nombre d'étapes en rendant le système moins réceptif mais plus sensible, est néfaste. Ceci conduit à des grafcet plus susceptibles vis-à-vis des pannes non consistantes.

Remarque: Cette règle illustrée par l'exemple de la figure 3.5 est confirmée dans le cas de la transformation présentée dans les figures 3.8a, 3.8b. En effet, l'estimateur de suceptibilité passe de C(V1S)=2/3 (fig. 3.8a) à C(V1S)=1 (fig. 3.8b). Par contre si SV1 (resp. SV2) peut être maitenu lorsque V1S (resp. V2S)est atteint, le grafcet de la figure 3.c est acceptable. La valeur de l'estimateur calculée dans ce cas ne doit toutefois pas être comparée à celles calculées pour les deux grafcets précédents. En effet, le cahier des charges n' y est pas rééllement équivalent.



- figure 3.8 -

Règle 2

La sensibilité doit être utilisée uniquement pour introduire des conditions d'inhibitions locales qui sont fonctionnellement combinatoires.

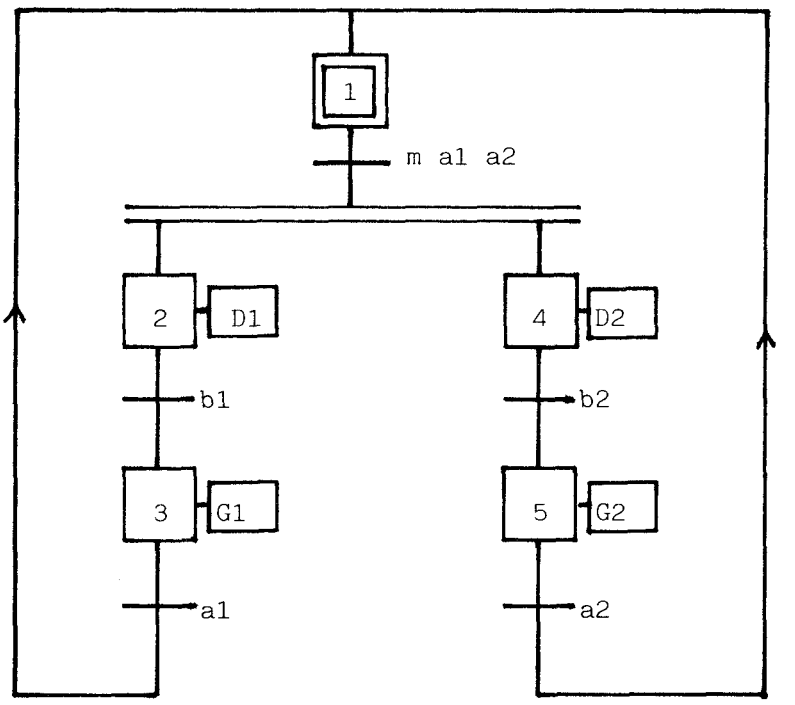
Exemple: Pour usiner une pièce, il faut que celle-ci reste bridée pendant toute la phase d'usinage.

La sécurité ne peut être garantie que par la présence d'un capteur qui détecte la permanence du bridage (par exemple un pressostat). Les actions d'usinage seront alors localement validées, en rendant le grafcet sensible à la variable représentée par ce capteur.

.4.2 CHOIX DES EVENEMENTS ET SIMPLIFICATIONS STRUCTURELLES

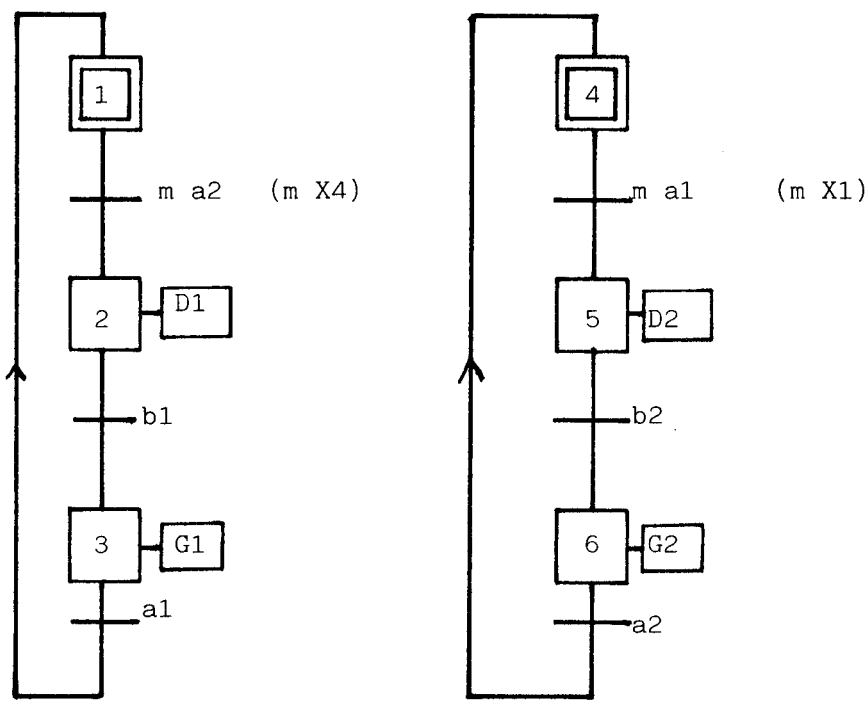
Certaines transformations de grafcet sont envisageables également en cas d'actions simultanées au niveau des étapes d'attente qui paraissent souvent inutiles. Voyons ce qu'il en est sur l'exemple de nos deux chariots.

En remarquant qu'à la fin de l'aller-retour du chariot i, le capteur ai reste à 1, on peut représenter le cahier des charges par le grafcet suivant (Fig 3.9):



- figure 3.9 -

On trouve C(a1) = R(a1) = $\frac{7}{9}$ au lieu de $\frac{1}{3}$ avec le grafcet initial. Cette dégradation ne provient pas uniquement de la modification de structure mais surtout, du choix de l'événement m a1 a2 pour la première transition. En effet, si nous remplaçons m a1 a2 par m $\overline{X2}$ $\overline{X3}$ $\overline{X4}$ $\overline{X5}$,le cahier des charges est respecté et nous retrouvons C(a1) = $\frac{1}{3}$ avec le même structure.
Ce phénomène est encore plus net si nous comparons le graphe de la figure 3.10 obtenu à partir de celui de la figure 3.4c, en remplaçant mX4 et mX1 respectivement par ma2 et ma1.



- figure 3.10 -

L'estimateur de susceptibilité passe de C(a1) = $\frac{1}{3}$ pour le graphe initial à C(a1) = $\frac{2}{3}$.
Pour limiter l'influence des pannes non consistantes sur une entrée, il faut limiter le temps pendant lequel les système est réceptif ou sensible à celle-ci. Ceci nous conduit à énoncer la règle suivante.

Règle 3

Toute modification attendue d'une entrée constituant un événement normal, doit être prise en compte dans une réceptivité (donc privilégier la réceptivité).

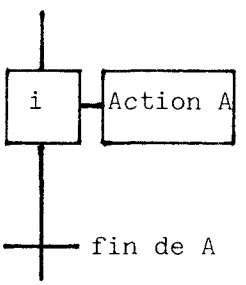
Dans la suite de l'évolution, il est préférable d'utiliser ce changement d'état interne plutôt que de reprendre l'état de la variable.

Remarque:

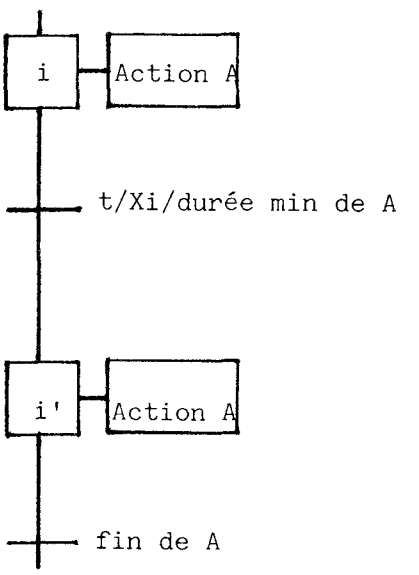
Il est d'usage courant de vérifier l'état de la partie opérative au niveau des transitions qui permettent de quitter l'état initial du grafcet. Cette pratique est acceptable, malgré l'augmentation de susceptibilité qu'elle entraîne, si toute panne, agissant à partir de cet état secondaire, correspond à un fonctionnement de sécurité. Cette condition est généralement satisfaite.

Remarque:

L'utilisation de temporisations choisies en fonction de la durée minimale prévisible des actions permet de réduire la susceptibilité de la commande. La figure 3.11 illustre cette proposition.



- a - grafcet initial



- b - grafcet modifié

- figure 3.11 -

D'autres solutions du même ordre peuvent être avancées. En fait, l'introduction des temporisations conduit à inclure dans la commande des informations relatives au comportement de la P.O. en l'absence de défaillances.

Il ressort de ce chapitre que toute démarche visant à diminuer le nombre d'étapes en augmentant le combinatoire, tant au niveau réceptivité qu'au niveau sensibilité est néfaste.

Nous avons envisagé un mode de défaillance particulier de l'automatisme correspondant à l'acquisition d'un compte rendu erroné, par la partie commande. Nous avons vu que l'influence d'une telle défaillance peut être limitée par une description judicieuse du cahier des charges, si cette anomalie est non consistante.
Il est certain qu'il existe d'autres modes de défaillances dont l'influence en peut être limitée par un choix judicieux de grafcet.
Dans le chapitre suivant, nous faisons une étude critique de solutions couramment proposées pour améliorer la sûreté de fonctionnement des systèmes commandés par des A.P.I. en présence d'une panne consistante.

CHAPITRE IV

IMPACT DES REDONDANCES MATERIELLES SUR LA SURETE DES AUTOMATISMES

L'obtention du niveau de sûreté défini par le cahier des charges passe par une analyse des modes de défaillance et une évaluation des différents estimateurs rappelés au chapitre premier.
Dans le cadre des automatismes réparables utilisés en production, toute panne consistante entraîne une intervention du service de maintenance. Il est clair que dans ces conditions,la disponibilité d'un système réparable est fortement tributaire de l'efficacité de la maintenance |LAP-75|. Comme l'illustre l'exemple traité au chapitre premier, la disponibilité asymptotique tend rapidement vers 1. L'augmentation de la disponibilité passe donc par une diminution des temps moyens de réparation qui justifie l'emploi des mécanismes de test (en ligne ou hors ligne) permettant la localisation de la défaillance.
Compte tenu des puissances mises en jeu dans les automatismes et des coûts des parties opératives commandées, l'amélioration de la sécurité est souvent une nécessité. Le niveau de sécurité requis peut être obtenu dans certains cas, par une approche non tolérante aux fautes. Souvent, une telle démarche s'avère technologiquement impossible ou trop coûteuse. Une modification d'architecture permettant de garantir la sécurité est alors indispensable.

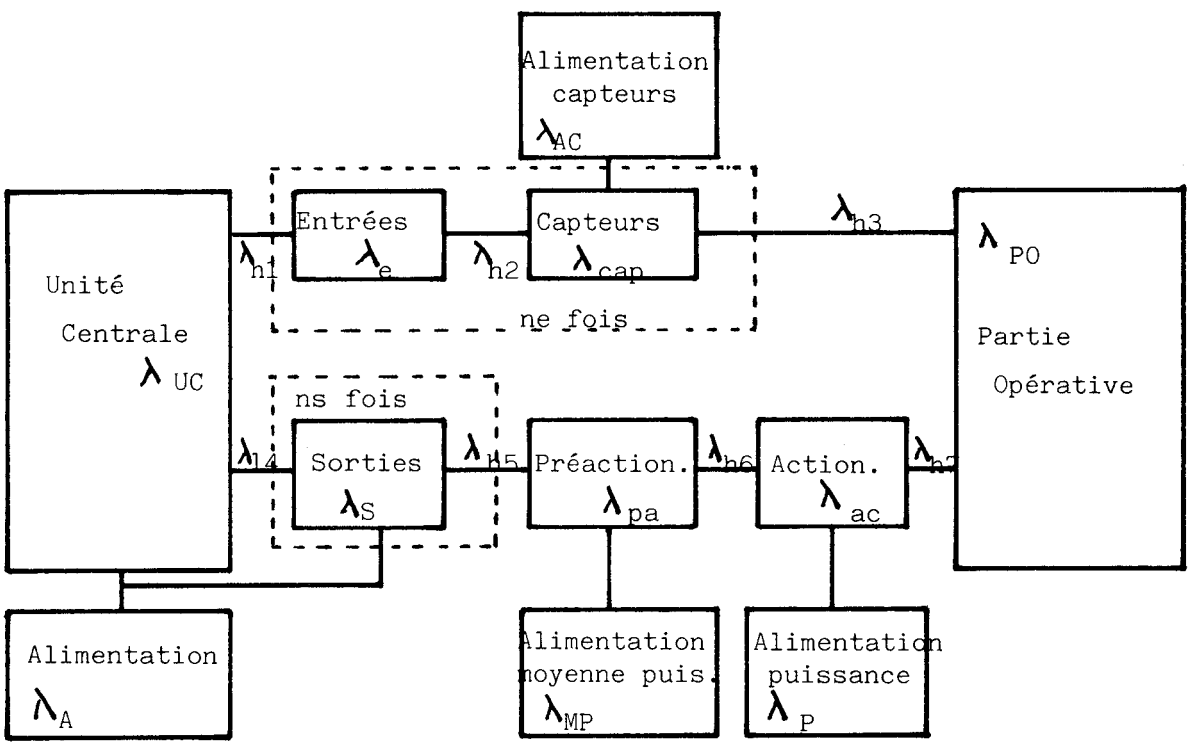
Dans le premier paragraphe, nous présentons l'architecture minimale d'un automatisme correspondant à une approche non tolérante aux fautes. Nous mettons en évidence les causes principales de non sécurité liées au matériel, en utilisant des taux de panne fournis par les constructeurs.

Dans le deuxième paragraphe, nous nous interessons à l'automate réalisant la partie commande de l'automatisme. Nous évaluons la sûreté obtenue pour différentes architectures couramment proposées.

Le dernier paragraphe porte sur les éléments hors automate.

4.1 ETUDE SUCCINTE DES CAUSES DE DEFAILLANCE

La figure (4.1) représente schématiquement un automatisme. En général, chacun des éléments de cette chaîne représente, pour le concepteur, des limites de fournitures. Son rôle consiste, d'une part à choisir les composants en tenant compte de leur fiabilité, d'autre part à trouver des assemblages permettant d'atteindre la sûreté attendue.



- figure 4.1 -

Les liaisons h3, h7 sont mécaniques alors que h6 est électrique ou pneumatique. h2 et h5 sont des liaisons électriques de moyenne puissance; h4 et h1 sont des bus internes à l'automate ou, de plus en plus, des bus supportés par une liaison série dans le cadre des architectures distribuées.
En supposant tous les éléments en série pour la fiabilité, nous pouvons admettre que le taux de panne de l'automatisme (Fig 4.1) est donné par la relation |4-1|. Nous supposons que les défaillances entre les divers éléments sont non corrélées et exponentiellement distribuées.

$$\lambda = \underbrace{\lambda_{uc}}_{\text{unité centrale}} + \underbrace{\lambda_{h1} + \lambda_{h4}}_{\text{bus E/S supposé indépendant du nbre d'E/S}} + \underbrace{ne\,\lambda_e + ne\,\lambda_{h2} + ne\,\lambda_{cap} + ne\,\lambda_{h3}}_{\text{fiabilité des entrées}}$$

$$+\underbrace{ns\,\lambda_s + ns\,\lambda_{h5} + npa\,\lambda_{pa} + nac\,\lambda_{h5} + nac\,\lambda_{ac} + nac\,\lambda_{h7}}_{\text{fiabilité des organes de sorties}} + \lambda_{PO}$$

$$+ \underbrace{\lambda_{AUC} + \lambda_{AC} + \lambda_{MP} + \lambda_P}_{\text{fiabilité des alimentations}} \qquad |4\text{-}1|$$

avec:

ne = nombre d'entrées de l'unité de traitement

ns = nombre de sorties de l'unité de traitement

npa = nombre de préactionneurs

nac = nombre d'actionneurs

Pour évaluer la sûreté, il faut connaître les valeurs des différents taux de pannes.

Pour la partie commande, nous utilisons les valeurs fournies par Merlin Gerin |CLA-84| concernant les automates programmables de la série PB 400 (Fig 4.2).

CARTE	λ (10^{-6} panne/h)	λ nc(10^{-6} panne/h)	Pc (%)
16 E 24 Vc	37.8	26.5	29.7
16 E 110 ou 220V	41.3	29.1	29.5
16 E 24/48 V ACDC	41.9	30.0	28.4
16 S Relais	54.0	26.6	50.7
16 S 24/48V 0.5A	45.9	26.4	42.5
8 S = 2A	37.8	23.4	38.2
8 S 110/220 V	39.9	25.4	36.3
12 E analogique	78.2	14.85	81.0
UC 4K RAM	124.7	64.5	48
UC 4K EEPROM	120.7	36.4	69.8
CARTE SURVEILLANCE	30.7	1.5	95
Alimentation	80	1	98.8

- fig. 4.2 -

λ nc représente le taux de pannes non couvertes par des mécanismes internes de détection implantés dans cet automate.

Remarque:
Le constructeur signale que les valeurs données dans ce tableau sont des taux de pannes théoriques obtenus par un calcul simplifié.*
L'expérience montre que le rapport λ calculé / λ mesuré en exploitation, varie de 4 à 10 selon les cartes considérées.
Pour une configuration moyenne comprenant 10 cartes d'entrées et 10 cartes de sorties, on trouve $\lambda a = 1227.10^{-6}$ pannes / heure dont 419.10^{-6} pannes / heure pour les 10 cartes d'entrées et 540.10^{-6} pannes/ heure pour les 10 cartes de sorties.
Pour une telle configuration de 160 entrées / 160 sorties, 76% des pannes sont à attribuer aux dispositifs E/S.

Pour évaluer les performances des capteurs, distributeurs et vérins, nous avons adopté les valeurs fournies par la Société Lecq France qui fabrique ces composants.
Pour des capteurs de type fin de course, la fiabilité est évaluée à partir du nombre de manoeuvres. $3 * 10^{6}$ manoeuvres représente une valeur moyenne pour ce type de composants. Ce qui donne un taux de pannes $\lambda = 0,3\ 10^{-6} * f$ pannes par heure.
si f représente le nombre de manoeuvres par heure.
Pour les distributeurs pneumatiques, les taux de pannes sont de l'ordre de $\lambda = 0,4\ 10^{-6} * f$ pannes par heure.
Pour un vérin, le constructeur donne une course de 2000 km avant défaillance; ce qui donne $\lambda v = 0,5\ 10^{-6} * h * f$ panne/heure.
h est la course en mètre du vérin et f le nombre de déplacements par heure.
Pour un vérin ayant une course de 0,2 m par exemple, λ v est égal à $0,1 * 10^{-6} * f$
Pour un automatisme, utilisant l'automate précédent, ayant un contact sur chaque entrée et un vérin associé à un distributeur pour 2 sorties (hypothèse de préactionneurs double pilotage), nous obtenons pour les capteurs, préactionneurs et actionneurs un taux de pannes $\lambda \neq (160 * 0,3 * 10^{-6} + 80 * 0,6 * 10^{-6})$ f.
Dans cette expression, f représente le nombre moyen de sollicitations par heure de chaque composant.

* NORME MIL-HDBA-217-B

Pour un taux moyen de 10 sollicitations par heure, nous obtenons pour ces éléments un taux de $960\,10^{-6}$ pannes/heure du même ordre de grandeur que le taux de pannes des interfaces d'Entrée / Sortie de l'A.P.I.
Ce calcul ne tient pas compte des taux de pannes des liaisons électriques, pneumatiques, mécaniques (rupture de clavettes par exemple).

L'opinion développée par Siemens est que globalement, 95% des pannes ont une cause extérieure à l'automate. Il estime, de plus, que près de 95% des pannes imputables à l'automate sont dues aux interfaces d'entrée / sortie |SIE- |.
Par cause extérieure à l'automate, il faut comprendre:
- les défaillances des capteurs, préactionneurs, actionneurs;
- les liaisons coupées ou court-circuitées;
- les pannes d'alimentation;
- les défaillances mécaniques de la P.O.;
- les incidents mécaniques liés aux perturbations apportées à l'environnement de la P.O. (variation dans la qualité du produit traité, positionnement défectueux de pièces ...);
- les erreurs de conduite commises par les opérateurs.

Pour mieux cerner les principales causes de défaillances, il faudrait effectuer une enquête auprès des entreprises. On peut toutefois tirer des remarques précédentes les enseignements suivants:
- pour des automatismes ayant peu d'entrée / sortie, la fiabilité de l'unité de traitement est prépondérante.
- Pour des automatismes prenant en compte beaucoup d'entrée/ sortie et utilisant donc un grand nombre de capteurs et d'actionneurs, la fiabilité de l'ensemble sera d'autant plus tributaire des éléments externes à la commande que la cadence de la partie opérative sera élevée.

A ces considérations, nous ajoutons l'opinion développée dans |DEI-82| sur la sûreté des automatismes pilotés par A.P.I.

Cet auteur comparant les circuits de commande électromécanique aux automates programmables justifie sa réserve face à l'emploi des automates par deux critères qui sont:

- les modes de défaillance mal connus et les possibilités de défaillances multiples engendrées par une même perturbation.
- l'impossibilité d'envisager les conséquences des défaillances de chacun des composants de l'automate.

Cette opinion est interessante car elle met en évidence des faits irréfutables, tels que la possibilité de pannes simultanées et un avis contestable en ce qui concerne l'impossibilité de déduire les conséquences d'une défaillance. Ce dernier point est infirmé, notamment par la démarche de Merlin Gérin, qui a déduit les taux de pannes des automates de la série PB 400 par application de la norme MIL-HDBK-217-B. De plus, une approche système permet à l'utilisateur de se contenter de la connaissance globale de la fiabilité de l'automate.
En fait, l'automate programmable industriel, comme tout calculateur industriel, est un produit trop jeune pour qu'on lui fasse confiance dans la commande des machines dangereuses.
Il n'est pas étonnant, dans ces conditions, que les constructeurs d'A.P.I. se préoccupent des problèmes de sûreté de fonctionnement de leurs produits.

Nous présentons maintenant quelques moyens habituellement proposés pour améliorer la sûreté des automatismes.

4.2 SURETE LIEE A L'AUTOMATE DE COMMANDE

Deux aspects sont ici abordés. Le premier concerne la validation de la réalisation. L'objectif est ici, à la fois de vérifier que l'automatisme répond aux spécifications du cahier des charges, mais aussi qu'aucune évolution, en l'absence de défaillance, ne puisse aboutir à un fonctionnement dangereux. Le traitement de l'information étant le plus souvent confié à un dispositif de commande programmé, nous parlons de validation du logiciel, ce qui inclut la validation des spécifications.

Le deuxième aspect abordé se rapporte à l'amélioration de la sûreté face aux défaillances matérielles probables. Les solutions les plus souvent proposées y sont étudiées.

4.2.1 VALIDATION DU LOGICIEL

Nous nous bornons à passer en revue les différentes voies suivies pour la validation des logiciels relatifs aux automatismes tels que nous les avons définis.
Les modules logiciels propres à l'automate, tels que gestion des entrées / sorties, comptage / temporisation, traitement numérique, régulation ... sont toujours simples et largement éprouvés.
L'intervention de l'utilisateur est essentiellement limitée au contrôle du séquencement des tâches. C'est donc le logiciel correspondant à cette fonction, essentielle en temps réel, qu'il faut valider.
Des outils spécifiques de représentation sont utilisés, notamment les réseaux de Pétri |DAC-76| |BRA-83| et le grafcet qui est plus particulièrement destiné aux automatismes logiques (Norme UTE C03 190) |BLA-80|. Mais l'existence d'outils ne règle pas l'aspect méthodologie. L'utilisation de Macro Etape |MOA-81| correspond à une approche modulaire qui reste insuffisante. Une analyse descendante s'apparentant à la programmation structurée, par le fait qu'elle utilise des structures types (représentées par un grafcet) est proposée notamment dans PIASTRE |CAR-83|.
Dans tous les cas, la description doit être validée puis codée pour être rendue exécutable.

L'utilisation de traducteurs/compilateurs limite les risques d'erreurs incombant à cette dernière phase |PAR- | |BOU-83|.
Pour ce qui est de la validation de la description fonctionnelle, nous empruntons à |VAL-80| la classification suivante:

- validation par dialogue
- validation par analyse
- validation par simulation
- validation par preuve formelle.

La validation par dialogue correspond à la mise en commun du savoir-faire de l'automatiseur et de l'automatisé.

La validation par analyse est utilisée pour les descriptions par réseaux de Pétri pour lesquels il a été défini les "bonnes propriétés" |BRA-83|. Ceci a donné naissance à des méthodes de validation par réductions de réseaux |BOU-78|, |PRA-79|, |BOU-80|, ou par recherche de composantes conservatrices ou répétitives comme dans OGIVE |PRA-79|.

La validation par simulation est parfois adoptée pour valider des descriptions par Réseaux de Pétri |MOA-79|, |MIC-79|, mais elle est surtout mise en oeuvre par les utilisateurs de grafcet pour qui l'interprétation du graphe est essentielle.
La simulation peut être purement interactive, l'opérateur forçant alors les entrées. Elle peut être automatique lorsque la partie opérative est simulée par un grafcet qui peut être considéré comme un graphe Dual de celui de commande. Ce type de modélisation présente un inconvénient majeur puisqu'il représente la P.O. dans le contexte de la commande. Les véritables valeurs des entrées à chaque instant ne sont pas connues, il est donc impossible de simuler l'apparition d'événements anormaux. Un modèle de P.O. tel que nous le présentons dans la deuxième partie pourrait améliorer notablement la simulation.

La validation par preuve formelle consiste à utiliser deux outils différents pour la même description et à vérifier l'équivalence des deux descriptions.
Des outils adaptés aux automatismes doivent être utilisés. Par exemple, la création et la validation par simulation de ces logiciels passent par:
-l'utilisation d'un langage de déscription naturel (type grafcet) et une méthode d'analyse qui reste encore à affiner;
- la création de simulateurs utilisant un véritable modèle de comportement de la P.O. et des interfaces (capteurs, préactionneurs, actionneurs;
- la génération automatique du programme d'application à partir du cahier des charges ainsi décrit et validé.

Dans les paragraphes 2.2 et 2.3, nous étudions des solutions, couramment proposées par les fabricants d'A.P.I. pour améliorer la sûreté liée au matériel. Les références suivantes |CLA-84|, |ITI-82|, |SYK-82|, |DEI-83|, |CGE-82|, |HOP-82|, |WEN-82| peuvent être consultées ainsi que les notes d'application des constructeurs (notamment Merlin Gérin, Siemens, CGEE, ...).

4.2.2 AMELIORATION DE LA SURETE LIEE AU MATERIEL SI LA CONTINUITE DE MISSION N'EST PAS IMPERATIVE

L'objectif est ici la détection des erreurs dues aux défaillances du matériel et éventuellement du logiciel d'une part, à l'influence du milieu (notamment susceptibilité électromagnétique) d'autre part. Ces fautes peuvent agir sur le séquencement de l'unité centrale, entraîner des erreurs d'adressage ou polluer les données ou les programmes.
En cas de panne détectée, les sorties sont forcées dans un état préétabli, correspondant à un fonctionnement en sécurité, ou maintenues.
Nous notons λa le taux de pannes de l'automate qui s'exprime par la relation: $\lambda a = \lambda uc + ne\ \lambda e + n0\ \lambda 0$

λuc représente le taux de pannes de l'unité centrale

λe celui d'une entrée

$\lambda 0$ celui d'une sortie

ne et n0 sont les nombres d'entrées et de sorties.

4.2.2.1 Approche théorique

Nous appelons ensemble primaire les organes qui participent directement à la mission. L'ensemble secondaire regroupe les éléments introduits pour tester le système.

Soit un index i permettant de distinguer les paramètres relatifs à l'ensemble primaire ($i \equiv p$) de ceux relatifs à l'ensemble secondaire ($i \equiv s$).

Notons Pci = $\mathcal{P}_s$ { une panne d'un ensemble i est détectée sachant qu'une panne affecte cet ensemble. }

où $\mathcal{P}_s$ représente la probabilité asymptotique ,Pci un taux de couverture

Soit $\lambda_{i,c}$ le taux de pannes détectables de l'ensemble i

$\lambda_{i,nc}$ le taux de pannes non détectables du même ensemble

Alors: $Pci = \frac{\lambda_{i,c}}{\lambda_i}$ |4-2|

$\lambda_i = \lambda_{i,c} + \lambda_{i,nc}$

nous avons donc: $Pcp = \frac{\lambda_{p,c}}{\lambda_p}$ pour l'ensemble primaire.

Si le mécanisme de détection est autotestable, nous définissons de la même façon:

$Pcs = \frac{\lambda_{s,c}}{\lambda_s}$ |4-3|

Il est clair qu'une défaillance de tout ou d'une partie de l'ensemble secondaire correspond à une dégradation de performances. Toutefois, un tel événement ne rend pas la machine dangereuse tant que l'ensemble primaire reste sain. A une défaillance de cet ensemble correspond donc une perte d'efficacité totale ou partielle du système de détection.

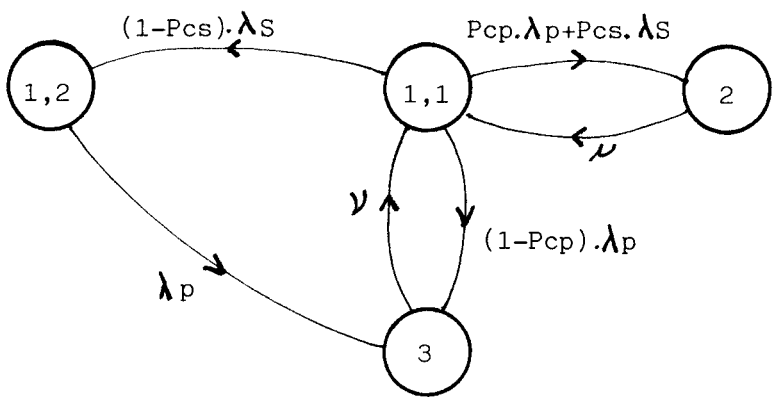
Pour simplifier notre étude générale, nous admettrons que toute panne de l'ensemble secondaire le rend inefficace. Dans ces conditions, la probabilité asymptotique de détecter une erreur issue de l'ensemble primaire est supposée nulle, lorsque l'ensemble secondaire est défaillant. Nous supposons que le temps de latence d'une panne détectable est très faible (hypothèse H3 § I 3.3.1). Le graphe des états d'une commande à détection d'erreurs obtenu en tenant compte de ces hypothèses est donné Fig 4.4. Nous supposons également qu'une défaillance détectée provoque une interruption de service (état défaillant), même si c'est l'ensemble secondaire qui est en cause.

Dans la suite de l'évaluation de performance, nous nous limitons à l'étude des automatismes réparables. Nous supposerons que les temps de réparation restent toujours faibles vis-à-vis des temps entre pannes (hypothèses H1 et H2).

En négligeant le temps de latence des pannes détectables, et en négligeant les temps de réparation, nous en arrivons à considérer

que seule l'existence d'au moins une panne non détectable peut entraîner l'existence de pannes multiples.
L'interprétation des résultats obtenus doit éventuellement tenir compte de ces simplifications.

Etats non défaillants	E1,1 commande saine E1,2 panne non détectée de de l'ensemble secondaire
Etats défaillants	E2 panne détectée de l'ensemble primaire ou secondaire E3 panne non détectée de l'ensemble primaire

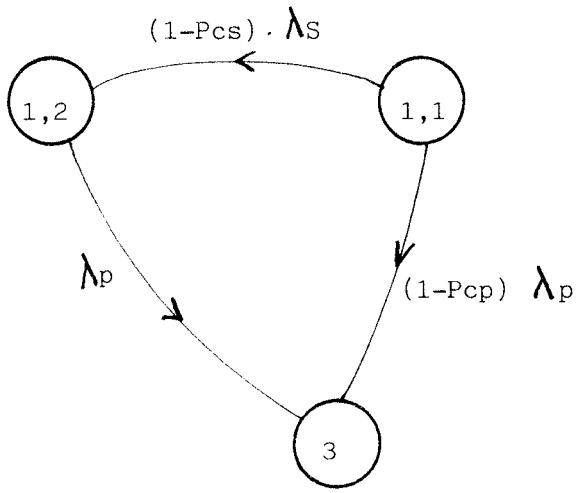


- figure 4.4 -

Evaluation de la sécurité:

Le seul état dangereux correspond à E3 que nous rendons absorbant. Pour un système réparable, compte tenu des hypothèses adoptées, nous obtenons le graphe suivant (Fig 4.5).

non dangereux	E1,1 Structure saine ou en réparation après erreur, détectée E1,2 Mise hors service non détectée de l'ensemble primaire
dangereux	E3 panne non détectée de l'ensemble primaire



- figure 4.5 -

Nous en déduisons:

$$S(t) = \frac{1}{\lambda p - \alpha} \left| Pcp.\ \lambda p e^{-\alpha t} - (1-Pcs) \cdot \lambda s . e^{-\lambda p t} \right| \qquad |4\text{-}4|$$

avec $\alpha = (1-Pcp) . \lambda p + (1-Pcs) . \lambda s$

qui donne:

$$MTFMF = \frac{Pcp\ \lambda p}{(\lambda p - \alpha)\alpha} - \frac{1-Pcs}{\lambda p - \alpha} \frac{\lambda s}{\lambda p} \qquad |4\text{-}5|$$

La relation |4-4| s'écrit:

$$S(t) = \frac{1}{1 - \frac{(1-Pcs).\lambda s}{Pcp.\lambda p}} \left[e^{-[(1-Pcp).\lambda p + (1-Pcs).\lambda s]t} - \frac{(1-Pcs).\lambda s}{Pcp.\lambda p} . e^{-\quad pt} \right]$$

|4-6|

Si Pcp $\lambda p \neq 0$, nous pouvons tirer de |4-6|

$$\lim_{(1-Pcs)\,\lambda s \to 0} S(t) = e^{-(1-Pcp)\,\lambda pt}$$

La sécurité tend alors vers la valeur obtenue avec un dispositif de détection non tolérant |1-13|.
Pour s'approcher au mieux de cette situation idéale, nous avons la possibilité d'améliorer l'efficacité des autotests (Pcs $\rightarrow$ 1) et/ou la fiabilité des organes secondaires ($\lambda s \rightarrow 0$).
La sécurité sera alors directement liée à l'efficacité de la détection comme nous l'avions signalé au premier chapitre.

Il est interessant de noter que pour $\frac{(1-Pcs)\,\lambda s}{Pcp\ \lambda p} = 1$, nous obtenons $S(t) = e^{-\lambda pt}$.
Dans ces conditions, la sécurité est identique à celle obtenue avec un système sans détection.
Ce paramètre donne la performance minimale à atteindre par le dispositif de détection. Pour obtenir un gain en sécurité, il faut $\frac{(1-Pcs)\,\lambda s}{Pcp\ \lambda p} < 1$

Nous noterons $Sc = \frac{(1-Pcs)\ \lambda s}{Pcp\ \lambda p}$ ce facteur d'amélioration de la sécurité des systèmes à détection de défaillance.

Pour étudier la fiabilité, nous reprenons la graphe de la Fig 4.4. Après avoir éliminé les arcs relatifs aux réparations, nous groupons les deux états défaillants E2 et E3. Nous obtenons le graphe de fiabilité suivant (Fig 4.6).

Nous avons:

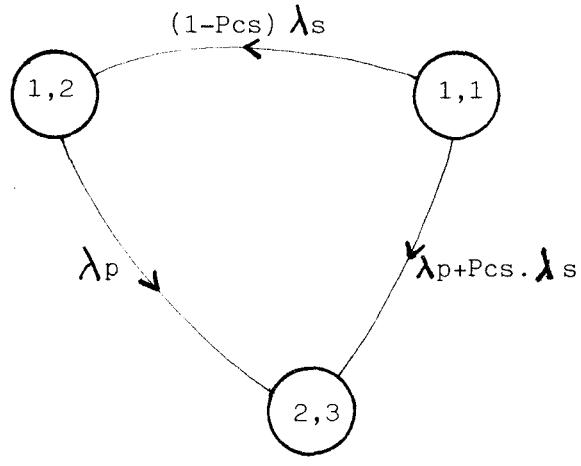
$$R(t) = e^{-\lambda pt} . [\, 1 - Pcs(1-e^{-\lambda st})]$$ |4-7|

Soit:

$$MTFF = \frac{\lambda p + (1-Pcs)\,\lambda s}{\lambda p(\lambda p + \lambda s)}$$ |4-8|

Nous avons évidemment $IR(t) = e^{-(\lambda p + \lambda s)t}$ et $IR(t) \leqslant R(t)$

non défaillant	E1,1 structure saine E1,2 panne non détectée de l'ensemble secondaire
défaillant	E2,3 système défaillant à la suite d'une défaillance de l'ensemble primaire ou d'une défaillance de l'ensemble secondaire.



- figure 4.6 -

La relation |4-7| fait ressortir la diminution de fiabilité par rapport à un dispositif de commande sans détection de défaillance. Cette dégradation dépend évidemment de la fiabilité des organes secondaires. Avec les hypothèses retenues, la dégradation est d'autant plus importante que le système de détection détecte ses propres défaillances avec efficacité.
Par contre, le taux de couverture des pannes de l'organe primaire reste sans influence sur la fiabilité.
Le dilemme sécurité/fiabilité est ici illustré par l'influence de Pcs. Augmenter Pcs améliore la sécurité au détriment de la fiabilité.

Pour Pcs ≠ 1, on trouve:

$R(t) = e^{-(\lambda p + \lambda s)t}$ $\qquad MTFF = \frac{1}{\lambda p + \lambda s}$

$S(t) = e^{-(1-Pcp)\ \lambda p}$ $\qquad MTFMF = \frac{1}{(1-Pcp)\ \lambda p}$

Dans ce cas, la sécurité est effectivement celle obtenue avec un système de détection totalement intolérant. Par contre, la fiabilité est diminuée par l'introduction du système de détection.
Réduire λs est bénéfique sur les deux plans.
Il est clair que $IR(t) = e^{-(\lambda p + \lambda s)t}$

Donc: $\forall \tau \in |0,t|$, $IR(\tau) \leqslant R(\tau)$.
L'égalité est obtenue pour toute valeur de τ si Pcs = 1.

Appliquons les résultats obtenus pour évaluer la sûreté de différentes architectures souvent proposées par les constructeurs d'A.P.I.
Comme au premier paragraphe, nous utilisons les valeurs numériques extraites de |CLA-84| pour une configuration à 160 entrées/160 sorties.
Pour un tel automate et, si aucun moyen n'est utilisé pour améliorer la sûreté de fonctionnement, nous avons:
$\lambda a = 1{,}23\ 10^{-3}$ pannes/heure, ce qui donne:
MTTF = MTFMF = 815h

4.2.2.2 Solution à un automate

a) Utilisation de mécanismes internes de détection

Des mécanismes internes de détection voire même de correction peuvent être mis en place.
Le système le plus élémentaire, et aussi le plus répandu, est constitué par le chien de garde. Ce mécanisme couvre en grande partie les erreurs de séquencement de l'unité de traitement dues à des défaillances consistantes ou non. Par contre, la zone des données n'est pas protégée (y compris le plan des entrées/sorties).
Des techniques complémentaires utilisant des codes non triviaux (contrôle de parité, code k parmi n, code redondant) permettent de détecter les erreurs par des circuits de contrôle d'appartenance au code, eux-mêmes autotestables |RED-73|,|AND-73|, |DIA-74|, |CAR-68|, |MER-74|.

Certains de ces codages permettent même la correction des erreurs. Des circuits LSI utilisant ces propriétés existent. Nous citerons le DP 8400 de N.S.* qui est capable de détecter trois erreurs et d'en corriger deux.

La même technique peut être envisagée pour le codage de l'évolution du grafcet. Ceci a été proposé pour les réseaux de Pétri |MAR-75|.

Des techniques d'échos lors des échanges entre fonctions internes à l'automate (par exemple entre unité de traitement et carte périphérique) permettent également de couvrir certaines erreurs.

A titre d'exemple, Merlin Gérin pour la série PB 400 ajoute une carte de surveillance. Il estime le taux de couverture de l'ordre de 52% pour la configuration de référence. La carte assurant la détection (ensemble secondaire) a un taux de panne de $30,7\ 10^{-6}$ pannes par heure. Le taux de couverture des autotests de cet ensemble est de 96%. Les valeurs retenues sont:

$\lambda p = 1227\ 10^{-6}$ pannes/heure et $1 - Pcp = 0,49$

$\lambda s = 30,7\ 10^{-6}$ pannes/heure et $1 - Pcs = 0,048$

Soit $\lambda(1-Pc) == 608\ 10^{-6}$ pannes/heure au lieu de $1227,3\ 10^{-6}$ pour la même configuration sans test; ce qui donne:

$S_c = 2,3\ 10^{-3}$.

	sans test	avec test
Fiabilité	MTFF = 815 h	MTFF = 814 h
Sécurité	MTFMF = 815 h	MTFMF = 1593 h

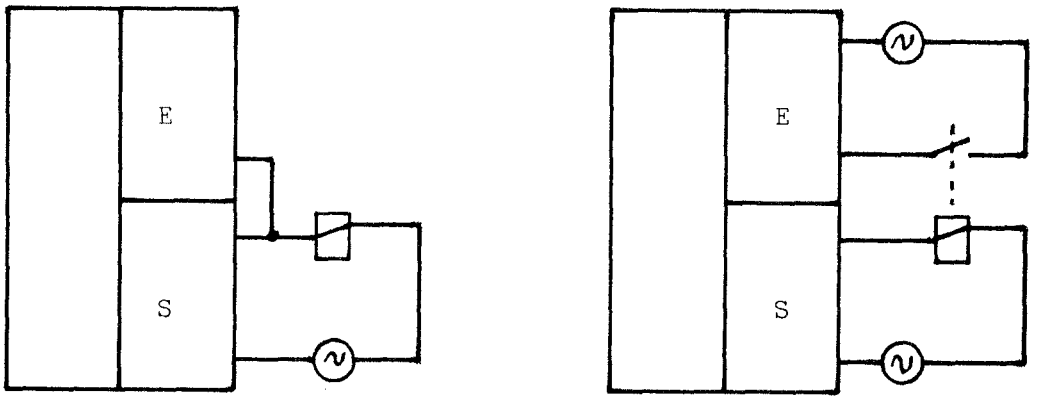
Cette réalisation illustre le gain de sécurité que peut apporter cette méthode sans pour autant trop réduire la fiabilité. Mis à part le chien de garde, cette technique est du ressort exclusif du constructeur de l'automate. Les autres solutions proposées résultent d'un choix d'architecture et peuvent être mises en oeuvre par l'utilisateur.

b) Le rebouclage des sorties

Nous avons vu que le taux de pannes des sorties comme des entrées devient prépondérant au niveau de l'automate lorsque le nombre des E/S augmente.

* National Semi conducteur

L'objectif est ici de couvrir les défauts des sorties. Le montage est celui de la figure 4.7. Deux options possibles sont représentées Fig 4.7a et 4.7b.



- a - - b -

- figure 4.7 -

Dans la première solution, la sortie est directement rebouclée sur une entrée. La deuxième proposition inclut dans la boucle le préactionneur. Un contact auxiliaire est alors nécessaire, ce n'est pas toujours réalisable.

Traitons le rebouclage direct.
Nous avons pour l'automate représentant l'ensemble primaire:

$$\lambda p = \lambda uc + ne\ \lambda e + n0\ \lambda 0$$

Le temps de latence d'une défaillance sur une sortie ou sur une entrée de rebouclage est faible. Dans ces conditions, nous admettons comme improbables les pannes simultanées d'une sortie et de l'entrée de rebouclage correspondante. Le taux de couverture est:

$$Pcp = \frac{n0\ \lambda 0}{\lambda p}$$

Les organes secondaires sont constitués des entrées de rebouclage. Toute défaillance de ces entrées conduit , à terme, à une détection de défaut. Dans ces conditions, nous avons donc:

$\lambda s = n0\ \lambda e$ et $Pcs \neq 1$

Nous pouvons en déduire: IR(t) = R(t) et Sc = 0.

Pour l'automate pris en exemple, nous avons:

$\lambda p = 1227\ 10^{-6}$ pannes/heure

$\lambda s = 419\ 10^{-6}$ pannes/heure

Pcp = 0,44

qui conduit à MTFF = 607h

MTFMF = 1455h

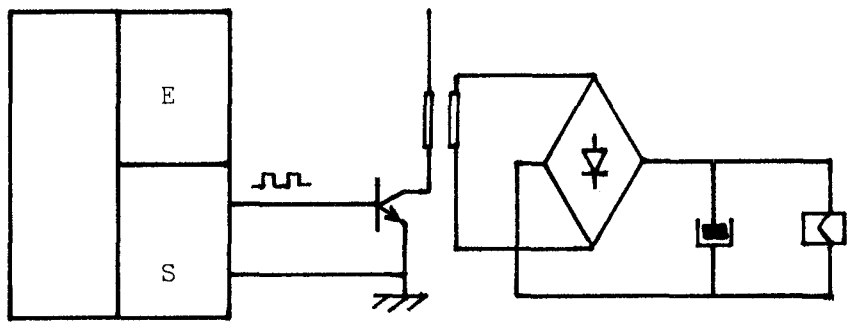
Cette architecture permet de diminuer l'insécurité due aux pannes des interfaces de sorties. Par contre, la fiabilité est fortement dégradée si le nombre de sorties est important puisque l'on passe de MTFF = $\frac{1}{\lambda p}$ sans test à MTFF = $\frac{1}{\lambda p + \lambda s}$ soit MTFF = 0,75 $\frac{1}{\lambda p}$ avec rebouclage.

Le deuxième montage élimine l'insécurité due aux préactionneurs mais conduit à une dégradation plus importante de la fiabilité puisque les contacts auxiliaires ont un taux de panne du même ordre de grandeur que les circuits d'interface.

c) Sorties autotestables ou sorties dynamiques |DEI-83|

Dans la logique statique, l'information logique est donnée sous forme d'un niveau. Une défaillance provoquant un collage à "1" ou à "0" génére donc une erreur. Dans la logique dynamique, l'ordre "actif" est fourni par un signal périodique. L'absence d'ordre est formée par un niveau maintenu correspondant à la mise en sécurité.

La figure 4.8 présente symboliquement un tel montage.



- figure 4.8 -

Tout collage de la sortie place le système en fonctionnement de sécurité. Ce dispositif joue également le rôle de chien de garde automatique dans la mesure où l'automate, pour générer un ordre, complémente la sortie correspondante, à chaque exécution de son programme. Le dispositif de sortie, grâce à son rôle de filtre passe bande élimine en partie les erreurs dues à un défaut d'exécution.

Dans ce montage, il n'y a pas à proprement parlé de circuit secondaire.

Toutefois, nous considérons comme circuits secondaires les composants ajoutés pour filtrer la sortie dynamique. Si nous notons λi leur taux de panne, nous obtenons:

$\lambda s = n^o \lambda i$ (taux qu'il faut évidemment réduire).

Comme dans le cas du chien de garde, il nous est difficile d'évaluer le taux de couverture, mais nous savons que sont couverts:

- certaines pannes de séquencement,
- les collages intervenant avant les filtres.

Ne sont pas couverts les pannes consistantes d'adressages d'E/S, les parasitages de contexte interne. Il faudrait pour cela, appliquer le même principe aux données internes.

En conclusion, le taux de couverture est certainement relativement important alors que la fiabilité est assez peu dégradée. Il est à noter que certains automates Klockner- Moeller utilisent des sorties dynamiques constituées de monostables.

d) Doublement des entrées

Les organes secondaires sont ici formés des entrées dupliquées.

Donc: $\lambda s = ne\ \lambda e$.

Les taux de couverture sont: $Pcp = \frac{ne\ \lambda e}{\lambda p}$; $Pcs \neq 1$

Dans ces conditions, nous obtenons pour notre automate de référence:

MTFF = 607h

MTFMF = 1234h

En utilisant à la fois le rebouclage des sorties et le doublement des entrées, nous obtenons:

$\lambda s = ne\ \lambda e + no\ \lambda e$

Pcs $\neq$ 1

$Pcp = \frac{ne\ \lambda e + no\ \lambda e}{\lambda p}$

MTFF = 457h

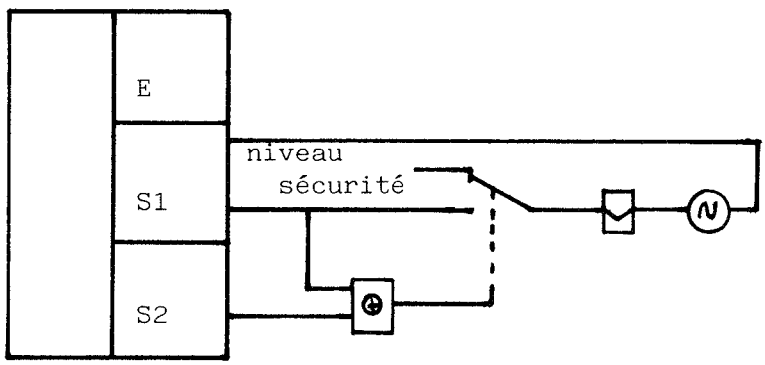
MTFMF = 3704h

Ce résultat illustre bien le dilemme sécurité/fiabilité.

e) Duplication des sorties

Dans le cadre de ce paragraphe, la duplication porte uniquement sur les organes de sorties de l'A.P.I. et en aucune façon sur les préactionneurs et actionneurs.

Le principe de base est illustré par la figure 4.9.



- figure 4.9 -

Cette architecture nécessite un dispositif de détection/réaction bien qu'il ne s'agisse pas à proprement parlé de redondance des sorties. En effet, en cas de panne sur un sortie, la sécurité est assurée en principe, mais pas la continuité de mission.

Dans ces conditions, nous avons:

$\lambda s = n_O(\lambda'_O + \lambda dc)$ où λdc est le taux de panne du système de détection/commutation et $\lambda'o$ le taux de panne d'un interface de sortie sans adaptation de puissance.

$Pcp = \frac{no\ \lambda o}{\lambda p}$

$Pcs = \frac{\lambda'o}{\lambda'o + \lambda_{DC}}$

Pour le calcul, nous adoptons la position suivante.

Le dispositif de détection/commutation est pratiquement identique à la partie adaptation de puissance d'une sortie normale; donc

$\lambda'o + \lambda dc \neq \lambda o$

Le taux de panne $\lambda'o$ correspond aux défaillances des ports de sorties et mécanisme d'adressage normalement couvert par les mécanismes d'adressage dans la série PB 400.

Nous adoptons alors $\lambda'o = 0,5. \lambda o.$

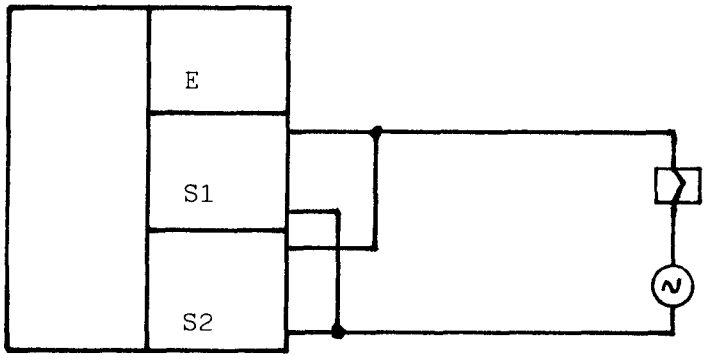
Ceci conduit à Pcs = 0,5.

Dans ces conditions, nous trouvons:

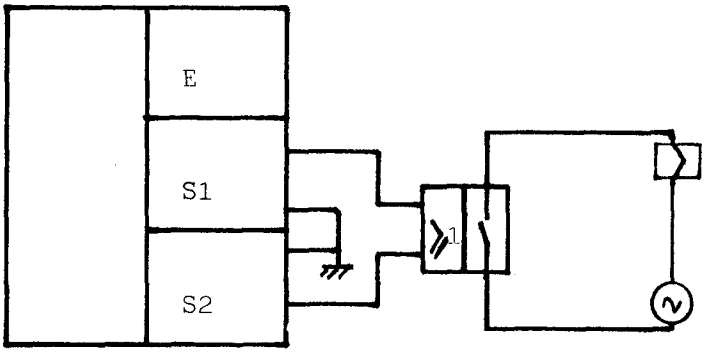
MTFF = 690h

MTFMF = 1274h

Dans la pratique, les montages proposés sont ceux des figures 4.10 et 4.11 qui réalisent respectivement un OU et un ET pour le système de détection/commutation.

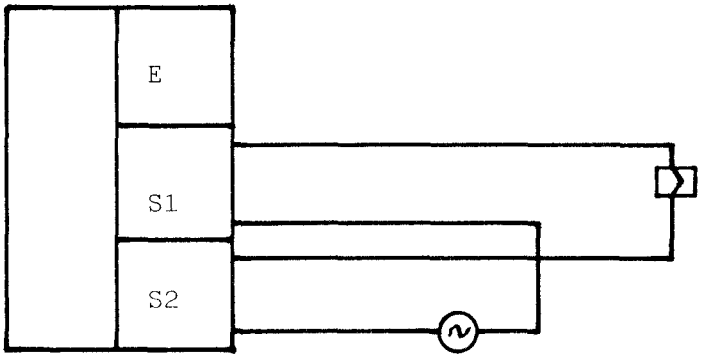


- a -

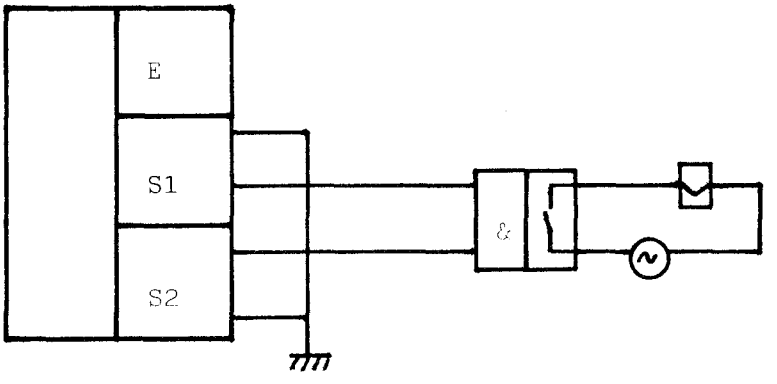


- b -

- figure 4.10 -



- a -



- b -

- figure 4.11 -

Les figures 4.10a, 4.11a correspondent aux architectures proposées couramment par les A.P.I., celles présentées en 4.10b et 4.11b se différencient des précédentes par le fait que l'étage d'adaptation de puissance n'est pas dupliqué. Ces montages sont en fait utilisés pour les sorties analogiques, l'organe non dupliqué étant le convertisseur N/A.

Notre étude porte sur l'architecture donnée en 4.11a qui correspond à une mise en sécurité par coupure d'alimentation. C'est la solution qui est en général choisie, car la législation exige une mise en sécurité en cas de coupure d'alimentation.

Pour évaluer les taux de couverture, nous sommes obligés de différencier les modes de défaillance.
Quelque soit l'origine de la défaillance, cela se traduit en sortie par un collage à "1" ou à "0". Dans la configuration choisie, un collage à "0" place l'automate dans un mode défaillant mais en sécurité (état E2 du graphe de la figure 4.4). Un collage à "1" correspond à une panne non couverte qui provoque la transition vers l'état E1,2 pour lequel la mission est maintenue,mais avec dégradation (supposée totale) du système de détection.

Remarque:
La défaillance d'une sortie est supposée affecter l'ensemble secondaire puisque la mission peut, en principe, être poursuivie par la sortie dupliquée. Nous passons en fait, dans cette configuration, à une redondance sélective active d'ordre 2 des sorties pour laquelle le mécanisme de localisation est absent.

Posons λ_o^o le taux de panne d'une sortie par collage à "0".
et λ_o^1 celui qui correspond à un collage à "1".
Il est clair que $\lambda_o^o + \lambda_o^1 = \lambda_o$
Dans ces conditions, nous avons:

$$Pcp = \frac{no\ \lambda_o}{\lambda_p} \quad ; \quad \lambda_s = no\ \lambda_o \quad ; \quad Pcs = \frac{\lambda_o^o}{\lambda_o}$$

Si les collages à "1" et à "0" sont équiprobables, alors
Pcs = 0,5 et nous obtenons dans ces conditions les mêmes résultats que précédemment, à savoir: MTFF = 690h

MTFMF = 1274h

Remarque:
Il est important de remarquer que $Sc = \frac{\lambda_o - \lambda_o^o}{\lambda_o}$. Si la sécurité est donc peu améliorée, Sc tend vers la valeur critique 1 si λ_o^o tend vers 0.

Dans le cas du montage de la figure 4.11b nous avons:

$$Pcp = \frac{no\ \lambda'_o}{p} \quad ; \quad s = no\ \lambda'_o \quad ; \quad Pcs = \frac{\lambda_o^{o'}}{\lambda'_o}$$

Avec les hypothèses précédentes, nous trouvons:

$$Pcp = 0{,}5\ \frac{no\ \lambda_o}{\lambda_p} \quad ; \quad \lambda_s = 0{,}5\ no\ \lambda_o \quad ; \quad Pcs = 0{,}5$$

ce qui donne: MTFF = 741h

MTFMF = 1016h.

Il est clair que le montage à duplication totale de la figure 4.11a est préférable avec les hypothèses choisies. Cet avis est renforcé par le fait que l'avantage essentiel de ce montage, qui est d'assurer la sécurité législative, est perdu lorque l'organe terminal n'est pas dupliqué.

Remarques:

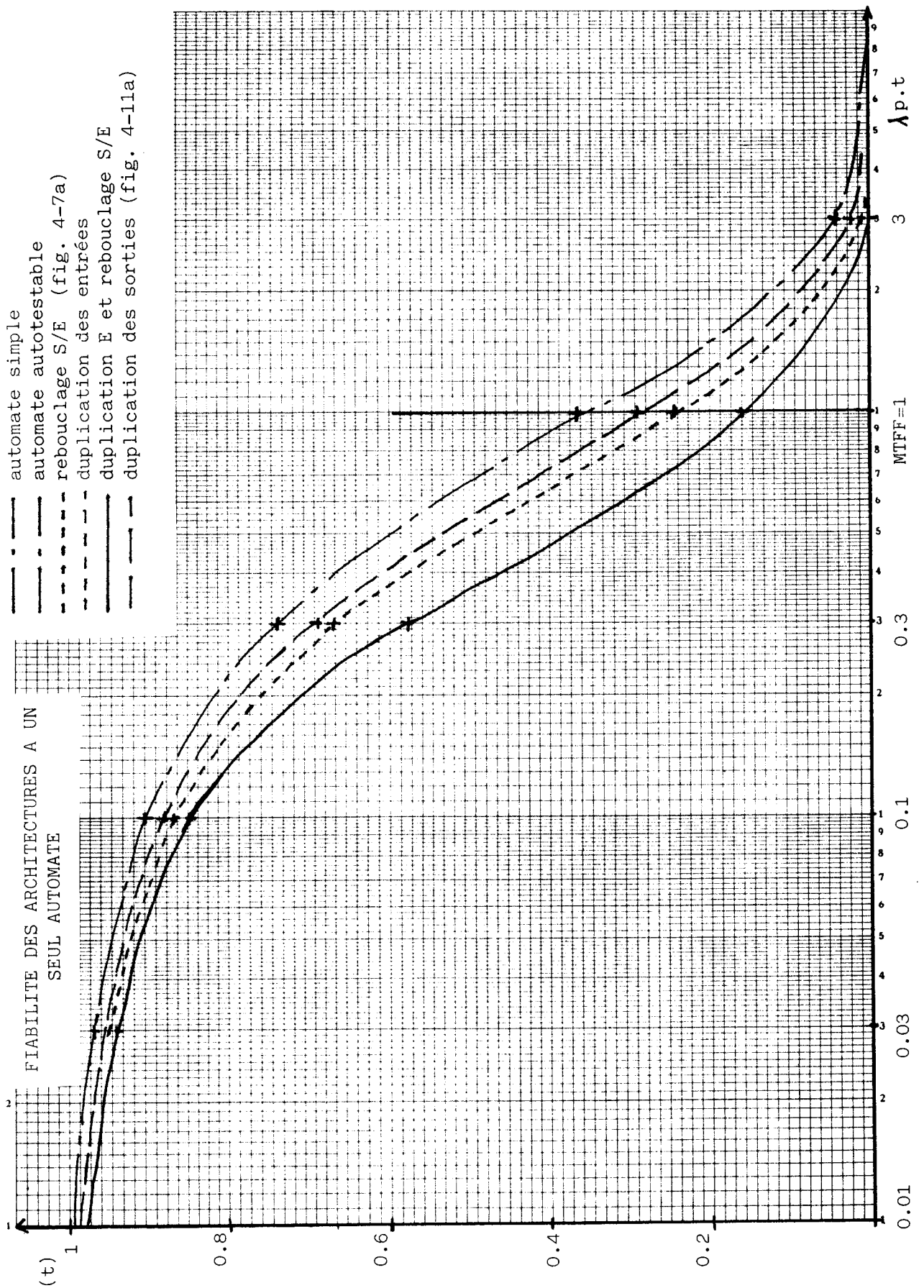
- L'intérêt de cette architecture (Fig 4.11a) est d'assurer la sécurité législative. En effet, si le MTFMF est du même ordre de grandeur qu'avec les autres systèmes de protection des sorties, les pannes dues aux sorties sont rendues non dangereuses dans la mesure où l'atteinte du fonctionnement en sécurité reste garanti. Ceci n'est pas le cas avec les autres montages.
- L'hypothèse de défaillance totale du système de détection dès la première panne de celui-ci est particulièrement pénalisante pour ce montage. En fait, l'hypothèse d'une dégradation progressive est plus réaliste.
- Les performances sont notablement améliorées par l'adjonction d'un dispositif annexe de signalisation de discordance mettant en évidence les collages à "1".

La comparaison des performances des différentes architectures à partir des valeurs moyennes de MTFF et MTFMF est critiquable dans la mesure où les probabilités de défaillances pour ces différentes valeurs de temps moyens sont différentes. Nous donnons donc figures 4.13 et 4.14 les courbes R(t) et S(t).

Le tableau de la figure 4.12 regroupe les différents résultats obtenus.

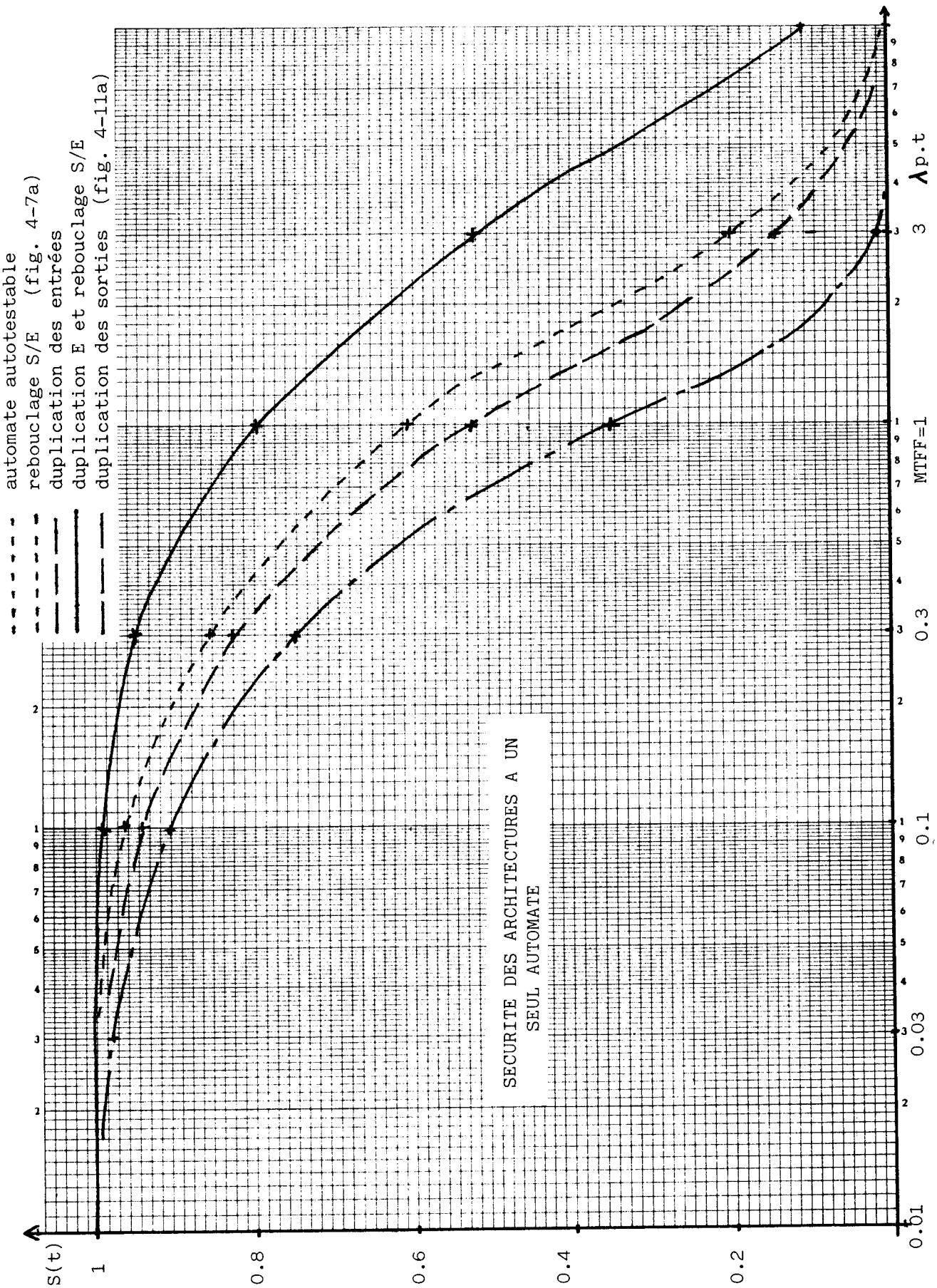
Montage	Amélioration de la sécurité	Diminution de la fiabilité	
mécanismes internes	bonne à très bonne	relativement faible	Remarque: - performances dépendantes des taux de couvertures, - les mécanismes mis en oeuvre doivent être autotestables si possible.
rebouclage sortie	élimine l'insécurité liée aux sorties	importante si beaucoup de sorties	
sortie dynamique	- diminue l'insécurité liée aux sorties - élimine en partie l'insécurité due aux défaillances de l'UC	très faible	pas de protection du contexte interne en mémoire vive.
doublement des sorties	n'élimine que partiellement l'insécurité liée aux sorties, dépend de la probabilité de collage à "0" ou à "1" des sorties	importantes si beaucoup de sorties	taux de couverture des pannes de sortie de l'ordre de 0,5 si $\mathcal{P}$(collage"0")= $\mathcal{P}$(collage"1") Amélioration si un système de détection de discordance est installé (redondance des sorties).
doublement des entrées	élimine l'insécurité liée aux entrées	importante si beaucoup d'entrées	

- figure 4.12 -



FIABILITE DES ARCHITECTURES A UN SEUL AUTOMATE

- figure 4.13 -

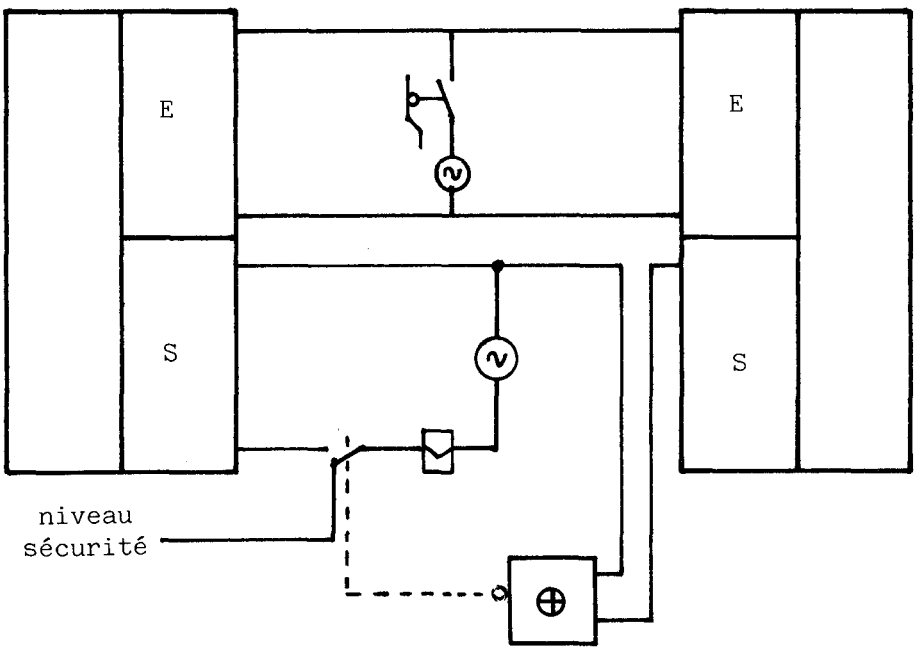


SECURITE DES ARCHITECTURES A UN SEUL AUTOMATE

- figure 4.14 -

4.2.2.3 Solution à deux automates

Le principe du test est ici de comparer les sorties de deux automates effectuant le même traitement. En cas de discordance, le système est placé en sécurité. La mission est alors interrompue; il ne s'agit pas d'une véritable redondance.
Le montage de principe est donné figure 4.15.



- figure 4.15 -

Par principe, les deux automates fonctionnent simultanément, ce qui élimine les problèmes d'échange de contexte. Seuls les tests de vraisemblance au niveau des sorties sont indispensables pour éliminer les ordres erronés. Nous ne nous préoccupons pas ici des inévitables problèmes de synchronisation.

Notons $\lambda 1$ le taux de panne de l'automate principal
$\lambda 2$ le taux de panne de l'automate de référence
λDC le taux de panne de l'organe de détection/commutation.

Nous supposons que toute panne du système de détection/commutation entraîne la défaillance totale de l'ensemble. Dans ces conditions nous posons:

$R(t) = e^{-\lambda_{DC}t} R'(t)$

$S(t) = e^{-\lambda_{DC}t} S'(t)$

où R'(t) et S'(t) représentent la fiabilité et la sécurité du dispositif avant prise en compte des défaillances de l'organe de décision/commutation.

Donc, $\lambda_p = \lambda_1$; $\lambda_s = \lambda_2$ et Pcp = Pcs = 1.

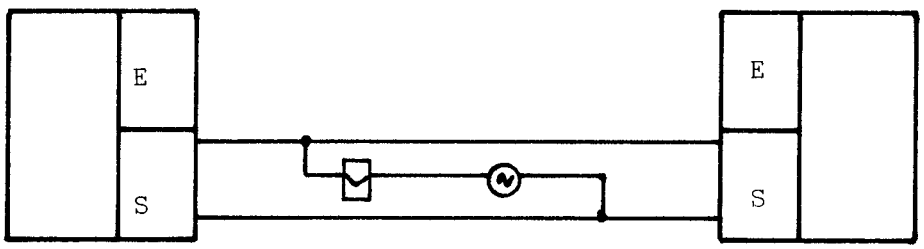
En prenant $\lambda_1 = \lambda_2$, nous obtenons pour l'automate de référence,

$S'(t) = 1 \Rightarrow S(t) = e^{-\lambda_{DC}t}$; $R'(t) = e^{-(\lambda_{DC} + 2\lambda_p)t}$

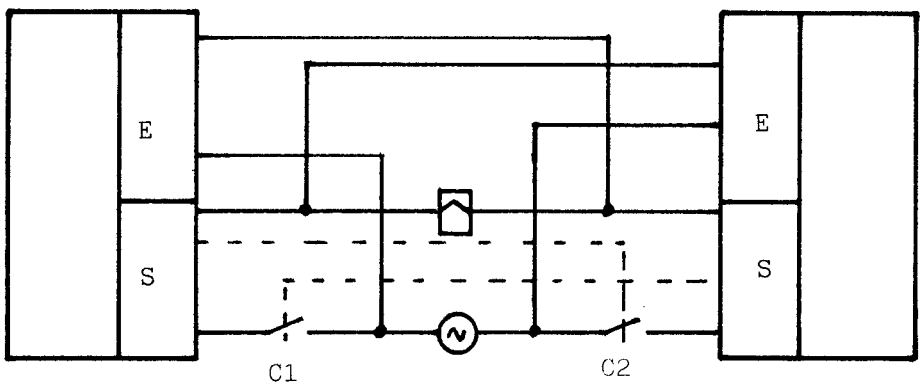
Donc, $MTFMF = \frac{1}{\lambda_{DC}}$

$$MTFF = \frac{1}{\lambda_{DC} + 2\lambda_p} \neq \frac{1}{2\lambda_p}$$

Les montages généralement proposés se rapprochent en fait de ceux qui sont envisagés dans le cadre de la duplication des sorties. Deux types de montage sont envisagés selon que la mise en sécurité est obtenue par forçage à "1" ou à "0" (Fig 4.16a et 4.16b).



- a -



- b -

-figure 4.16 -

Dans le montage 4.16b qui est le plus utilisé, la vérification de cohérence peut alors être obtenue par rebouclage croisé des sorties d'un automate sur les entrées de l'autre.
L'automate qui constate une anomalie ouvre donc le commun d'alimentation provoquant ainsi la mise en sécurité.
Dans l'hypothèse des temps de latence suffisamment courts pour éviter les pannes multiples, sur un même actionneur, la sécurité est très grande, elle est limitée par la probabilité de collage à "1" simultané des deux contacts C1 et C2 (Fig 4.16b).
Par contre, le MTFF = $\frac{1}{2\lambda 1+2ne\lambda e}$ traduit la diminution importante de fiabilité qui en résulte.

4.2.3 AMELIORATION DE LA SURETE LIEE AU MATERIEL SI LA CONTINUITE DE MISSION EST REQUISE

Pour satisfaire à la condition de continuité de mission, il faut recourir aux systèmes redondants. Une telle structure met en oeuvre:

- des unités capables d'assurer la mission dans des conditions préétablies. Le nombre d'unités donne l'ordre de la redondance,
- un mécanisme de détection et de localisation des erreurs,
- un mécanisme de réaction ou de reconfiguration.

Les mécanismes de détection et de localisation sont mis en oeuvre au niveau de chaque unité et/ou globalement au niveau de la structure. Ils constituent les éléments non fonctionnels du procédé. Ils sont formés de matériel et/ou de logiciel.

Les défaillances peuvent affecter n'importe quel élément de la structure. En adoptant l'hypothèse des temps de réparation très faibles, nous pouvons admettre que le risque de pannes multiples est très faible. Toutefois, les limites des performances des circuits non fonctionnels sont à l'origine de situations dégradées pouvant placer le système dans un état de fonctionnement dangereux. Sont considérés comme tels la présence d'une défaillance non détectée ou l'arrêt de la commande.
Compte-tenu de la contrainte de continuité de mission qui guide ici notre choix, la sécurité et la fiabilité sont confondues.

Seules les réparations pouvant être effectuées en cours de mission sont acceptables.

Par contre, nous aurons $R(t) \neq IR(t)$

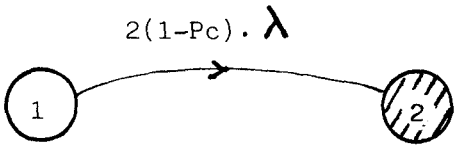
4.2.3.1 Utilisation de deux automates

La solution couramment retenue est la redondance sélective active. Chaque automate travaille sur son contexte propre, ce qui permet d'éliminer une source d'erreurs. Les deux automates traitent la commande simultanément. Au départ, un automate est élu de façon aléatoire (ou préétabli). Chaque automate exécute périodiquement ses autotests. Toute défaillance d'un automate provoque sa déconnexion.

La sécurité obtenue est alors liée essentiellement à l'efficacité des mécanismes d'autotest et à la fiabilité du circuit de sortie qui constitue généralement un point dur de la structure.

En première approximation, nous pouvons admettre que les différents états de la structure, hors circuit de commutation,dans l'hypothèse des temps de réparation et de latence très courts, peuvent être représentés par le graphe de la figure 4.17.

E1	mission assurée, un automate au moins est sain;
E2	la structure contient au moins une défaillance non détectée.



- figure 4.17 -

Avec pour chaque unité:

$$Pc = \frac{Pcp \cdot \lambda p + Pcs \cdot \lambda s}{\lambda p + \lambda s} \quad ; \quad \lambda = \lambda p + \lambda s$$

Ceci constitue une approximation justifiée dans |LAP-74|.

En effet, une panne non révélée du dispositif d'autotest d'une unité entraîne une dégradation de performance, mais ne constitue pas un état dangereux (pas de sortie erronée, pas d'arrêt de commande) comme nous le supposons ici.

Cette approximation est d'autant mieux acceptée que Pcs est proche de 1.

Dans ces conditions, $S(t) = R(t) = e^{-[2(1-Pc)\lambda + \lambda DC]t}$

où λDC représente la fiabilité du circuit de commutation.

Et $IR(t) = e^{-(2\lambda + \lambda DC)t}$

Il est à remarquer qu'une amélioration de la sécurité par rapport à la structure à un seul automate sans dispositif de test est obtenu si:

$$Pc > 0{,}5 \left(1 + \frac{\lambda_{DC}}{\lambda} \right)$$

Pour notre automate de référence dont le taux de couverture des autotests est voisin de 50%, la mise en redondance de deux automates correspond à une dégradation de la sécurité par rapport à un seul automate sans autotest. Le résultat est autre si la continuité de mission n'est pas requise (paragraphe I.2).

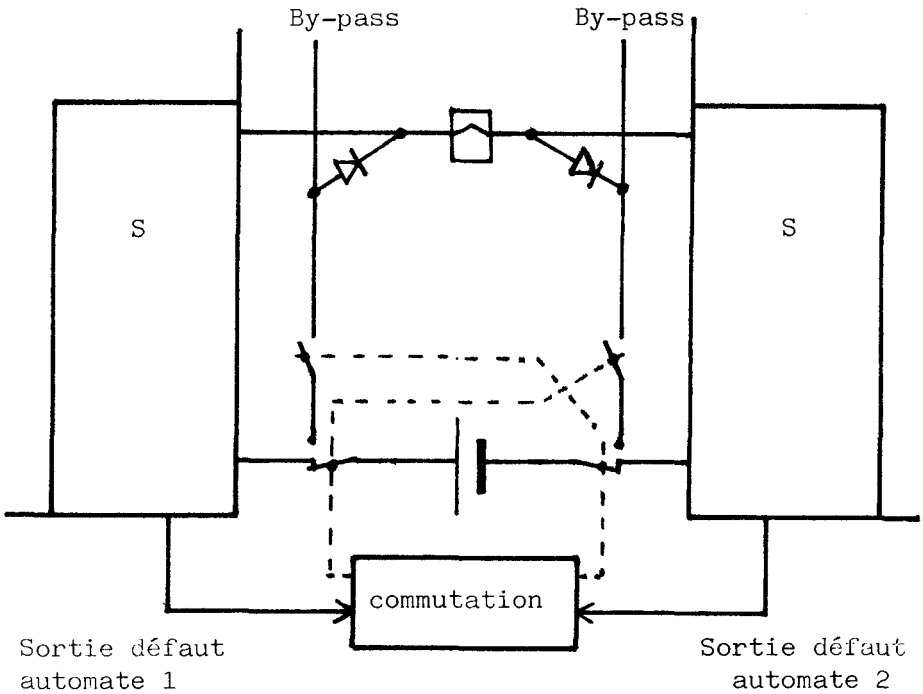
La mise en place d'un procédé de vérification de concordance entre les sorties (ou les contextes) des deux automates ne peut avoir d'effet bénéfique si la continuité de service est requise. En effet, dans le cas idéalisé où ce dispositif aurait un taux de panne nul et en négligeant le temps de latence, nous aurions le même graphe (Fig 4.17), mais avec le cartouche modifié ci-dessous.

état non défaillant	E1	tous sains ou un automate en réparation après panne localisée
état défaillant	E2	panne d'une unité détectée par discordance mais non localisable

Conclusions:

- Pour que la structure à deux automates permette d'augmenter la sécurité, lorsque la continuité de service est requise, il faut un taux de couverture des autotests excellent. L'introduction d'un comparateur n'apporte aucune amélioration.

- Comme nous l'avons montré précédemment, si un arrêt sur position de sécurité est acceptable, la structure à deux automates avec test de concordance donne une sécurité excellente. L'utilisation d'autotest des unités permet alors d'augmenter la fiabilité.
On trouve dans les notices techniques M.G. une solution réduisant l'influence de λ_{DC} par introduction de redondance au niveau de la commutation (Fig 4.18).



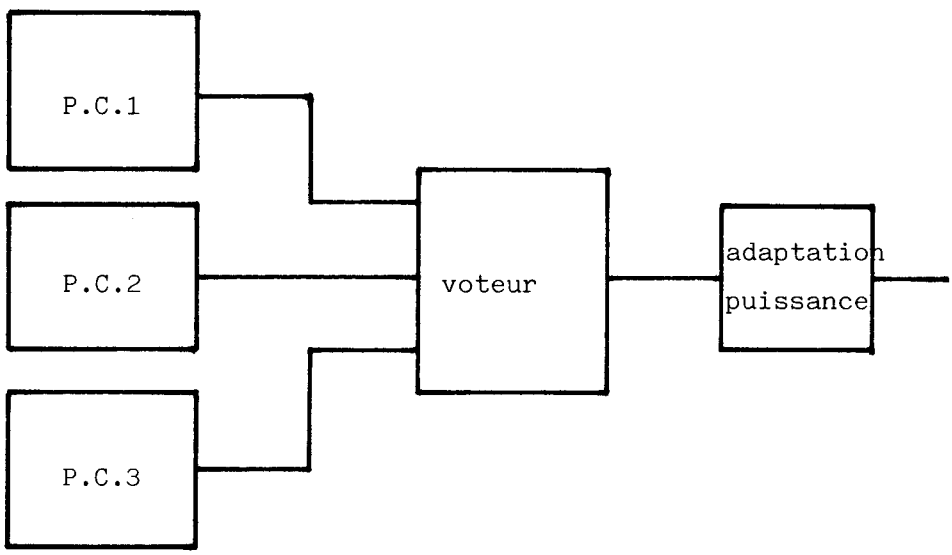
- figure 4.18 -

Lorsque les deux automates sont sains, ils participent tous les deux à la mission.

4.2.3.2 Utilisation de trois automates en redondance massive

Le schéma de la figure 4.19 rappelle l'architecture très classique de cette redondance massive d'ordre 3. La sortie est effectuée à partir d'un voteur. La concordance des sorties d'au moins deux automates sur les trois permet à la fois la détection et la localisation de défauts. Dans l'hypothèse des temps de réparation

très courts, la sécurité comme la continuité de service est limitée uniquement par les performances du voteur et du circuit de commutation qui lui est associé.
$R(t) = S(t) = e^{-\lambda_{DC}t}$; λ_{DC} est le taux de panne du voteur et du commutateur.
Dans le cadre de la commande de procédés industriels, le grand nombre de sorties mises en jeu impose une conception particulièrement soignée de ces éléments.



- figure 4.19 -

De plus, $IR(t) = e^{-(3\lambda + \lambda_{DC})t}$. Un tel dispositif peut nécessiter beaucoup d'attention de la part du service maintenance.

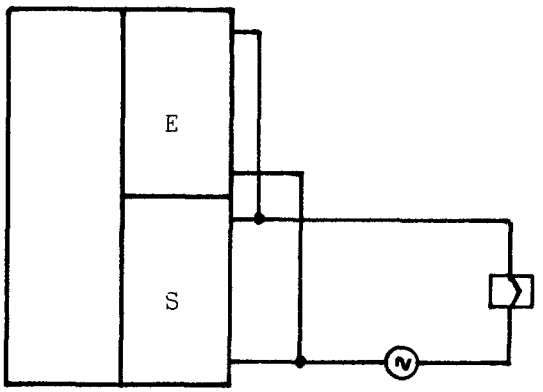
4.3 SURETE DES ELEMENTS HORS AUTOMATES

Il s'agit ici des capteurs, préactionneurs, actionneurs et de leurs liaisons; la P.O. elle-même,pouvant introduire des erreurs dont le nombre ne peut être limité que par une approche intolérante.

4.3.1 LES PREACTIONNEURS ET LEURS LIAISONS AVEC L'AUTOMATE

Le rôle des préactionneurs est d'agir sur la puissance fournie à l'actionneur, soit en tout ou rien (commutation), soit de façon analogique. Ils modulent donc la puissance fournie en agissant sur un "fluide": électricité, air comprimé, huile.
Classiquement, nous rencontrons dans cette catégorie les contacteurs électromécaniques, les distributeurs pneumatiques ou hydrauliques, les montages d'électronique de puissance (gradateurs, redresseurs commandés, onduleurs, amplificateurs...).
Pour améliorer la sécurité des préactionneurs tout-ou-rien, les méthodes généralement proposées s'apparentent à celles utilisées pour les organes de sortie de l'automate. Si la continuité de service n'est pas requise, deux éléments sont placés en série ou en parallèle selon le fonctionnement de sécurité choisi. En général, les préactionneurs sont considérés comme un simple prolongement de ces organes de sorties.
Comme nous l'avons vu dans le cas de l'automate, pour éviter les défauts multiples et pouvoir bénéficier des avantages des systèmes réparables, il faut mettre en oeuvre un moyen de détection de discordance. La solution généralement proposée dans le cas des contacteurs est l'utilisation de contacts auxiliaires commandés par les préactionneurs. Un effet similaire peut être obtenu par l'introduction de pressostat en pneumatique. Si la continuité de service est requise, il faut faire appel, soit à un circuit de commutation qui constitue un point dur, soit à un système à vote majoritaire |WEN-82|.
Pour les préactionneurs analogiques, les possibilités de redondance sont beaucoup moins souvent exploitées pour des raisons de coût. L'indépendance des systèmes en cas de défaillance est difficilement envisageable, il est donc nécessaire de faire appel à un véritable circuit de commutation qui constitue un point dur.
Pour les actionneurs, il est rare de pouvoir envisager des redondances pour diverses raisons (coût, encombrement...). Les actionneurs sont souvent considérés comme des éléments de la P.O., une approche intolérante leur est appliquée. Il est à rappeler qu'à

ce niveau la législation est très contraignante. En général, il est défini un état de sécurité pour chaque actionneur dont la défaillance risque de provoquer un accident (frein par manque de courant, système parachute...).
Les liaisons préactionneurs automate peuvent éventuellement être testées, soit par des mesures d'impédances en ligne (mais cette solution est tributaire de la qualité et de la longueur de la ligne), soit par observation du comportement de la ligne face à des micro-coupures volontaires des sorties. Le montage de la figure 4.20 permet de détecter les ruptures de connexion.



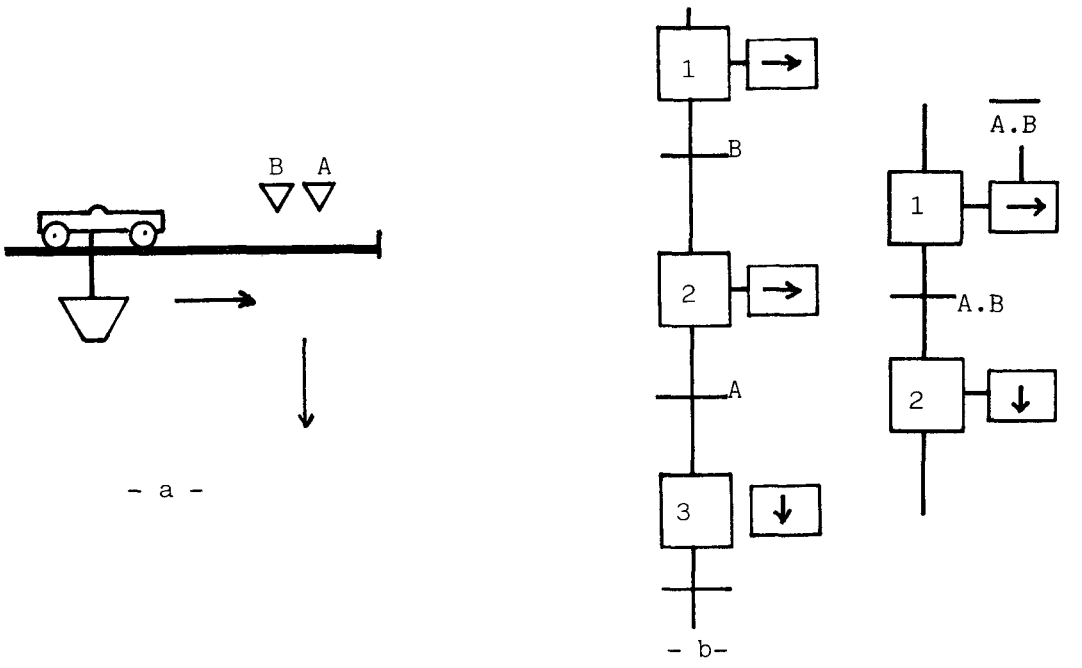
- figure 4.20 -

Chaque fois que c'est possible, les préactionneurs et actionneurs sont choisis de façon à ce qu'en cas de coupure d'alimentation, la machine se place en état de sécurité. Si cet état n'est pas destructif (ex: utilisation de parachute), un mécanisme de détection peut alors agir par coupure d'alimentation. Une approche non tolérante de cet élément de réaction généralement réduit à sa plus simple expression offre alors les meilleurs résultats.

4.3.2 LES CAPTEURS ET LEURS LIAISONS

Les propositions habituelles consistent en un doublement des capteurs (et des entrées de l'automate) ou le triplement avec vote majoritaire si la continuité de service est nécessaire. Dans le cadre des dispositifs binaires, le doublement des capteurs revient à utiliser un codage de la position permettant une détection de défaut par un mécanisme d'appartenance au code. Ceci est obtenu, soit en doublant les capteurs, soit en utilisant des contacts guidés dans le cas du fin de course électromécanique. En général, c'est l'automate qui teste par programme l'appartenance au code. Dès qu'une anomalie est détectée, il faut alors mettre en oeuvre une procédure d'exception visant à éliminer le défaut que constitue l'état transitoire. Ceci est obtenu par une relecture des entrées discordantes dans un laps de temps préétabli.
Dans le cas de capteurs analogiques, il faut effectuer un filtrage des informations avant comparaison pour s'affranchir du bruit de mesure. Il peut aussi être nécessaire de tolérer une différence entre les mesures pour tenir compte d'un éventuel gradient de la grandeur mesurée par rapport à la distance entre les éléments de prise d'information. En général, les interfaces d'entrée de l'automate sont également doublés. C'est souvent le seul moyen d'éviter les pannes de mode commun sur les capteurs doublés.
Ces éléments ajoutés ont un effet néfaste sur la fiabilité, mais leurs défaillances étant très largement couvertes, la sécurité se trouve globalement améliorée.

Une approche différente consiste à ajouter des capteurs pour éviter des actions dangereuses. Ceci relève de ce que nous appelerons la protection de zone. Ces capteurs sont alors exploités directement par la commande. Supposons, par exemple, un appareil de levage automatique (Fig 4.21a) effectuant un mouvement de translation jusqu'au capteur A, puis une descente de la charge. En plaçant un capteur B devant A et en introduisant cet élément dans la commande, comme l'indique le grafcet de la figure 4.21b nous limitons le risque de voir la charge descendre avant la zone autorisée.



- figure 4.21 -

4.3.3 CHOIX D'UNE ARCHITECTURE

Le choix d'une architecture implique un compromis coût - performance. Il doit être fait en tenant compte du coût initial, mais aussi du coût de maintenance en exploitation. La fiabilité intrinsèque (ou sa valeur moyenne) permet d'appréhender la fréquence des interventions de maintenance.

L'étude menée sur la sûreté de différentes architectures d'automate peut être étendue dans bien des cas à l'ensemble de l'automatisme.

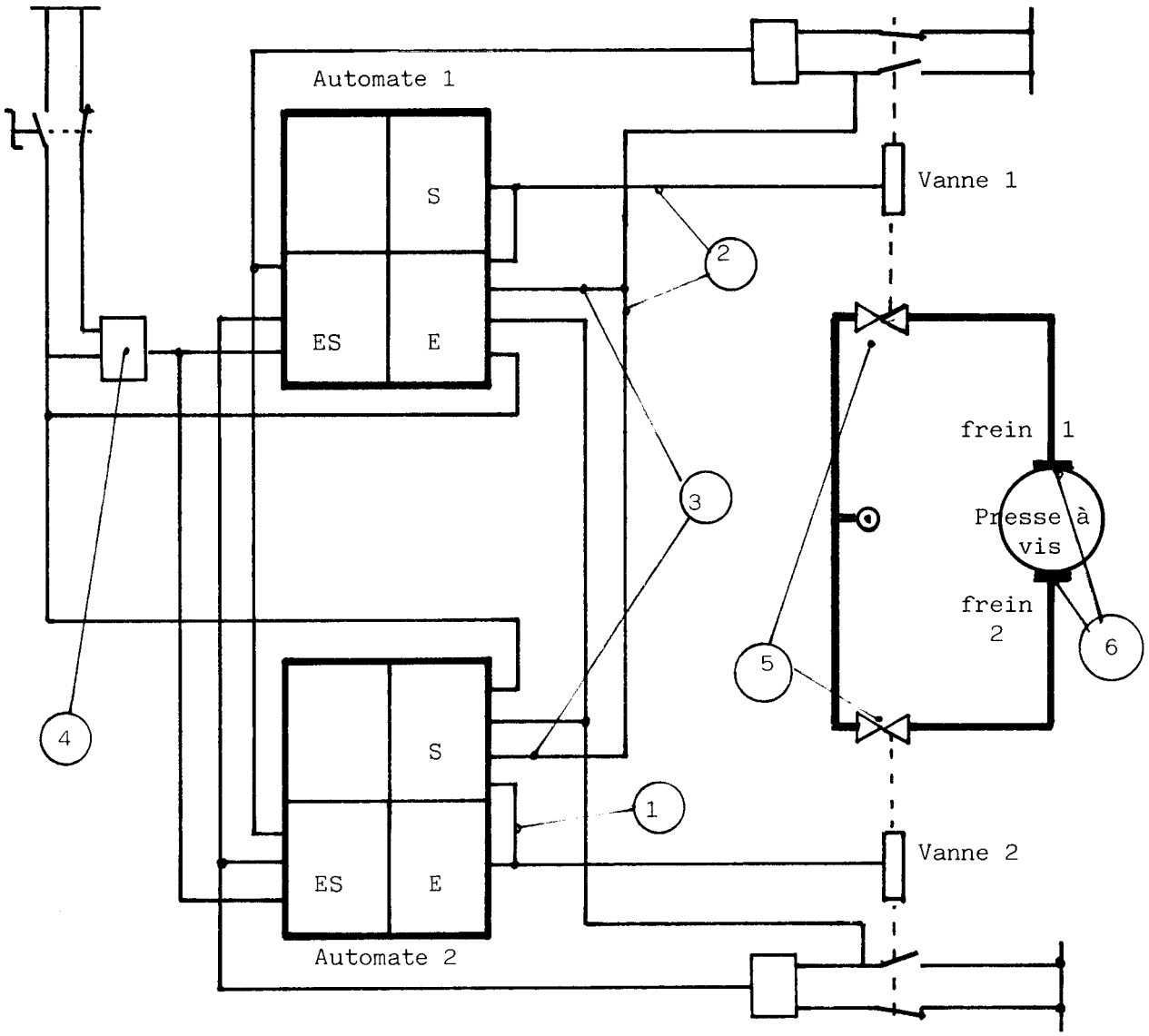
Le doublement des entrées peut s'étendre à la duplication des capteurs.

Le rebouclage des sorties sur les entrées se transpose en une chaîne sortie, préactionneur, contact auxiliaire (donnant une image de la position du préactionneur), entrée...

L'évolution des paramètres de la sûreté, constatée pour l'automate seul, peut alors être considérée comme valable pour l'automatisme complet.

Il est impossible de comparer les performances des différents montages puisque les objectifs visés sont différents.
L'utilisation de mécanismes d'autotest proposée par Merlin Gérin améliore la sécurité par rapport aux défaillances de l'unité de traitement et des communications internes avec les circuits d'interface. L'électronique d'adaptation aux circuits hors automate n'est absolument pas couverte.
Le doublement des entrées comme celui des sorties (ou le rebouclage sortie - entrée) vise à réduire la non sécurité résultant de ces seuls circuits.
La comparaison entre le rebouclage sortie - entrée et la duplication des sorties fait apparaître une meilleure performance pour la technique du rebouclage. En fait, les objectifs sont différents. L'architecture proposée dans le cadre de la duplication des sorties inclut le dispositif de réaction. Celui-ci est conçu de façon à obtenir la meilleure probabilité d'atteinte de fonctionnement de sécurité. Ce fonctionnement est alors obtenu sur la seule sortie discordante sans perturbation des autres actions commandées.
Dans le cas du rebouclage sortie - entrée, le mécanisme de réaction, dont le taux de pannes n'a pas été pris en compte pour le calcul, correspond à une mise hors service des préactionneurs qui conduit à la mise en sécurité de l'ensemble de l'automatisme. La simplicité du mécanisme de réaction justifie l'approximation faite. Le choix entre ces deux architectures résulte donc de la définition du comportement de l'automatisme en présence d'une défaillance d'une sortie de l'automate.

Il est clair que les différentes techniques proposées peuvent être simultanément mises en oeuvre.
Pour améliorer globalement la sécurité d'un automatisme, on peut aller jusqu'à une redondance totale couvrant les capteurs, l'automate, les préactionneurs et même les actionneurs. La figure 4.22 ci-après, extraite de |ITI-82| illustre cette structure.



-En cas de manque de source électrique ou hydraulique, les freins sont actionnés.

1- rebouclage E-S

2- vérification état actionneur

3- vérification de concordance entre les deux voies

4- capteurs/ éléments de commande antivalents (code 1/2)

5- redondance préactionneurs

6- redondance actionneurs de sécurité (freins).

- figure 4.22 -

En fait, l'amélioration de la sécurité passe par la mise en place de mécanismes de détection. Lorsque la continuité de service n'est pas requise pour assurer la sécurité, le système de réaction permet de placer l'automatisme dans un état privilégié. La mission est alors interrompue. Lorsque la continuité de service doit être assurée, une reconfiguration est nécessaire. Ceci impose l'existence d'éléments capables d'assurer les tâches normalement confiées à l'élément défaillant (redondance) et d'un mécanisme de localisation de la défaillance. Dans tous les cas, la détection doit être assurée. Un tel mécanisme fait appel à des éléments autotestables (programme d'autotest, chien de garde, sorties dynamiques ...) ou à une duplication des éléments à tester complétée par des circuits de mise en évidence des discordances. La sécurité est alors très étroitement liée à l'efficacité du mécanisme de détection. Lorsque la continuité de service est requise, la mauvaise performance du dispositif de localisation peut remettre en cause la sécurité. Ce phénomène est mis en évidence dans la redondance sélective active des deux automates. Le taux de couverture de 50% des programmes d'autotest assurant la fonction de localisation remet en cause le choix de l'architecture.

C O N C L U S I O N D E L A P R E M I E R E P A R T I E

Dans cette première partie, nous avons rappelé ce qu'est la sûreté de fonctionnement et quels sont les moyens mis en oeuvre pour quantifier les différentes composantes de cette sûreté.
Dans le cadre des automatismes industriels réparables, un paramètre supplémentaire a été proposé; il se définit comme la fiabilité intrinsèque (inherent reliability) d'une structure. Il permet d'évaluer la fréquence des interventions de maintenance nécessaires pour conserver au système mis en place sa sûreté originelle sans tenir compte (comme la fiabilité) de l'aptitude à poursuivre la mission.

L'amélioration de certaines composantes de la sûreté s'avère parfois indispensable. Toutefois, la conception même du logiciel de commande permet de diminuer l'influence des défaillances non consistantes (disparaissant sans réparation) sur l'évolution de l'automatisme.
Cette approche qui conduit à une meilleure tolérance de ce type de défaillance est présentée dans le chapitre trois, à partir des résultats du chapitre deux.
L'intérêt des propositions qui y sont énoncées réside dans le fait que cette amélioration portant sur l'ensemble des paramètres de la sûreté est obtenue sans apport notable de matériel, ni même de logiciel supplémentaire.

Pour les pannes consistantes, d'autres moyens doivent éventuellement être utilisés. Différentes architectures, parmi les plus couramment proposées pour améliorer la sûreté, sont étudiées dans le chapitre quatre. Ceci conduit à une analyse comparative des résultats obtenus. Toutefois, un certain nombre de conclusions d'ordre général peuvent être avancées.

Dans le cadre des automatismes réparables, il est souvent plus rentable d'augmenter la disponibilité en augmentant le taux de réparation, par la mise en oeuvre de dispositifs de tests par exemple, que d'augmenter la fiabilité.

L'utilisation de redondance ne s'impose vraiment que si la continuité de service est indispensable au maintien de la sécurité. Les redondances ne sont pas indispensables en général, lorsqu'un fonctionnement de sécurité, directement accessible, peut être défini.

Par contre, l'introduction de mécanismes de tests en ligne destinés à détecter les défaillances est souvent nécessaire pour améliorer la sécurité. Ces mécanismes de détection permettent de réagir en forçant l'automatisme dans son fonctionnement de sécurité, s'il n'y a pas de redondance, ou en reconfigurant le système, dans le cas contraire. Les mécanismes d'autotests des unités de traitement sont relativement classiques. Par contre, pour le test des interfaces de l'automate et des éléments hors automate, les solutions généralement proposées passent par une duplication de ces éléments. Or, il s'avère que ces composants sont, de part leur nombre, les plus pénalisants pour la fiabilité de l'ensemble. La mise en place d'un dispositif de test de ces éléments permet un meilleur compromis sécurité / fiabilité que la duplication, si le taux de pannes de ce dispositif est inférieur à celui des organes sous test.

La suite de notre étude porte sur la définition et l'analyse des performances d'un mécanisme de test des éléments hors automate, évitant la duplication de ceux-ci. De plus, la partie opérative proprement dite (c'est-à-dire la partie mécanique de l'automatisme) n'est pas duplicable, en général.

DEUXIEME PARTIE

MODELISATION DE LA PARTIE OPERATIVE

ANALYSE SYNTAXIQUE ET SEMANTIQUE

INTRODUCTION DE LA DEUXIEME PARTIE

La séquence de C.R. étant vue comme une suite de symboles ou éléments d'un alphabet, il est légitime de tenter de dégager des règles d'association de ceux-ci. L'ensemble de ces règles peut être considéré comme une grammaire. Vérifier que la séquence de C.R. observée peut être obtenue par l'utilisation stricte d'un ensemble de règles correspond à une analyse de type syntaxique.
A la P.O. générant la séquence de C.R. est alors associée une grammaire. Toute défaillance introduisant une règle nouvelle, donc non comprise dans la grammaire, sera reconnue dès que cette règle sera utilisée.

Il est clair qu'une phrase syntaxiquement correcte peut ne pas avoir de sens. L'étude du signifié d'une phrase relève de la sémantique. L'analyse sémantique nécessite la prise en compte du contexte dans lequel s'insère la phrase étudiée. La connaissance des ordres reçus et des perturbations admissibles subies par la P.O. forme pour celle-ci un contexte qui peut permettre d'accepter ou de refuser certaines séquences de C.R. syntaxiquement correctes. Une telle approche s'apparente à une analyse sémantique de la séquence observée.

Dans cette deuxième partie de l'étude, nous voyons comment il est possible de traduire le comportement d'une P.O. saine par une grammaire. Selon l'importance de la prise en compte du contexte, nous obtenons différents niveaux d'analyses correspondant à des modèles de plus en plus complexes.
Comme pour tout mécanisme de test, l'objectif à atteindre est l'obtention du meilleur compromis entre la signalisation de fausses pannes et le masquage de vrais défauts.

CHAPITRE V

APPLICATION DE L'ANALYSE SYNTAXIQUE AU TEST EN LIGNE DES AUTOMATISMES

Les comptes rendus (C.R.) étant fournis par la P.O., toute défaillance de celle-ci qui entraîne à terme l'apparition d'une erreur de C.R., peut à priori être diagnostiquée.
Le test est alors basé sur l'analyse de la séquence de compte rendu.

Présentation des bases de l'analyse

Soit R l'alphabet de compte rendu de la P.O.. Soumise à un cycle de travail, cette dernière génère une séquence de compte rendu $r(1,n) \in R^+$.
Si le cycle de travail est répétitif, nous pouvons admettre qu'il existe une séquence de C.R. de référence notée $r^*(1,n^*)$ correspondant à ce que génére la P.O. en l'absence de défaillance.
Comparer la séquence de référence $r^*(1,n^*)$ à celle $r(1,n)$, qui est générée doit permettre la détection d'erreurs. Dans le cadre du test en ligne, cette comparaison doit être dynamique. L'automatisme n'étant pas un système strictement répétitif (influence des perturbations, parallélismes), il est nécessaire de tolérer certaines différences entre les deux séquences. Ceci correspond à ce qui est habituellement appelé la distance d'édition. Une quantification de cette distance peut constituer la base d'une méthode de test tolérante. Ceci est couramment utilisé en reconnaissance des formes. De nombreux algorithmes d'évaluation sont connus |WAG-74|, |LEV-66:, |SAK-78|, |RAB-78|.
Par principe, ces méthodes sont mal adaptées au test en ligne à cause du temps de réponse qu'elles nécessitent. Nous préférons nous orienter vers des méthodes d'analyse syntaxique. De plus, nous étudierons dans quelle mesure il est possible de s'affranchir du cycle de travail répétitif qui constitue une restriction importante du champ d'application.
Nous empruntons à |MIC-84| les définitions suivantes.

Nous faisons également apparaître les relations qui existent entre les notions de langage et celles déjà utilisées dans l'approche des machines séquentielles.

5.1 TERMINOLOGIE : LANGAGE ET GRAMMAIRE

On appelle grammaire le quadruplet G = (X,V,S,P)

où

X est l'alphabet où sont écrites les phrases du langage,

V est un alphabet auxiliaire ou non terminal,

S est un élément de V appelé axiome,

P est un ensemble de règles de production de la forme.

$\alpha \rightarrow \beta$ avec $\alpha \in (V \cup X)^* V (V \cup X)^*$

et $\beta \in (V \cup X)^*$

5.1.1 GENERATION D'UNE PHRASE PAR UNE GRAMMAIRE

Soit $\nu \in (V \cup X)^*$. On dit que ν est réécrite en σ par la règle ($\alpha \rightarrow \beta$) si

$\nu = \nu_1 \alpha \nu_2$

et que $\sigma = \nu_1 \beta \nu_2$.

La réécriture s'arrête lorsque l'on obtient une phrase d'éléments de X.

Nous avons alors une séquence $x(1,n) \in X^+$.

L'ensemble des phrases qui peuvent être générées par utilisation de la grammaire G forme un langage L(G).

Grammaire régulière

Une grammaire est dite régulière quand ses règles de production sont de la forme

$A \rightarrow aB$ avec $A,B \in V$; $a \in X$

ou $A \rightarrow a$

Un tel langage est dit régulier.

Automates finis

Une grammaire régulière peut être représentée par un automate d'états finis.

On appelle automate fini un quintuplet

$\mathcal{A} = (X, Q, \delta, qo, T)$

où X est un alphabet (d'éléments terminaux),

Q un ensemble fini d'états (correspondant à l'alphabet auxiliaire),

qo $\in$ Q l'état initial,

T $\subset$ Q l'ensemble des états finals,

S: Q$_*$X $\rightarrow$ Q la fonction de transition.

Représentation graphique

Chaque élément q $\in$ Q est représenté par: (q) et chaque transition de δ (notée: (qi,x,qj) où x $\in$ X; qi,qj $\in$ Q) est symbolisée par un arc étiqueté: $\xrightarrow{x}$;

qo est repéré par: $\Longrightarrow$ (qo) ;

qf $\in$ T est symbolisé par: (qf) $\Longrightarrow$.

Automate non déterministe

Un automate est non déterministe s'il existe deux arcs (q,x,qj), (q,x,qk) de même origine q, de même étiquette x $\in$ X, d'extrémité distincte qj,qk $\in$ Q; qj $\neq$ qk.

Nous appelons G = (X,Q, δ) le graphe associé à $\mathcal{A}$.

Chemin dans un graphe

C'est une séquence d'arcs α (1 ,n) prise dans G tel que si α ij = (qi,xij,qj) et α pk = (qp,xpk,qk) sont de rang consécutif dans la séquence, alors qj = qp.

La séquence x(1,n) d'étiquettes correspondant à un chemin est appelée séquence générée par le chemin.

Un chemin d'origine qi, d'extrémité qj générant une séquence x(1, n) sera noté (qi, x(1,n) , qj).

Les arcs sont parcourus exclusivement dans le sens de la flèche, un chemin est donc orienté.

Phrase générée par l'automate

C'est la séquence générée par un chemin d'origine qo et d'extrémité qf $\in$ T.

L'ensemble des phrases générées par un automate $\mathcal{A}$ constitue un langage L($\mathcal{A}$).

Automates dérivés

Soit $\mathcal{A}$ = (X,Q, δ ,qo,T) un automate fini et P = (P1,...,Pr) une partition de Q.

On appelle automate dérivé de $\mathcal{A}$ pour la partition P l'automate fini $\mathcal{A}' = (X, P, \gamma, Po, R)$

où a) $Po \in P$ est tel que $qo \in Po$,

b) $R = \{ Pi \in P \mid \exists\ qj \in T\ ,\ qj \in Pi \}$

c) $(Pi\ a\ Pj) \in \gamma$ si $\exists\ qk \in Pi$ et $qe \in Pj$ tels que $(qk\ a\ qe) \in \delta$

$\mathcal{A}'$ est construit par fusionnement de certains éléments de Q. C'est la base des méthodes d'inférence.

Propriétés

Soit $\mathcal{A}$ un automate fini et $\mathcal{A}'$ un automate dérivé de $\mathcal{A}$. Pour une partition quelconque de Q, on montre que

$L(\mathcal{A}) \subset L(\mathcal{A}')$.

Cette propriété stipule donc que certaines phrases acceptées par $\mathcal{A}'$ sont rejetées par $\mathcal{A}$.

5.1.2 TEST DE LA P.O. PAR ANALYSE SYNTAXIQUE

Soit Rj l'alphabet de C.R. associé au capteur j et R l'alphabet obtenu en faisant le produit cartésien des Rj étendu aux n capteurs de la P.O..

Compte tenu de l'état de la P.O. à un instant donné, nous admettons qu'il existe un sous ensemble $U \subset R$ de C.R. acceptable à l'instant suivant.

En faisant l'hypothèse à priori que U dépend de l'état considéré de la P.O., nous pouvons admettre l'existence d'un alphabet auxiliaire Q. Dans ce cas, si $qj \in Q$ est l'état envisagé et si $ri \in U$, nous obtenons des règles de production qui s'écrivent:

$qj \longrightarrow ri\ qk \quad | \quad ri \in U;$

$qj \longrightarrow rk \quad | \quad rk \in U.$

Ceci sous entend que l'apparition de rk à partir de qj conduit à un élément de T. C'est une finale.

Cette approche intuitive amène à considérer que toute séquence de C.R. peut être générée à partir d'une grammaire régulière. L'automate correspondant peut alors être considéré comme un modèle de la P.O.. Toute phrase acceptée par le modèle est révélateur de l'existence d'une défaillance.

Remarque

Il est important que le langage accepté par le modèle soit identique à celui qui est accepté par la P.O.. Ceci impose des contraintes dans la définition, l'exploitation et la caractérisation du modèle.
Soit $\mathcal{A}'$ l'automate utilisé comme modèle et $\mathcal{A}$ celui qui correspond à la P.O.. Si $\mathcal{A}'$ est un automate dérivé de $\mathcal{A}$ (obtenu par inférence grammaticale à partir d'une séquence échantillon par exemple), alors la propriété $L(\mathcal{A}) \subset L(\mathcal{A}')$ entraîne la signalisation de fausses pannes. Ceci peut conduire à ce que nous étudierons sous le vocable "décidabilité du test".
Il est clair que l'inclusion inverse ($L(\mathcal{A}) \supset L(\mathcal{A}')$) n'est guère plus favorable car elle conduit au phénomène de masquage.
L'objet de ce chapitre est la définition du modèle de la P.O.

5.2 MODELISATION DE LA P.O. EN DEHORS DU CONTEXTE DE LA COMMANDE

Dans ce paragraphe, nous envisageons l'utilisation d'un modèle (automate fini) qui permette une validation des C.R. émis par la P.O. indépendamment du cycle de travail.

5.2.1 ANALYSE DE LA P.O.

5.2.1.1 Trajectoire, trajectoire singulière

Dans un système automatisé, certaines grandeurs physiques sont mesurées par un capteur (généralement analogique dans ce cas) ou par une association de capteurs (généralement de type tout-ou-rien).

Trajectoire

Nous appelons trajectoire l'ensemble ordonné des valeurs que peut prendre une grandeur physique lors d'une évolution monotone entre deux extrémums selon un degré de liberté.

Remarque

L'ordre est choisi arbitrairement. Une évolution dans le sens croissant correspond à une vitesse positive.

Nous considérons que les grandeurs mesurées sont discrétisées soit par les capteurs, soit par l'utilisation de convertisseurs A/D.
L'ensemble des valeurs prises par une grandeur physique mesurée, vue de la P.C., est de dimension finie.

Exemples

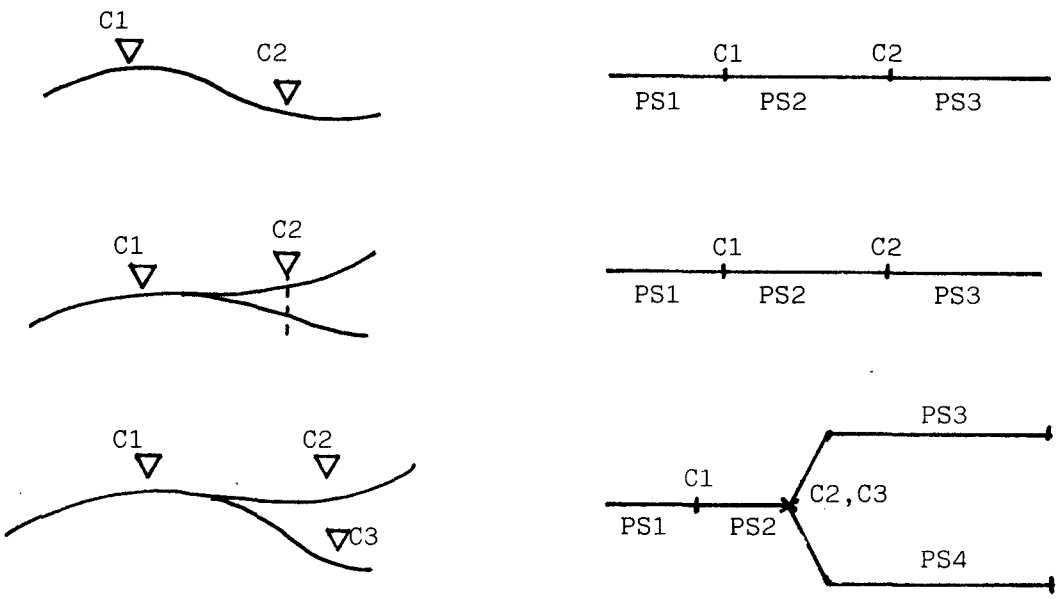
- La température d'un autoclave est évaluée à partir d'une mesure de courant dans un circuit électrique,incluant une résistance au platine en contact thermique avec le bain.
  Un convertisseur A/D 8 bits permet une discrétisation de la température en 256 valeurs distinctes.
- Une came se déplaçant entre deux fins de course tout ou rien permet une évaluation de la position de cette came en 3 valeurs distinctes.

Point singulier d'une trajectoire (noté PS)

Nous appelons point singulier d'une trajectoire l'ensemble des points qui peuvent être atteints sans modification de C.R. lors d'une évolution de la grandeur mesurée.

Remarque

L'ensemble ordonné des points singuliers forme une trajectoire singulière qui est une discrétisation de la trajectoire réelle.
Une telle trajectoire a une dimension finie par définition.
La figure 5.1 illustre la règle adoptée notamment lors des embranchements.
Les Ci représentent les points de trajectoire pour lesquels il y a modification des comptes-rendus.



- figure 5.1 -

5.2.1.2 Alphabet associé aux trajectoires

Soit k le nombre de grandeurs mesurées d'une P.O.. Considérons la jème grandeur et notons $\mathcal{T}j$ sa trajectoire.

Soient n le nombre de P.S. de $\mathcal{T}j$ et

si un symbole associé au ième élément de $\mathcal{T}j$.

Soit dj un symbole traduisant une défaillance affectant la grandeur ou sa mesure.

Nous définissons un alphabet Sj relatif à la grandeur j par:

$Sj = \{ s1, s2, \ldots si, \ldots sn, dj \}$

Pour l'ensemble de la P.O., nous définissons un alphabet S en effectuant le produit cartésien des k alphabets Sj

$$S = \bigtimes_{j=1}^{k} Sj$$

5.2.1.3 Codage des trajectoires

Soient une P.O. de k grandeurs mesurées et

n le nombre de variables binaires utilisées pour coder la valeur de la jème de ces grandeurs.

Notons $Rj = B^n$ l'alphabet de C.R. relatif à cette grandeur.

Le codage établit une correspondance C entre les éléments de Sj et ceux de Rj.

$C: Sj \longrightarrow Rj$

Ayant défini l'alphabet R par $R = \bigtimes_{j=1}^{k} Rj$, nous avons donc une correspondance C telle que:

$C: S \longrightarrow R$.

Nous noterons $R' \subset R$ l'ensemble des symboles de R qui ne sont associés à aucun élément de S.

Remarque:

Une image de dj par C est un élément de RJ. L'application Cj à ce niveau sera spécifiée selon le modèle utilisé.

5.2.2 MODELISATION DE LA PARTIE OPERATIVE

5.2.2.1 Modélisation des trajectoires

Etat de la P.O. par rapport aux trajectoires

Nous définissons l'état d'une grandeur mesurée, à un instant donné, par le P.S. qui contient la valeur de cette grandeur à l'instant considéré.

Aux nj états possibles de la jème grandeur, nous ajoutons un état initial qjo. Cet état représente le point d'entrée dans le modèle; il signifie que dans certaines circonstances (notamment à la mise sous tension), la valeur de la jème grandeur est inconnue.

Nous ajoutons également un état qjf accessible à la suite d'une défaillance affectant la grandeur j. Notons Qj l'ensemble de ces états.

Transitions entre états

Deux P.S. sont dits contigus si le passage de l'un à l'autre est possible. Les transitions entre états représentent le passage d'un P.S. à un autre qui lui est contigu.

Les transitions entre états sont donc définies par la géométrie des trajectoires ainsi que par les interactions entre les trajectoires.

Les états qjo et qjf sont des P.S. fictifs.

En l'absence de toute interaction avec une autre trajectoire, une trajectoire est alors représentée par un automate.

$\mathcal{A}_j = (Sj, Qj, \delta j, qjo, qjf)$

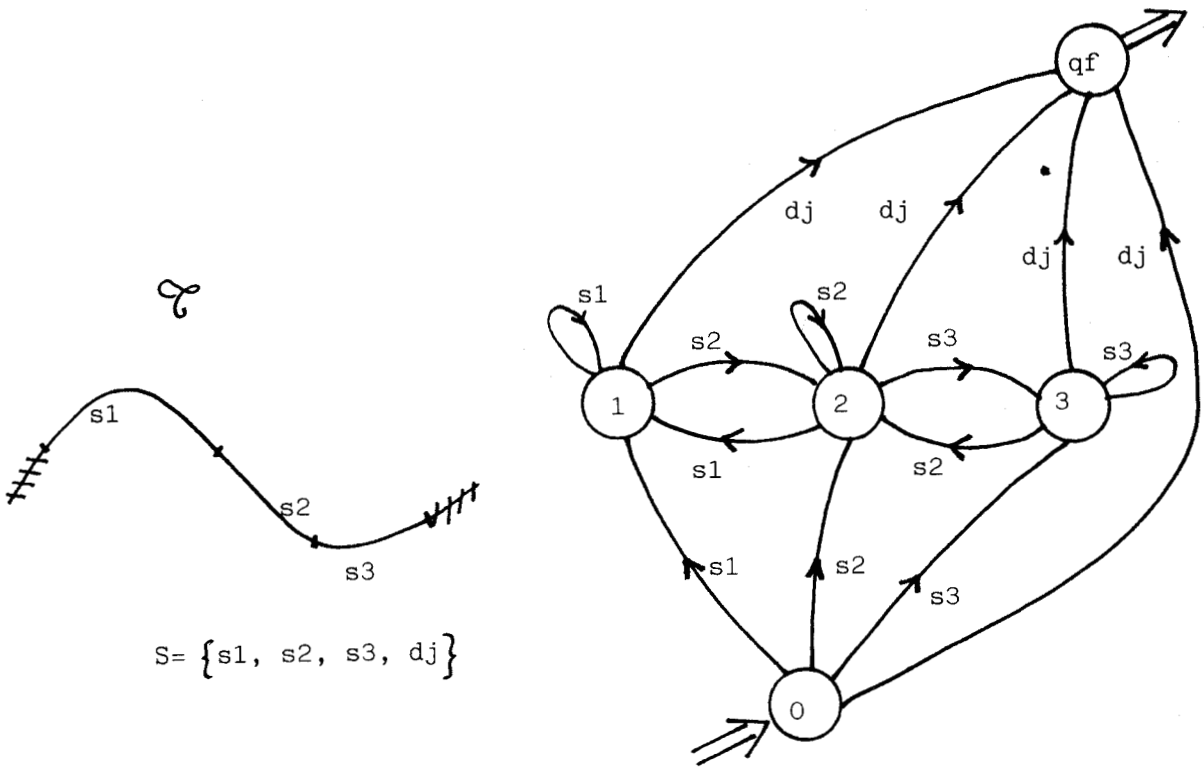
où Sj est l'alphabet dont les éléments sont associés aux arcs (étiquettes)

Qj est l'ensemble des états possibles assimilés aux P.S.,

δj un ensemble d'arcs étiquetés réalisant une application de

$Sj * Qj \longrightarrow Qj$

La figure 5.2 donne l'exemple d'un tel automate.



- figure 5.2 -

5.2.2.2 Modélisation des C.R. issus de la P.O.

L'état de la P.O. est représenté par l'ensemble des valeurs prises par les n grandeurs mesurées de la P.O..

L'évolution de l'état de cette P.O. peut alors être représentée par un graphe dont les sommets sont obtenus en faisant le produit cartésien des Qj. Par convention, nous éliminerons de Qj les sommets qui contiennent plus d'un qjf. L'ensemble des éléments finaux (noté T) comprend donc n représentants.

Cette convention repose sur les avis suivants:

- L'arrivée dans un état appartenant à T correspond à une défaillance détectée. La mission de surveillance est terminée. Il faut maintenant faire intervenir le mécanisme de réaction ou le réparateur.
- En évitant le fusionnement des éléments finaux, nous facilitons la localisation de la défaillance.
- Il est toujours possible de réduire le graphe à la suite d'une défaillance détectée en j par élimination des éléments correspondants à cette grandeur. Le test des parties saines de la P.O. peut ainsi être maintenu.

La P.O. est alors modélisée par un automate

$\mathcal{A} = (S, Q, \delta, qo, T)$

où S est l'alphabet des trajectoires,

Q l'ensemble des sommets précédemments définis,

qo le sommet initial correspondant à $\bigtimes_{j=1}^{k} qjo$

T l'ensemble des sommets terminaux,

δ un ensemble d'arcs étiquetés, construits en tenant compte de la géométrie des trajectoires et des éventuelles interactions entre les grandeurs mesurées.

Remarque

Une P.O. peut généralement être décomposée en sous P.O.. En effet, il existe des grandeurs mesurées qui sont indépendantes. Cette constatation permet de limiter l'explosion combinatoire liée au produit cartésien des ensembles de sommets. Au niveau réalisation, nous envisagerons de prendre en compte certaines interactions entre trajectoires par des moyens spécifiques, de façon à séparer artificiellement les modéles. Dans la suite de l'étude, nous appelons P.O., toute sous P.O. qui peut être modélisée par un automate.

Introduction du codage

Il a été défini une application de S dans R appelée codage. A partir de l'automate $\mathcal{A} = (S, Q, \delta, qo, T)$, il est possible de construire l'automate $\mathcal{A}' = (R, Q, \delta', qo, T)$ tel que:

- à tout arc (q1,s,q2) de δ , correspond un arc (q1,r,q2) pour lequel C: s $\longrightarrow$ r $\in$ R.
- seul le codage de dj n'est pas défini.

Hypothèse de défaillance

Toute modification de C.R. compatible avec l'état antérieur de la P.O. est considérée comme étant le résultat d'une évolution normale. Il est clair que ce principe entraîne la non détection de certaines défaillances comme nous le verrons ultérieurement.

Remarque

Soit n le nombre d'éléments de Rj.

Si (qi,s,qf)$\in \delta$ est tel que s dépende de dj (s= s1*...*dj*..*sn), il est clair que,dans ces conditions, qf $\in$ T. Compte tenu de la constitution de T, nous pouvons affirmer qu'il n'existe pas de rang m≠j tel que sm = dm.

Codage de dj

Le codage de dj dans (qi,s,qf) est opéré de la façon suivante.

Soient (qi,s1,q1), (qi,s2,q2), ... , (qi,sk,qk); s1, s2, ... ,sk $\in$ S, l'ensemble des arcs de δ ,de même origine qi, dont les étiquettes différent uniquement par le jème élément.

Si parmi ces arcs il en existe un, tel que (qi,sf,qf) pour lequel le jème élément de sf est dj, alors il est créé autant d'arcs dans δ', que nécessaire pour satisfaire à la propriété suivante.

- Soit an = r1* ... *rjn* ... *rk, l'étiquette d'un arc αif= (qi,an,qf) créé à partir de (qi,sf,qf) envisagé ci-dessus.
- Soient p le nombre des arcs tels que α if,

  Ej l'ensemble des étiquettes {r1}*{r2}*...*Rj*...*{rk}

  où {ri} $\supset$ Ri est un sous emsemble de Ri formé d'un seul élément

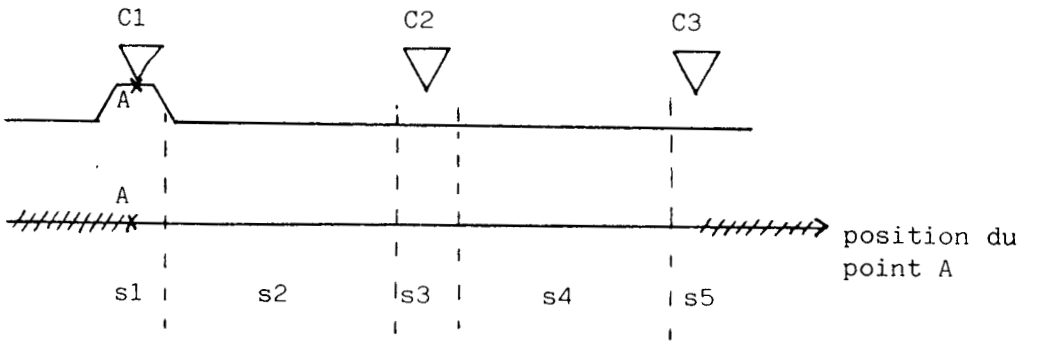
  $\bigcup_{n=1}^{p}$ an = Ej.

Ce codage satisfait à l'hypothèse de défaillance envisagée ci-dessus.

Exemple

Soit une P.O. réduite à une came se déplaçant devant trois capteurs. Compte tenu de la forme de la came, nous dénombrons cinq P.S. (figure 5.3a), le tableau de la figure 5.3b donne la correspondance par codage.

La figure 5.4a donne le modéle de la trajectoire (automate $\mathcal{A}$). Nos conventions de codage des dj conduisent à l'automate $\mathcal{A}'$ de la figur 5.4b.



- a -

S	R = C1	x C2	x C3
s1	1	0	0
s2	0	0	0
s3	0	1	0
s4	0	0	0
s5	0	0	1

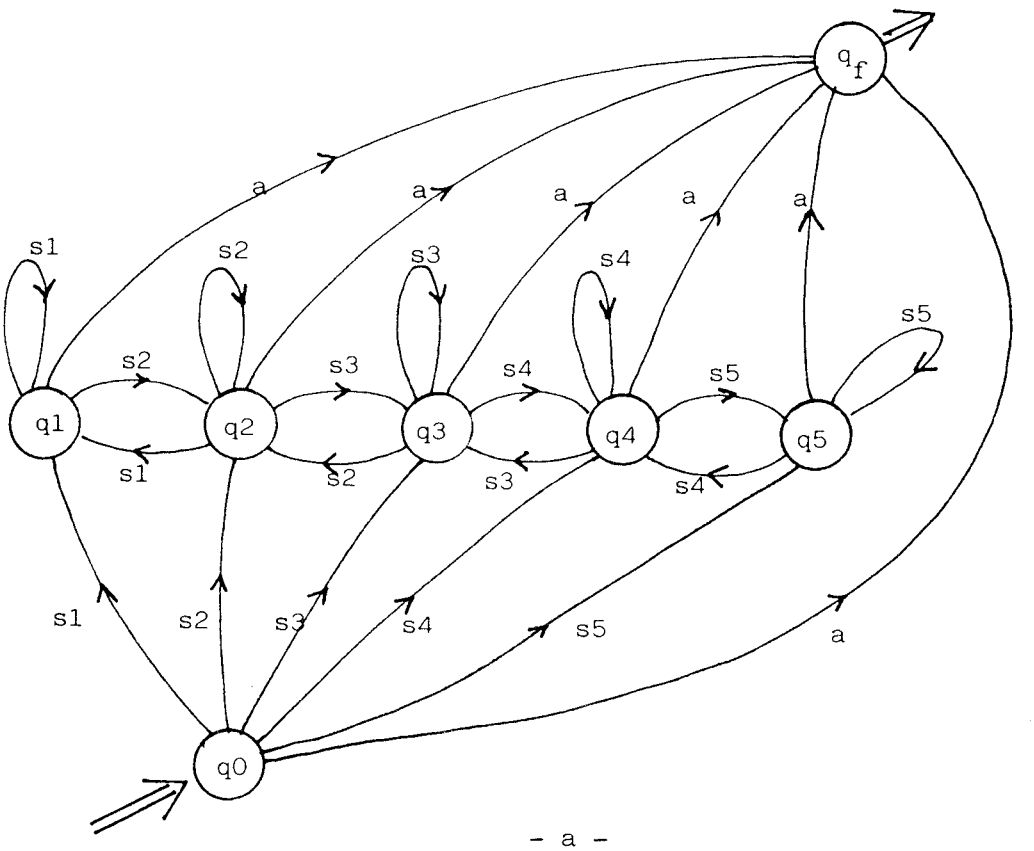
- b -

-figure 5.3 -

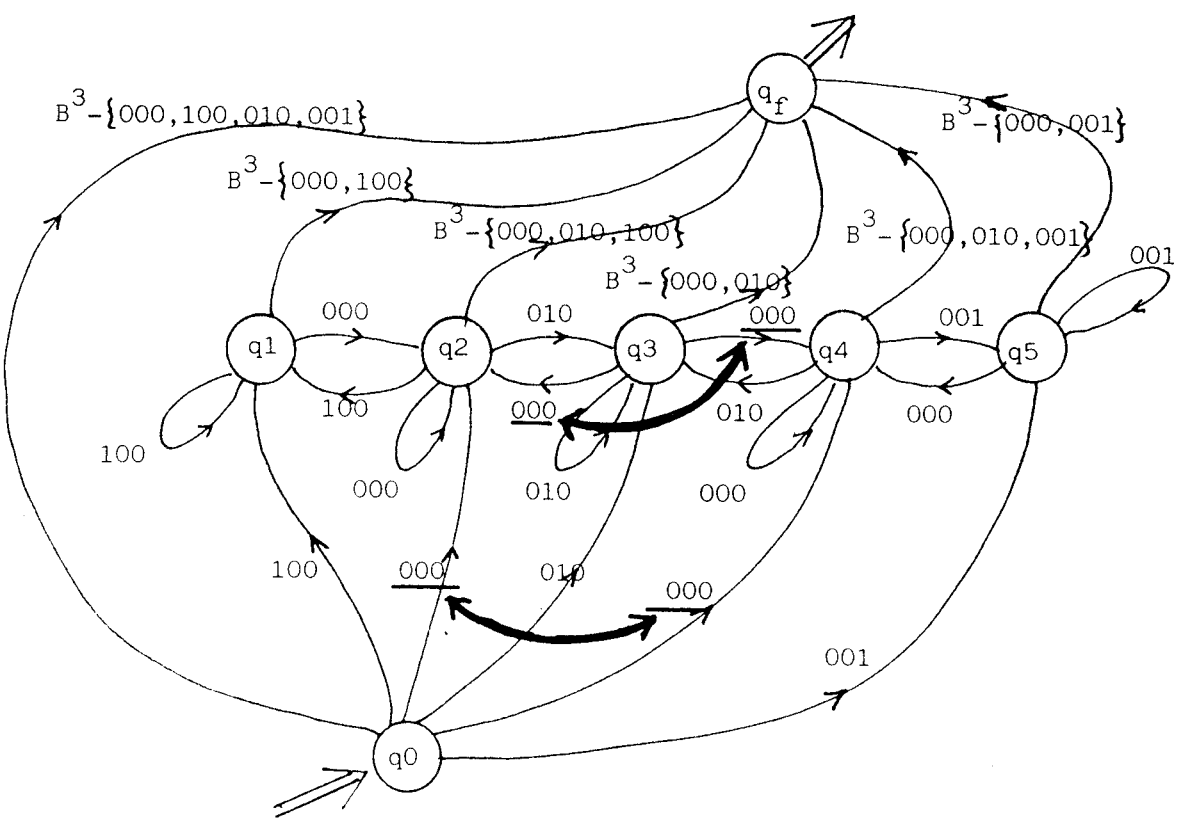
5.2.3 INFLUENCE DU CODAGE

Proposition

Le graphe G=(S, Q, δ) est déterministe, c'est-à-dire que s'il existe deux arcs (q1,s1,q2), (q1 ,s2,q3)$\in \delta$, tels que q1, q2, q3 $\in$ Q; s1, s2 $\in$ S; q2 $\neq$ q3 alors s1 $\neq$ s2.


q_f
a
s1
s2
s3
s4
s5
q1
q2
q3
q4
q5
q0

- a -


q_f
$B^3-\{000,100,010,001\}$
$B^3-\{000,100\}$
$B^3-\{000,010,100\}$
$B^3-\{000,010\}$
$B^3-\{000,010,001\}$
$B^3-\{000,001\}$
001
000
010
100
q1
q2
q3
q4
q5
q0

- b -

- figure 5.4 -

Cette proposition découle de la définition des points singuliers (§ V-1.2). Si s1 = s2, les points singuliers correspondants sont confondus, ce qui entraîne q2 = q3.

Proposition

Si le codage établit une correspondance univoque entre S et R, l'automate $\mathcal{A}'$ peut ne pas être déterministe.

Soit sa, sb $\in$ S et r $\in$ R tels que:

C : sa $\longrightarrow$ r

C : sb $\longrightarrow$ r

Il existe donc des arcs d'étiquettes distinctes (sa, sb) dans δ qui ont même étiquette (r) dans δ'.

Si, de plus, deux au moins de ces arcs ont même sommet origine, l'automate $\mathcal{A}'$ n'est plus déterministe.

Condition nécessaire et suffisante pour que l'automate reste déterministe

Condition suffisante:

Pour que l'automate soit déterministe, il faut que chaque élément de R soit l'image d'un élément de S au plus.

Cette condition est aussi nécessaire.

En effet, s'il existe au moins deux éléments sa et sb $\in$ S tels que

$\exists$ r $\in$ R, C : sa $\longrightarrow$ r

C: sb $\longrightarrow$ r

alors, il existe au moins deux arcs de δ' ayant comme origine qo et comme étiquette r.

Proposition

La condition nécessaire et suffisante pour que l'automate qui modèlise la P.O. reste déterministe, après codage, est que chaque élément de R soit l'image d'un élément de S au plus.

Remarque

Par contre, l'affectation à un même élément de S de plusieurs éléments de R,ne rend pas l'automate non déterministe si la condition ci-dessus est satisfaite.

5.3 PRINCIPE DU TEST PAR OBSERVATION DES COMPTES RENDUS

5.3.1 SEQUENCE GENEREE PAR LA P.O.

Notons Δt la période d'échantillonnage de l'observateur.
Sur une période d'observation T, la P.O. génére une séquence r(1,n) de longueur n = partie entière de T /Δt, sur l'alphabet R.
Dans le test temps réel, cette période d'observation est une période de travail de l'automatisme.
Si la séquence observée est liée à l'évolution imposée à la P.O. par la P.C., il est clair que notre modèle est valable quelque soit le cycle imposé à la P.O..

Séquence acceptable

Soit une séquence de C.R., r(1,n) $\in R^n$ fournie par la P.O.. Cette séquence est acceptable si elle peut être obtenue pour un chemin issu de l'état initial qo ne conduisant pas à un état final q $\in$ T.
Cette séquence est donc acceptable si elle ne fait pas partie du langage L($\mathcal{A}$').
En général, on considère qu'une phrase est acceptée si elle fait partie du langage défini par l'automate. Notre appellation est en contradiction avec les termes habituels, mais elle est plus significative dans le cas du test.
Notre espoir est en fait, d'obtenir une séquence, la plus longue possible, traduisant ainsi la fiabilité de l'automatisme.

5.3.2 INFLUENCE D'UNE DEFAILLANCE

La P.C. fixe le chemin que doit emprunter la P.O..
Constater que la séquence observée est non acceptable pour la P.O. considérée, c'est détecter une défaillance de celle-ci.
Notons r*(1,n) la séquence que doit générer la P.O. saine, compte tenu du chemin imposé par la P.C.
et r(1,n) la séquence observée.

Substitution dans une séquence

Nous appelons substitution du kème élément d'une séquence de longueur n $\geqslant$ k sur l'alphabet X, le remplacement de x(k) $\in$ X par un élément x'(k) $\in$ X tel que x'(k) $\neq$ x(k).

Si une défaillance de la P.O. engendre une erreur de C.R., alors la séquence r(1,n) est obtenue par un nombre fini de substitutions dans r*(1,n).

Défaillance détectable

Pour qu'une défaillance soit détectable, il faut:

- qu'elle engendre au moins une substitution appelée erreur de C.R.
- que la substitution appliquée à $r^*(1,n)$ donne une séquence non acceptable.

Si la première condition n'est pas satisfaite, la défaillance n'est pas détectable avec le principe retenu.

Si la substitution ne conduit pas à une séquence non acceptable, l'erreur est latente.

Parmi les défaillances qui introduisent des erreurs de C.R. et qui ne provoquent pas la génération de séquences non acceptables, nous trouvons:

- les évolutions différentes de celles attendues,
- une variation notable de la vitesse d'évolution sur la trajectoire pouvant aller jusqu'au blocage,
- un retard ou une anticipation de C.R.

Les deux premiers points correspondent à des défaillances affectant les préactionneurs, les actionneurs, la P.O., les liaisons électriques, fluidiques ou mécaniques entre ces éléments.

Le troisième événement est lié alors à une défaillance de capteur ou de couplage de celui-ci à la P.C.

Une telle erreur est non révélable avec le mécanisme de test mis en jeu; elle est masquée.

De plus, le principe avancé, pour être retenu, doit permettre une décision non ambiguë.

Règle

Le test de la P.O. par recherche de séquence non acceptable est décidable s'il est impossible de trouver deux chemins distincts, de même longueur n, issus de l'état initial, générant deux séquences identiques dont une seule soit acceptable.

Proposition

Dans un automate fini déterministe, il est impossible de trouver deux chemins de même longueur, issus du sommet initial, générant deux séquences identiques et conduisant à des sommets différents.

Soit deux chemins (qo, r(1 , n), q)

et (qo, r(1 , n), q')

de même origine qo, générant la même séquence r(1, n) conduisant à des états différents q et q'.

Soit un entier k; $1 \leqslant k < n$ tel que le chemin (qo, r(1, n), qn) puisse être décomposé en deux chemins (qo, r(1, k), qk)(qk, r(k+1, n), q) et (qo, r(1, n), q') soit représenté par (qo, r(1, k), q'k)(q'k, r(k+1, n), q').

Puisque $q \neq q'$, $\exists$ k tel que qk = q'k et tel que $\exists$ deux arcs (qk, r(k), qk+1), (qk, r'(k), q'k+1) pour lesquels $qk+1 \neq q'k+1$.
Comme la séquence est la même, r(k) = r'(k).

L'existence de ces deux arcs donne un automate non déterministe.

Proposition

Si l'automate est déterministe, le test de la P.O. à partir des séquences de C.R. non acceptables est décidable.
D'après la proposition précédente, il est en effet impossible de trouver deux chemins issus du sommet qo générant deux séquences identiques conduisant à un sommet $q \in T$ d'une part, et $q' \notin T$ d'autre part.

Remarque

Cette propriété des automates déterministes est souvent énoncée sous une forme voisine. Une séquence est dite ambiguë si elle peut être générée par deux chemins différents de même origine. La propriété s'exprime alors de la façon suivante:
aucune séquence n'est ambiguë pour un automate déterministe.
Nous avons vu que le codage peut conduire à un automate non déterministe.
L'étude de la décidabilité du test doit donc être approfondie.

5.3.3 ETUDE DE LA DECIDABILITE DU TEST

Il a été démontré, notamment dans |FU -75| que tout langage $L(\mathcal{A})$ accepté par un automate fini non déterministe $\mathcal{A}$ peut être accepté par un automate fini déterministe $\mathcal{A}^1$. De par la définition d'une séquence non acceptable, une telle séquence est assimilable à une phrase du langage $L(\mathcal{A}^1)$. Le théorème ci-dessus peut donc être utilisé.

La P.O. peut être représentée par un automate fini déterministe:
$\mathcal{A}^1 = (R, Q^1, \delta^1, qo^1, T^1)$.

tel que $Q^1 \subset \mathcal{P}(Q)$ ensemble des parties de Q

$qo^1 = qo$

$T^1 = \mathcal{P}(T) \subset Q^1$ avec $\mathcal{P}(T) \cap T \neq \emptyset$

δ^1 un ensemble d'arcs étiquetés.

$\mathcal{A}^1$ est déduit de $\mathcal{A}'$ à partir de l'algorithme ci-dessous.

Notons Uj $\in Q^1$ le jème élément de Q^1

qi $\in$ Q le ième élément de Q

$\delta_{i,rp} \subset \delta'$ l'ensemble des arcs de δ' d'origine qi et de même étiquette rp

Soit $\mathcal{U}$ un ensemble de parties de Q.

a/ Initialiser la procédure en introduisant qo dans $\mathcal{U}$.

b1/ Prendre un élément Uj$\in \mathcal{U}$, l'introduire dans Q^1, l'éliminer de $\mathcal{U}$.

b2/ Pour tout élément qi représenté dans Uj et pour tout ensemble d'arcs $\delta_{i,rp}$ de même étiquette rp faire:

c1/ créer une partie de Q, notée Un, regroupant toutes les extrémités qk des arcs tels que (qi,rp,qk)$\in \delta_{i,rp}$;

c2/ introduire l'arc (Uj,rp,Un) dans δ^1;

c3/ si Un n'est ni dans $\mathcal{U}$, ni dans Q^1, alors introduire Un dans $\mathcal{U}$.

d/ Reprendre en b1 jusqu'à ce que $\mathcal{U}$ soit vide.

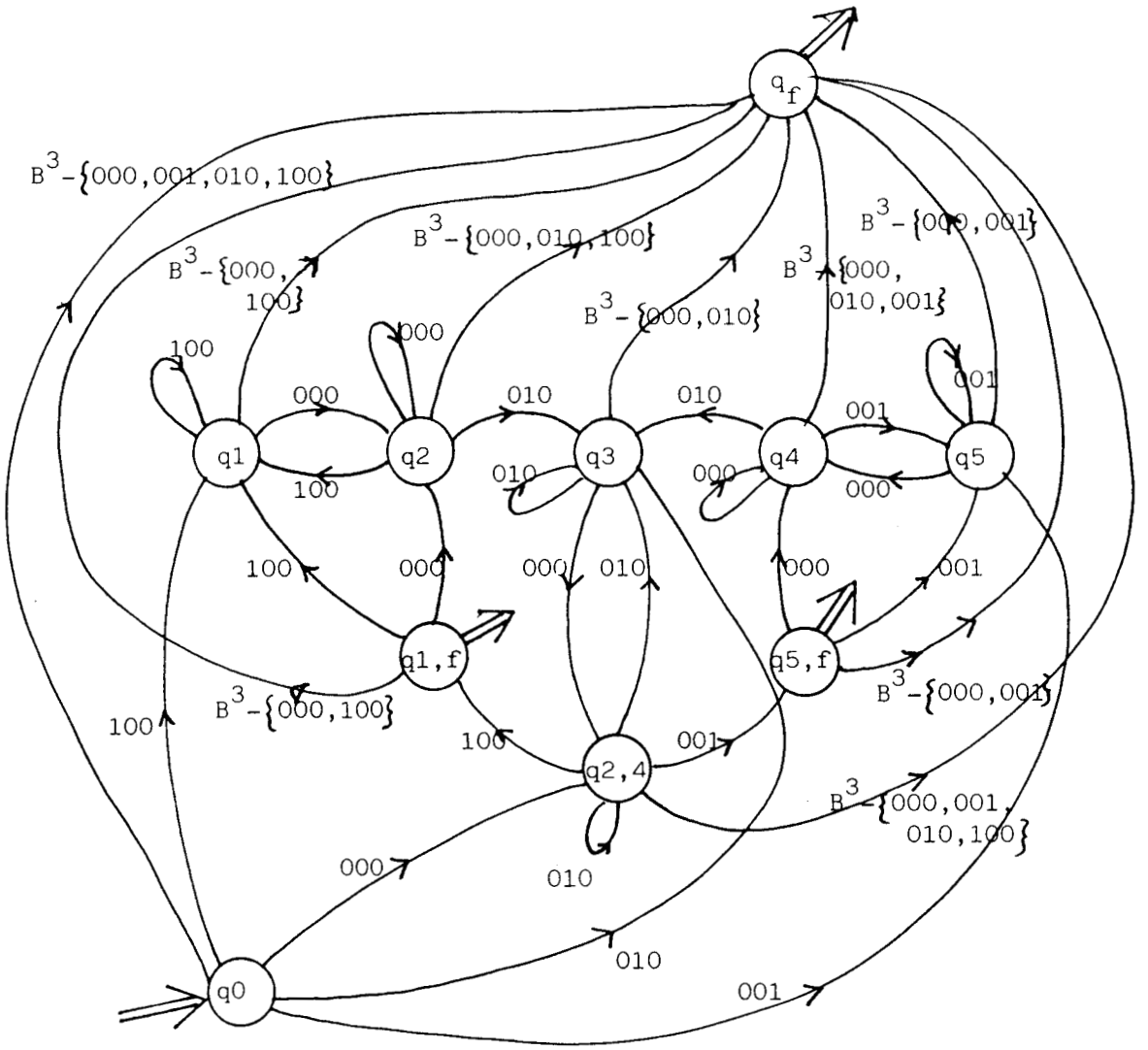
A titre d'exemple, cet algorithme est appliqué à l'automate $\mathcal{A}'$ défini fig 5.4b. Le résultat est présenté dans la table 5.5 qui conduit à la représentation de la figure 5.6.

Q^1	000	100	010	001	autres	liste des éléments de Q^1 non traités
						qo
qo	2,4	1	3	5	qf	2,4; 1; 3; 5; qf
2,4	2,4	1,qf	3	5,qf	qf	1; 3; 5; qf; 1,qf; 5,qf
1	2	1	qf	qf	qf	3; 5; qf; 1,qf; 5,qf; 2
3	2,4	qf	3	qf	qf	5; qf; 1,qf; 5,qf; 2
5	4	qf	qf	5	qf	qf; 1,qf; 5,qf; 2; 4
qf	qf	qf	qf	qf	qf	1,qf; 5,qf; 2; 4
1,qf	2	1	qf	qf	qf	1,qf; 5,qf; 2; 4
5,qf	4	qf	qf	5	qf	5,qf; 2; 4
2	2	1	3	qf	qf	2; 4
4	4	qf	3	5	qf	4

$Q^1 = \{$ qo; 1; 2; 3; 4; 5; 2,4; 1,qf; 5,qf; qf $\}$

$T^1 = \{$ 1,qf; 5,qf; qf $\}$

Fig. 5.5


q_f
$B^3-\{000,001,010,100\}$
$B^3-\{000,010,100\}$
$B^3-\{000,$
100$\}$
$B^3-\{000,010\}$
$B^3-\{000,001\}$
$B^3-\{000,$
010,001$\}$
100
000
010
010
001
001
q1
q2
q3
q4
q5
100
010
000
000
100
000
000
010
000
001
q1,f
q5,f
$B^3-\{000,100\}$
$B^3-\{000,001\}$
100
100
001
q2,4
$B^3-\{000,001,$
010,100$\}$
000
010
010
001
q0

- figure 5.6 -

Remarques

Il existe dans l'ensemble des parties de Q des éléments qui représentent à la fois des sommets de $\mathcal{A}'$ relatifs à une P.O. défaillante d'une part, et non défaillante d'autre part. Nous dirons d'un tel sommet qu'il est ambigu.

Dans l'exemple, q1,f et q5,f sont dans ce cas.

Il est interessant de noter que tout chemin passant par q3 passe obligatoirement par q2,4 puis q1,f (ou q5,f).Le problème n'existe pas seulement à l'initialisation.

Toute évolution vers un sommet ambigu ne permet pas de faire la distinction entre événement normal (P.O. saine) et anormal (défaillance). Une telle transition ne permet pas de décider de l'état de la P.O..

.3.4 EVALUATION QUALITATIVE DU TEST DE LA P.O. PAR LES C.R.

Pour rendre le test décidable, nous adoptons la règle suivante.

Règle

Nous constituons un ensemble de sommets finaux T^2 formé des éléments non ambigus de T^1.

La P.O. est alors modélisée par l'automate $\mathcal{A}^2 = (R, Q^1, qo, \delta^1, T^2)$

Notons T^3 l'ensemble des sommets tel que $T^2 \cup T^3 = T^1$, $T^2 \cap T^3 = \emptyset$

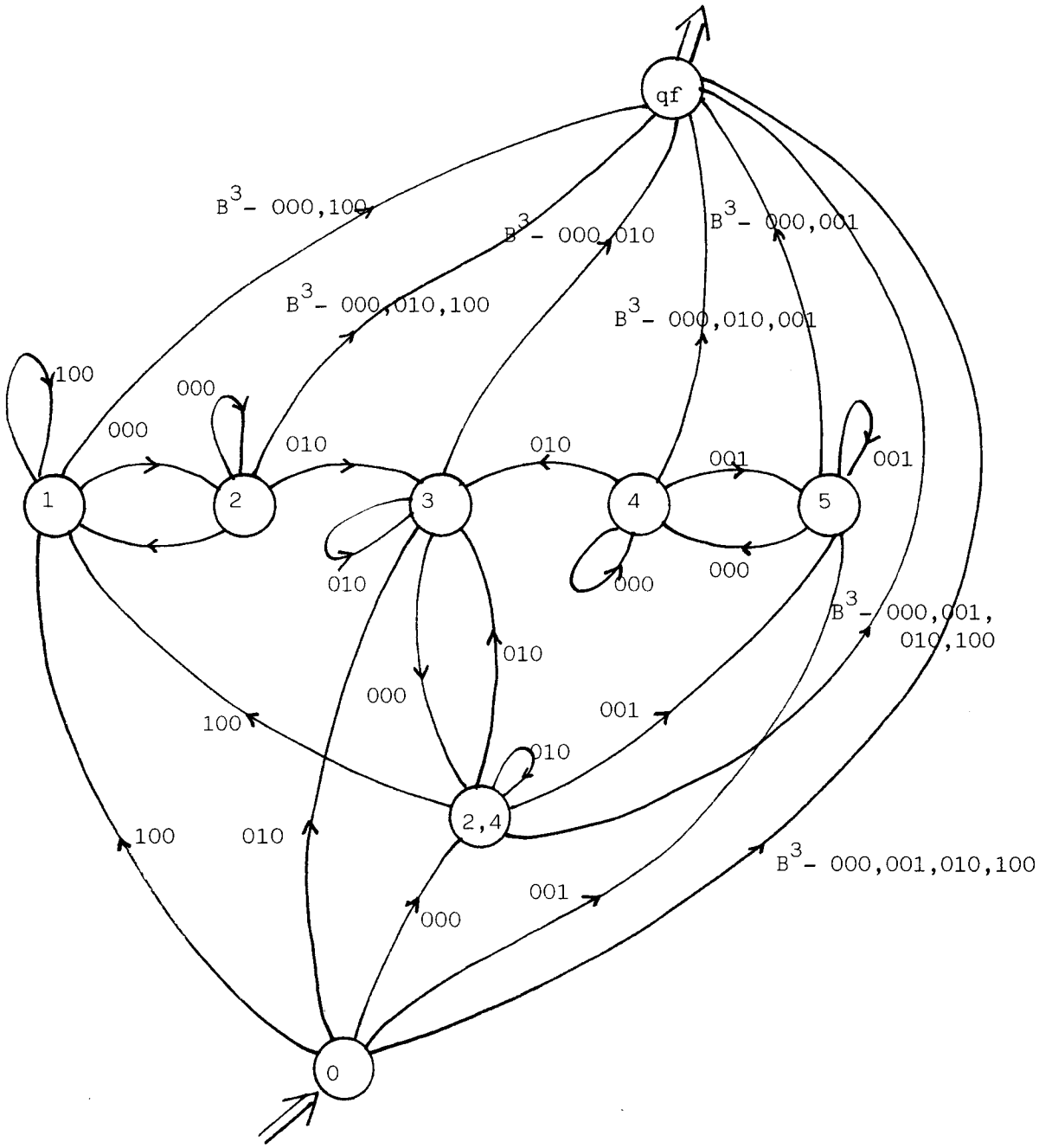
Ceci implique que seuls les chemins passant par un sommet non ambigu, correspondant à une P.O. contenant au moins une erreur,générent une séquence non acceptable.

Cette règle est justifiée si la probabilité d'atteindre un tel sommet sans défaillance est supérieure à celle d'y parvenir par erreur. Il va de soi qu'une telle erreur est alors non révélée. Si aucune hypothèse n'est faite sur la séquence de travail, notre règle est la plus réaliste. Pour une P.O. dont le cycle de travail est répétitif, il est possible d'imaginer que le rapport des probabilités soit inversé.

Dans notre exemple, l'application de cette règle donne $T^2 = \{qf\}$

Remarque

L'adoption de cette règle permet alors des fusionnements d'états ambigus avec des états non ambigus. C'est le cas des sommets q1 et q1f d'une part, q5 et q5f d'autre part dans le cas de la figure 5.6. Le modéle simplifié est représenté figure 5.7.


qf
B^3- 000,100
B^3- 000,010
B^3- 000,001
B^3- 000,010,100
B^3- 000,010,001
100
000
000
010
010
001
001
1
2
3
4
5
010
000
000
B^3- 000,001,
010,100
010
000
001
100
010
2,4
100
010
B^3- 000,001,010,100
001
000
0

- figure 5.7 -

Etudions le comportement du test face à une défaillance conduisant à une séquence non acceptable.

Temps de latence d'une erreur

Toute apparition d'une erreur implique une transition vers un sommet ambigu ou final.

Seul le passage par un sommet final révéle une erreur.

Soit une séquence non acceptable r(1, n) correspondant à un chemin

(qo, r(1, n)) , q); $q \in T^1$ pour laquelle

$\exists k$; $1 \leqslant k < n$ tel que (qo, r(1, k) , qk) $\longrightarrow$ $qk \in T^3$

Le temps de latence s'exprime alors par (n - k) Δt si Δt représente la période d'échantillonnage de l'observateur.

L'application de notre règle a donc pour premier résultat d'augmenter le temps de latence. Voyons son effet sur le masquage possible des erreurs.

Proposition

Une P.O. contenant une erreur consistante ne peut pas évoluer en dehors des chemins reliant les sommets de T^1.

Cette proposition découle de la définition des éléments de T^1.

Proposition

Le risque de masquage d'une erreur susceptible de générer une séquence non acceptable est nul s'il n'existe pas dans l'automate $\mathcal{A}^1$ de cycle joignant exclusivement des éléments de T^3 (cycle de sommets ambigus).

Cette proposition découle de la précédente. Si un tel cycle n'existe pas, alors tout chemin joignant des éléments de T^3 passe nécessairement par au moins un élément appartenant à T^2.

Cette propriété est vérifiée dans notre exemple (Fig 5.6). On y vérifie notamment qu'il n'y a pas de rebouclage de (1,qf) sur lui-même.

Nous avons mis en évidence un risque de masquage d'erreurs lié au codage des points singuliers. Ces masquages s'ajoutent à ceux vus précédemment (§ 3.2) qui sont inhérents à la méthode.

5.3.5 COMPARAISON ENTRE TEST STATIQUE ET TEST DYNAMIQUE DES C.R.

Le test statique consiste à vérifier à chaque instant que le C.R. observé r(k) appartient à l'ensemble des symboles de R utilisés pour le codage de S (§ 2.2).

Le test dynamique des C.R. consiste à vérifier la cohérence de l'évolution des C.R.. C'est le test que nous avons envisagé ici.
Le test dynamique inclut le test statique; il a donc des performances au moins égales. Il permet de plus d'éliminer certaines transitions entre éléments de R réellement utilisés par le codage.

Par exemple, dans l'automate considéré (Fig 5.6), en plus des éléments 110, 101, 011, 111 exclus par un test statique, le test dynamique permet de rejeter toute transition d'étiquette 010 en 001 à partir du sommet q1.
Dans les deux cas, ces procédures de test portent sur les défaillances des capteurs. De plus, les performances sont dégradées par le codage lorsqu'il introduit un certain indéterminisme.

Pour améliorer les performances du test dynamique, nous pouvons envisager d'inclure des informations supplémentaires de nature probabiliste et/ou déterministe dans la modélisation.

CHAPITRE VI

PRISE EN COMPTE DE LA COMMANDE
NS LE MODELE DE LA PARTIE OPERATIVE

Dans le chapitre précédent, nous avons défini un test dynamique de la P.O. par observation des C.R. qui est une amélioration du test statique. Nous avons pu constater que le codage peut conduire à un automate non déterministe. Ceci amène une dégradation des performances du test dans la mesure où certains sommets sont ambigus. Toute évolution conduisant à un tel sommet oblige à choisir entre, déclarer une erreur qui dans la majorité des cas sera une fausse alarme, ou ignorer l'information. Dans ce second cas, nous avons vu qu'il y avait augmentation du temps de latence, voire même risque de masquage. Dans ce chapitre, nous envisageons de rendre le modèle déterministe par l'introduction d'une information relative à la commande.

.1 EVOLUTION DE LA P.O. DANS L'HYPOTHESE D'UN CYCLE DE TRAVAIL REPETITIF

.1 GENERATION DE LA SEQUENCE DE C.R.

L'objectif consiste à éliminer le non déterminisme introduit par le codage en exploitant le fait que le cycle de travail est répétitif. Il est évident que la séquence de C.R. présente elle-même une certaine périodicité. Toutefois, compte-tenu des perturbations qui agissent sur la P.O. d'une part, et des fluctuations de la sequence imposée par la P.C. dues à la boucle P.O.-P.C. d'autre part, la séquence de C.R. n'est jamais strictement répétitive.

Même sa longueur est fluctuante. Il est donc nécessaire de tolérer des différences entre les séquences.
Le passage d'une séquence observée r(1,n) à une séquence de référence r*(1,n*) nécessite trois transformations qui sont:

la substitution ri → rj ; ri, rj ∈ R

l'insertion NIL → ri

la destruction ri → NIL

Si nous associons à chaque transformation un coût, il est possible de calculer le coût total du passage de r*(1,n*) à r(1,n). Ce coût est appelé distance d'édition.
Elle peut être calculée par l'algorithme de Wagner et Fisher |WAG-74|.
Le choix d'un seuil à la distance d'édition permet de décider s'il y a une erreur. Un tel algorithme est orienté vers le test hors ligne. Un algorithme de comparaison dynamique, tel que celui présenté dans |MIC-84| pour la reconnaissance de la parole, est mieux orienté vers le test en ligne. Toutefois, cette méthode n'est pas parfaitement adaptée à notre problème, car elle suppose un coût constant pour une transformation donnée.

Exemple:
Soit une séquence $r(1,n+m+p+q+s) = (100)^n\ (000)^m\ (010)^p\ (000)^q\ (001)^s$
Le coût de la substitution C(000 → 001) est sûrement différent en début et en fin de séquence.
Nous préférons utiliser comme modèle un automate fini B(R,Q,δ, qo, qf) tel que R est l'alphabet de compte-rendu,

Q un ensemble de sommets,

qo le sommet initial,

qf le sommet terminal,

δ un ensemble d'arcs.

δ est construit de la manière suivante:
le sommet qo est associé à l'état supposé connu de la machine en début du cycle de travail.
Soit ro l'élément de R présent dans ces conditions.
A partir d'un état qi, l'apparition possible d'un élément rij ∈ R dans le C.R., entraîne la création d'un arc (qi rij qj), d'un nouveau sommet qj introduit dans Q. Il est clair qu'en cas d'actions simultanées, plusieurs arcs peuvent être issus d'un même sommet.

La construction est arrêtée lorsque tous les chemins possibles sont des cycles passant par qo.
L'ensemble des sommets ainsi créé auquel est ajouté le sommet qf forme l'ensemble Q des sommets du graphe. A partir de chaque sommet qi ainsi obtenu, il est créé k arcs (qi rif qf) d'extrémité qf, tel que pour tout arc d'origine qi,

$$\bigcup_{j=1}^{N} \{rij\} = R.$$

$\bigcup_{j=1}^{N} rij$ représente l'ensemble des étiquettes associées aux arcs d'origine qi et N = Card (R).

Remarque:
Considérons l'automate B = (R,Q, δ,qo,qf) et une partition P de Q.
Nous appelons automate dérivé de B, pour la partition P, l'automate
Bp =(R, P, δp, qo, qf)
tel que $\forall$ qk $\in$ pi; $\forall$ qe $\in$ pj ;pi,pj $\in$ P.
Si $\exists$ (qk, r(1,n) , qe) $\Rightarrow$ $\exists$ chemin (pi, r(1,n) , pj) dans Bp
Cet automate est obtenu en confondant des états de l'automate B.
Notons G = (Q, $\mathcal{S}$) un graphe pour lequel Q est l'ensemble des sommets et $\mathcal{S}$ l'ensemble des arcs.
Un sous graphe GA = (A, $\mathcal{S}$A) de G est obtenu en prenant un ensemble de sommets A $\subset$ Q et un ensemble d'arcs $\mathcal{S}$A tel que $\mathcal{S}$A regroupe tous les arcs de $\mathcal{S}$ qui joignent deux sommets de A.

6.1.2 EVALUATION DES PERFORMANCES

Comme dans le cadre du chapitre précédent, le test consiste à vérifier si la séquence est acceptable, c'est-à-dire à contrôler qu'elle n'appartient pas au langage de B.
Par construction, B est déterministe; ceci a donc un effet favorable sur le test. Pour comparer cette méthode à celle du chapitre précédent, nous utilisons la remarque suivante.
La séquence de C.R. observée pendant une période de travail a pu être générée par un chemin de l'automate $\mathcal{A}^I$ vu précédemment. Il existe donc une partition P des sommets de l'automate B telle que l'automate dérivé B(p) soit défini sur un sous graphe partiel de $\mathcal{A}^1$. Ce sous graphe est partiel dans la mesure où certains arcs y sont éliminés puisqu'ils ne sont jamais empruntés au cours du cycle de travail.

En fait ce sous graphe de $\mathcal{A}^1$ peut être obtenu par inférence à partir des séquences qui servent à construire B. A tout chemin acceptable de B correspond un chemin acceptable dans $\mathcal{A}^1$ mais la réciproque n'est pas vraie.

Exemple: Soient B=(R,q,δ,qo,qf) le modéle d'une P.O. soumise à un cycle de travail répétitif;

$\mathcal{A}'$=(R,P,δ',po,T) le modéle, non déterministe à priori, de la même P.O. pris en dehors de toute considération de commande.

soit qi $\in$ Q un sommet représenté dans pi$\in$ P. qi et pi représentent un même point singulier PSi dans les deux modéles, même si la notion d'état y est différente.

S'il y a n points singuliers accessibles à partir de PSi, il y a au moins n arcs d'origine pi dans $\mathcal{A}^1$. Dans le cadre du cycle de travail répétitif, seuls m$\leqslant$n points singuliers sont réellement atteignables. les m-n arcs présents dans $\mathcal{A}^1$ et absents de B sont à l'origine de la différence de comportement des deux modéles. Tout chemin empruntant l'un des m-n arcs de $\mathcal{A}^1$ précités n'est pas accepté par B.

Cette remarque justifie à elle seule l'interêt de la prise en compte de la commande. Voyons plus particuliérement ce qui ce passe lorsque le codage introduit un non déterminisme au voisinage de pi.

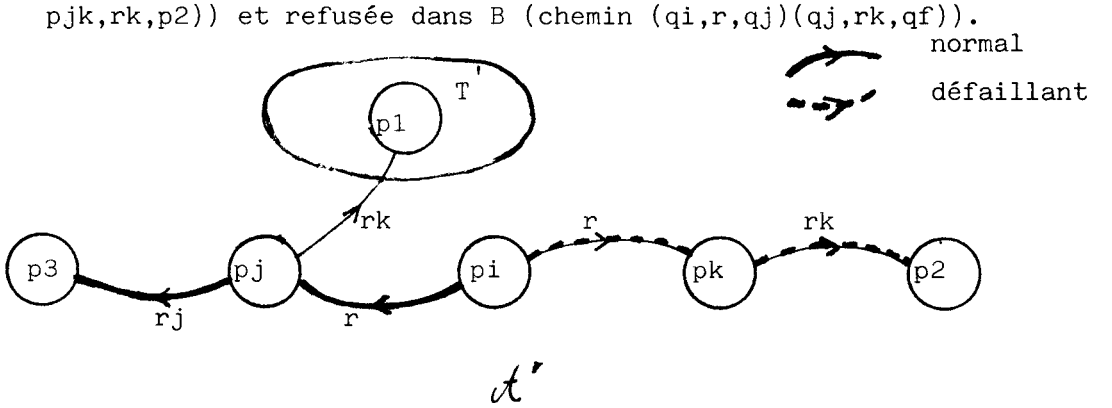
Soient B =(R,q,δ,qo,qf) et $\mathcal{A}'$=(R,P,δ',qo,T) les modéles d'une même P.O..Notons $\mathcal{A}^1$=(R,P^1,δ^1,qo,T^1) l'automate déterministe tiré de $\mathcal{A}'$.

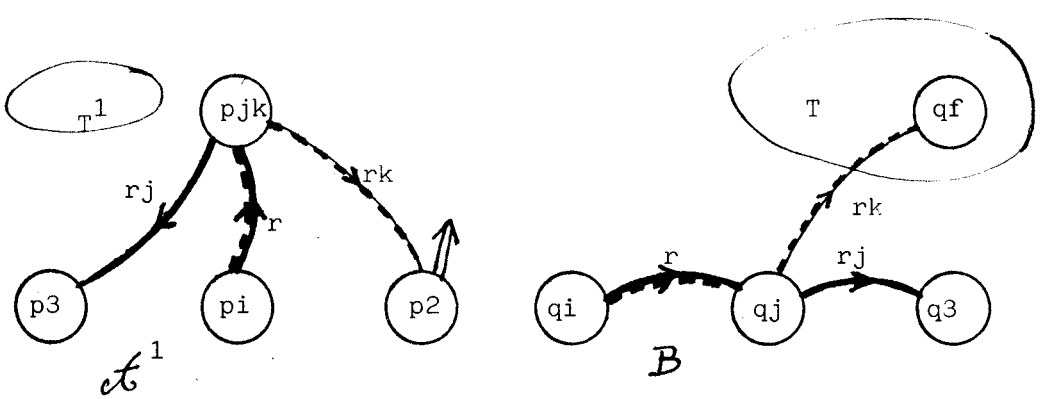
Notons qi $\in$ Q et pi $\in$ P les sommets définis comme ci-dessus. Supposons qu'il existe deux arcs de même étiquette r dans $\mathcal{A}'$ tels que: (pi,r,pj),(pi,r,pk)$\in\delta'$.

Par définition de $\mathcal{A}'$, pj et pk représentent des PS situés de part et d'autre de pi sur la trajectoire (fig. 6.1). Il existe alors dans B un arc (qi,r,qj).Supposons qu'il y est une correspondance entre $\mathcal{A}'$, $\mathcal{A}^1$ et B définie par la figure 6.1. Le tableau ci-après illustre les relations entre arcs des trois automates.

$\mathcal{A}'$	$\mathcal{A}^1$	B
(pi,r,pj)	(pi,r,pjk)	(qi,r,qj)
(pi,r,pk)		pas de corespondance
(pj,rj,p3)	(pjk,rj,p3) p3 ambigu	(qj;rj;q3)
(pk,rk,p2)	(pjk,rk,p2) p2 ambigu	(qj,rk,qf) non accepté

Supposons que la commande impose une trajectoire représentée par la séquence d'arcs (pi,r,pj)(pj,rj,p3) dans $\mathcal{A}'$ et qu'une défaillance entraine une évolution anormale selon (pi,r,pk)(pk,rk,p2). La séquence générée r rj est acceptée dans $\mathcal{A}^1$ (chemin (pi,r,pjk) pjk,rk,p2)) et refusée dans B (chemin (qi,r,qj)(qj,rk,qf)).





- figure 6.1 -

La prise en compte du cycle de travail répétitif réduit, dans un tel cas le temps de latence. Ceci permet également d'éliminer le risque de masquage lié à l'existence des sommets ambigus dans $\mathcal{A}^1$. Toutefois cette hypothèse réduit notablement le champs d'application du dispositif de test.

Voyons comment évolue le problème si nous introduisons la commande sans faire l'hypothèse du cycle répétitif.

6.2 INTRODUCTION DE RELATIONS CAUSALES DANS LE TEST DE LA P.O.

Le principe posé est qu'il existe des relations de cause à effet entre un ensemble de grandeurs formant la commande et l'évolution des grandeurs mesurées.

Hypothèse:
Nous supposons que la connaissance de la commande détermine la vitesse d'évolution des grandeurs mesurées.

L'expression "détermine la vitesse d'évolution" implique que:
à partir d'un état représenté par l'ensemble des éléments de trajectoires occupés par les grandeurs mesurées, la commande détermine les points singuliers atteignables (détermination stricte du sens).
Ce point doit être satisfait de façon stricte. Il est important ici de ne pas confondre le point singulier atteignable (élément de trajectoire) et la valeur du C.R. obtenu en ce point.

6.2.1 INTRODUCTION DE LA COMMANDE DANS LE MODELE DE LA P.O.

Commande relative à une grandeur mesurée

Soit c1, c2, ... cn un ensemble supposé fini de commandes agissant sur l'évolution de la jème grandeur mesurée.
Notons Cj l'alphabet ainsi constitué.

Modélisation de la trajectoire de la grandeur j

Soient Sj l'alphabet associé aux points singuliers de la trajectoire γ_j
Cj l'alphabet de commande.
Notons $S = \overset{k}{\underset{j=1}{\times}} Sj$ et $C = \overset{k}{\underset{j=1}{\times}} Cj$ les alphabets relatifs à l'ensemble de la P.O..
Nous reprenons la notion d'état de la jème grandeur vue au chapitre V, à savoir:

- à chaque PSi de trajectoire est associé un état qi
- il est créé un état initial qjo
  un état final qjf

tels que la grandeur j est dans l'état qjo à l'initialisation; elle atteint qjf à la suite d'une défaillance affectant la grandeur j.
L'ensemble des états de la P.O. reste identique à celui défini précédemment.
La P.O. est alors modélisée par un automate fini $\mathcal{D}=(S * C, Q, \gamma, qo, T)$.

$S*C$ est un alphabet correspondant au produit cartésien des alphabets de trajectoire et de commande,
Q l'ensemble des états de la P.O.,
γ un ensemble d'arcs étiquetés établissant une relation $S*Q \rightarrow Q$
qo l'état initial
T l'ensemble des états de la P.O. contenant un état final qjf.

La comparaison entre l'automate $\mathcal{A}=(S,Q,\delta,qo,T)$ défini au chapitre V et $\mathcal{D}=(S*C,Q,\gamma,qo,T)$ montre que ces deux modèles différent par leur alphabet et par l'ensemble des arcs.
En fait, si n est le nombre de commandes, chaque arc $\alpha ij = (qi,sij,qj) \in \delta$ donne naissance à n arcs $\alpha ij,p = (qi,sij*cp,qj) \in \gamma$.

Comme $\mathcal{A}=(S,Q,\delta,qo,T)$ est déterministe, l'automate $\mathcal{D}=(S*C,Q,\gamma,qo,T)$ l'est aussi.

6.2.2 INFLUENCE DU CODAGE

Il a été introduit l'application

$$\text{codage: } S \rightarrow R.$$

Nous créons l'automate $\mathcal{D}'=(R*C,Q,\gamma',qo,T)$ en remplaçant dans $\mathcal{D}$ les étiquettes $(s*c)ij \in S*C$ par les étiquettes $(r*c)ij \in R*C$ tel que r est l'image de s par le codage.
Le codage de $dj \in S$ représentant les défaillances affectant la jème grandeur mesurée se pose comme précédemment (§ V 2.2).

Hypothèse de défaillance

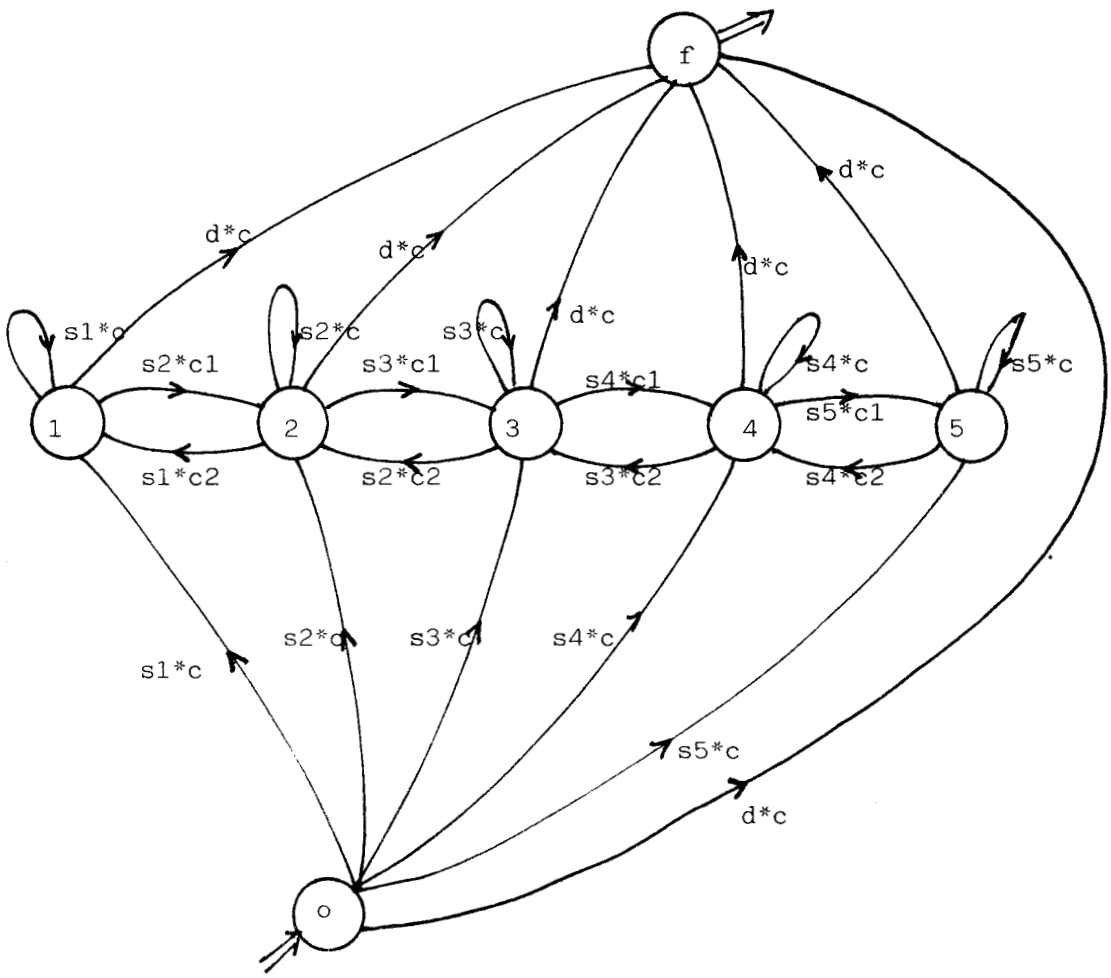
Comme au chapitre V, nous adoptons l'hypothèse suivante:
Toute modification de C.R. compatible avec l'état antérieur et la commande présente est considérée comme une évolution normale de la P.O..

Cette hypothèse conduit à la proposition suivante:

codage de dj: Tout arc conduisant vers un sommet appartenant à T (donc utilisant un symbole tel que dj) est codé de façon à ce que la réunion des étiquettes des arcs d'origine qi soit identique à R*C.

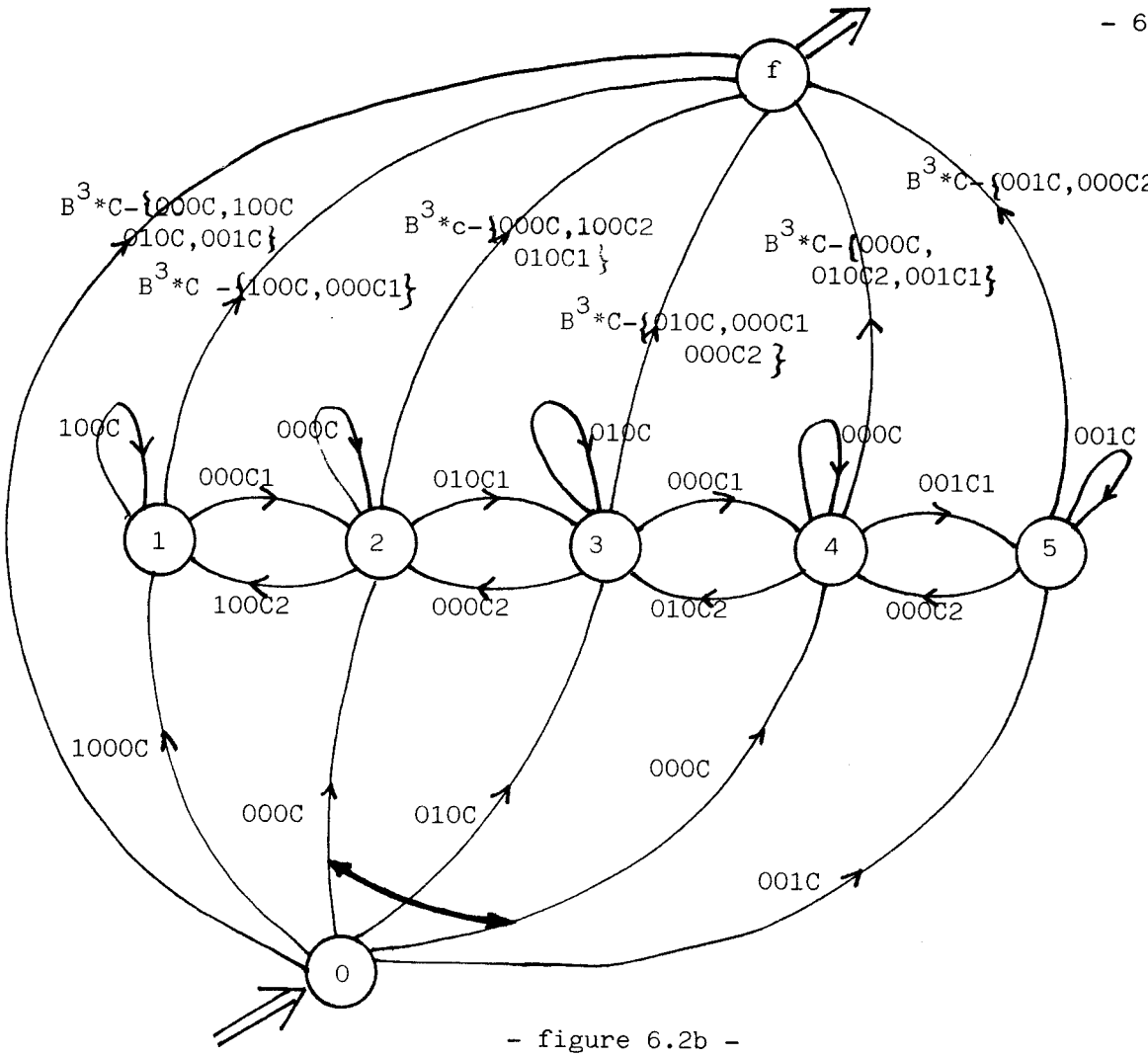
Exemple:

Nous reprenons l'exemple défini figure 5.3. Nous supposons que l'alphabet de commande est C = {c0,c1,c2} avec c0 = vitesse nulle, c1 = avance, c2 = recul. La figure 6.2 donne l'automate $\mathcal{D}$=(S*C,Q, γ ,qo,T) d'une part et $\mathcal{D}'$=(R*C,Q, γ',qo,T) d'autre part.



- figure 6.2a -

Pour alléger la représentation, nous adoptons la notation condensée suivante: si*c = si*c0, si*c1, si*c2



- figure 6.2b -

Notons Qo $\subset$ Q l'ensemble des états qui contiennent au moins une grandeur mesurée non localisée à la suite d'une initialisation.

Proposition

Le sous graphe $G^2 = (R*C, Q^2, \gamma^2)$ obtenu à partir de $G'(R*C, Q, \gamma')$ en réduisant Q^2 au complément dans Q de Qo ($Q^2 = \int_Q$ Qo) est déterministe.

Soit qi,qj,qk $\in Q^2$ et αij = (qi aij qj), αik = (qi aik qk) $\in \delta^2$ avec aij = rij * cij, aik = rik * cik.

- Si, cij $\neq$ cik alors aij $\neq$ aik
- Si cij = cik, qj $\neq$ qk alors rij $\neq$ rik donc aij $\neq$ aik.

En effet, qi $\neq$ qk entraîne le passage dans deux points singuliers PSi et PSk différents. Comme cij = cik, ces deux points singuliers sont du même côté de qi sur la trajectoire.
Par définition des points singuliers, rij et rik sont alors différents.
De par le principe de codage des éléments dj (événements de défaillance), cette proposition reste vraie si qj,qk $\in$ T.

6.2.3 LOCALISATION DE LA P.O.

Comme nous l'avons rappelé au chapitre précédent, il est possible de construire à partir de $\mathcal{D}' = (R * C, Q, \gamma', qo, T)$ un automate déterministe $\mathcal{D}^1 = (R * C, Q^1, \gamma^1, qo, T^1)$ où:

Q^1 est un ensemble de parties de Q

et T^1 l'ensemble des parties de Q appartenant à Q^1 qui contiennent au moins un élément de T.

Définition:

Nous dirons que le modèle de la P.O. est localisable s'il existe un sous graphe de $G^1 = (R * C, Q^1, \gamma^1)$ identique à G^2.

Définition:

Tout chemin (qo, r(1..n), qj) d'origine qo, de longueur n, d'extrémité qj $\in Q^2$ tel que

qi $\notin Q^2$ | (qo, r(1 ,n), qj) = (qo, r(1 ,n-1), qi)(qi,r(n),qj)

(avec r(n)= nème élément de r(1,n)),

est appelé phase de localisation.

Ce point est à rapprocher de la phase de synchronisation développée par |COU-72| pour le test hors ligne des machines séquentielles.

Proposition:

Lorsqu'une P.O. admet un modèle localisable, toute évolution qui suit une phase de localisation se fait dans le graphe déterministe G^2. Une nouvelle phase de localisation est nécessaire à l'issue de toute initialisation.

$\forall$ (qi, r(1, n) qj), qi $\in Q^2 \longrightarrow$ qj $\in Q^2$

qi $\in Q^2$ signifie que tous les points singuliers occupés par l'ensemble des grandeurs mesurées sont connus. Puisque toutes les évolutions de C.R. normales et anormales sont modélisées par G^2, aucun chemin ne peut conduire en dehors de ce sous graphe de G^1.

Intérêt d'un modèle localisable:

Nous avons vu au chapitre V que la transformation de l'automate non déterministe en automate déterministe introduit des sommets ambigus. Ces sommets peuvent être atteints aussi bien à la suite d'une évolution normale que d'un événement anormal (défaillance).

Pour rendre le test décidable, nous avons admis que les taux de panne des automatismes permettent de considérer la probabilité de l'événement normal supérieure à celle de la défaillance. Nous avons donc réduit l'ensemble des éléments terminaux aux seuls éléments non ambigus de T^1. Une telle démarche rendue indispensable pour que le test soit décidable présente deux défauts qui sont:

- une augmentation du temps de latence,
- un risque de masquage s'il existe un cycle de sommets ambigus dans $\mathcal{D}^1$.

L'intérêt d'un modèle localisable est alors évident.

Il permet d'éliminer ces défauts en dehors de la phase de localisation. Cette remarque découle du fait que G^2 est déterministe et qu'il ne contient pas de sommets ambigus.

6.2.4 MODELE LOCALISABLE

Condition nécessaire et suffisante pour que le modèle de la P.O. soit localisable

Le modèle de la P.O. est localisable si et seulement si il existe une séquence qui puisse être générée de façon exclusive par des chemins de $\mathcal{D}'$ d'origine qo et de même extrémité $q \in Q^2$.

Cette condition est suffisante:

Soit (qo, a(1,k),q) un tel chemin. Pour toute séquence générée de longueur $n \geq k$, telle que a(1, n) = a(1, k) a(k+1, n), le passage par $q \in Q^2$ est obligatoire par hypothèse.

Tous les arcs d'origine q ont des étiquettes différentes; donc, par construction de $\mathcal{D}^1$, les extrémités de ses arcs, éléments de Q', sont inclus dans Q^1. Cette construction reprise de proche en proche conduit au sous graphe G^2.

Cette condition est nécessaire:

$\forall$ a(1, n) $\in R_* C^+$;s'il existe des chemins (qo,a(1,n),q1),(qo,a(1,n),q2) de $\mathcal{D}'$ telsque q1 $\neq$ q2 $\longrightarrow$ par construction de $\mathcal{D}^1$ les sommets q1, q2 sont regroupés dans une même partie de Q.

Cet élément de Q^1 ne fait pas partie de Q^2. Ce raisonnement étant supposé vrai pour toute séquence générée, G^2 n'est pas un sous graphe du graphe associé à $\mathcal{D}^1$.

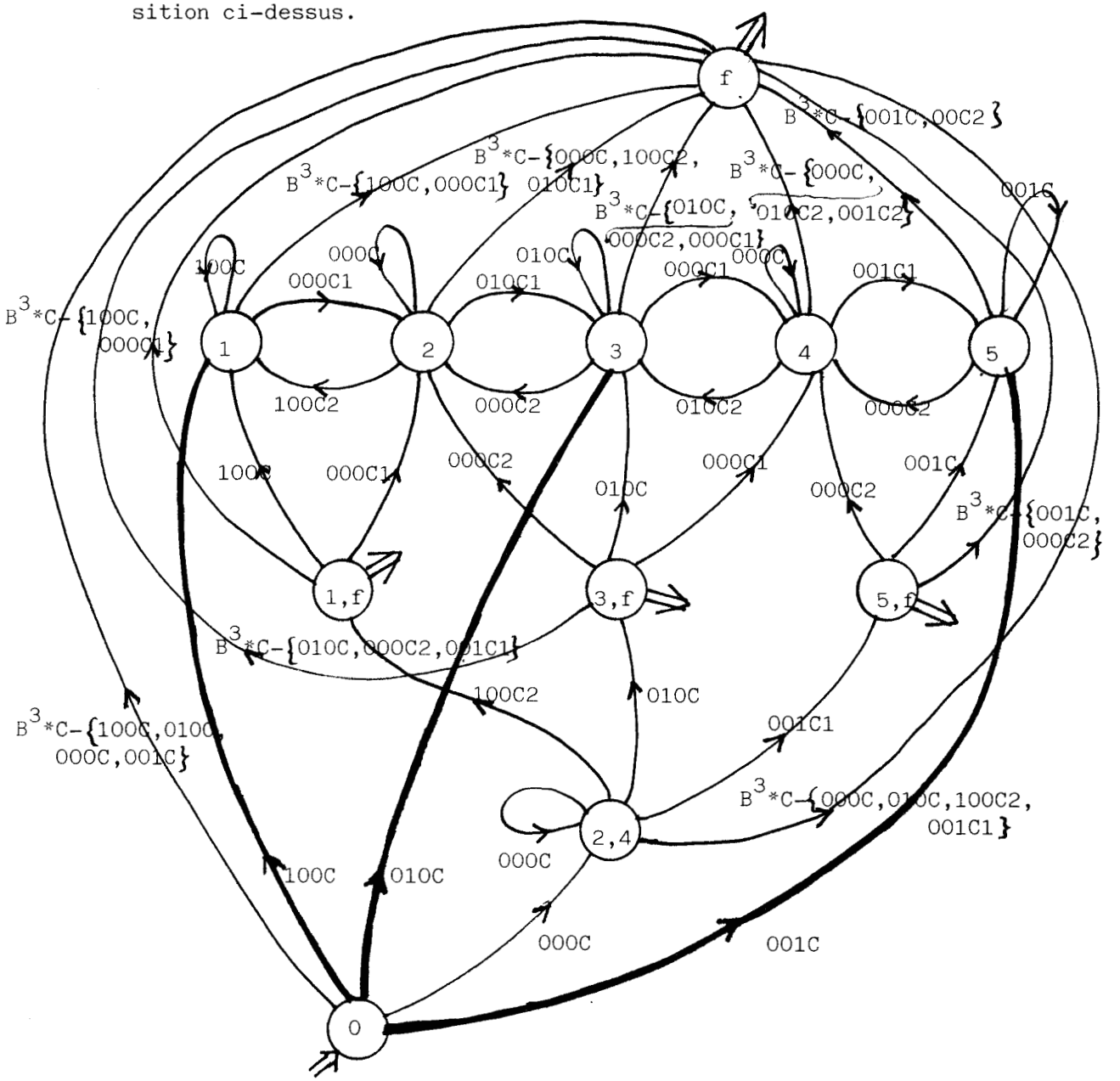
Dans la pratique, de nombreux modèles de P.O. sont localisables. Nous envisageons ici deux conditions suffisantes parmi les plus répandues.

Proposition:

S'il existe au moins une étiquette $r*c \in R*C$ telle que tout arc de $\mathcal{G}'$, d'étiquette r_*c a même extrémité $q \in Q$ et si de plus $q \notin T$, alors le modèle est localisable.

Soit (qo,a(1,n),qn) un chemin pour lequel il existe k $1 \leq k \leq n$ avec $a(k)=r_*c$. Par hypothèse, (qo,a(1,k),q) méne au sommet $q \in Q'$. a(1,k) est une phase de localisation.

Cette proposition couvre le cas où l'automate $\mathcal{A}'$ est lui même déterministe. la figure 6.3 donne l'automate déterministe $\mathcal{A}^1$ construit à partir de l'automate non déterministe de la figure 6.2. Les arcs (qo,100*c,q1),(qo,010*c,q3),(qo,001*c,q5) répondent à la proposition ci-dessus.



- figure 6.3 -

Trajectoire ouverte

S'il existe au moins une commande c pour laquelle il n'existe pas de cycle de longueur strictement supérieur à 1 dans G^2 (sous graphe de G'); alors, le modèle de la P.O. admet une trajectoire ouverte pour la commande.

Dans ces conditions, il existe un sommet $qt \in Q$, $qt \notin T$ appelé terminaison tel que:

$$\exists (qi, ri*c, qt) \in \gamma'$$

$$\forall r \in R, \text{ si } \exists (qt, r*c, qj) \in \gamma' \rightarrow qj \in T.$$

Proposition

Toute P.O. qui présente une trajectoire ouverte pour une commande c est localisable.

Soit (qo, r(1, n1),qt)(qo, r(1, n2), qt) ... (qo, r(1, np), qt) l'ensemble des chemins sans itérations d'arcs joignant qo à qt générant une séquence d'étiquettes dépendant de c.

Comme l'automate est fini et que, par définition de la trajectoire ouverte, il n'y a pas de cycle pour la commande c, la longueur de telles séquences est bornée.

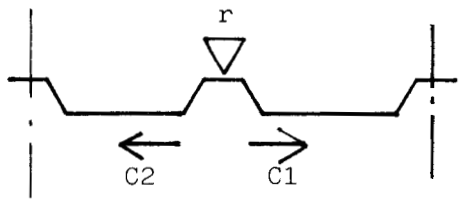
Notons Max(ni) cette borne. Tout chemin sans itération d'origine qo de longueur Max(ni) pour lequel c est maintenu, est une phase d'initialisation. Un tel chemin est caractérisé par un nombre de changements d'étiquette égal à Max(ni). Cette propriété est conservée si nous admettons les itérations d'étiquettes liées au maintien dans les points singuliers considérés comme états stables de l'automate.

Nous avons recherché ici, l'existence d'une phase de localisation sans nous préoccuper de la reconnaissance d'une telle phase en exploitation. Celle-ci dépend en effet de la séquence de travail imposée par la commande.

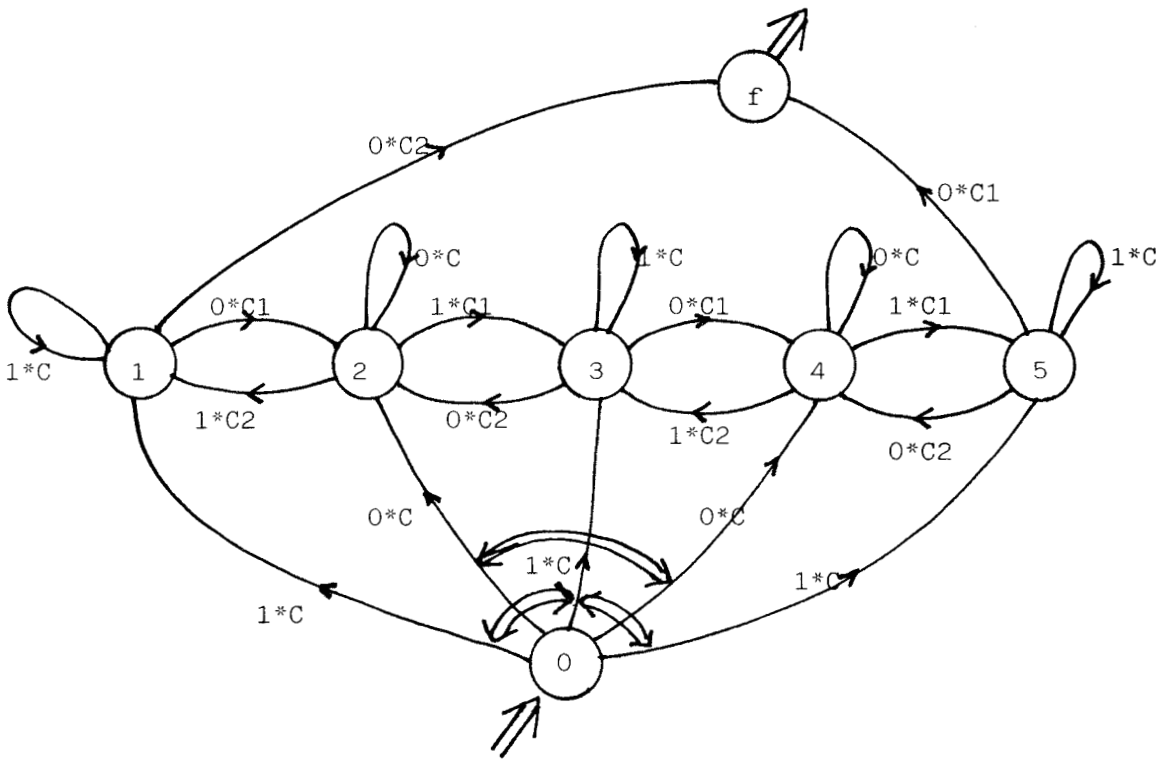
Si nous faisons l'hypothèse que la P.O. reste saine pendant toute la phase de localisation, la suppression des sommets ambigus ne remet pas en cause la valeur du modèle déterministe obtenu. Tout chemin dans ce graphe qui conduit de qo à qt est donc non ambigu et constitue une phase de localisation. Toute P.O. localisable peut donc être localisée. La longueur de cette phase est alors tributaire du nombre de cycles de $\mathcal{D}^1$ contenu dans la séquence correspondante.

Nous traitons à titre d'exemple le cas d'un capteur incrémental sur une trajectoire ouverte (Figure 6.4), modélisé par les automates $\mathcal{D}'$ (Figuer 6.5a) et $\mathcal{D}^1$ (Figure 6.5b).
Les figures 6.7 et 6.8 illustrent le cas d'une trajectoire non ouverte.

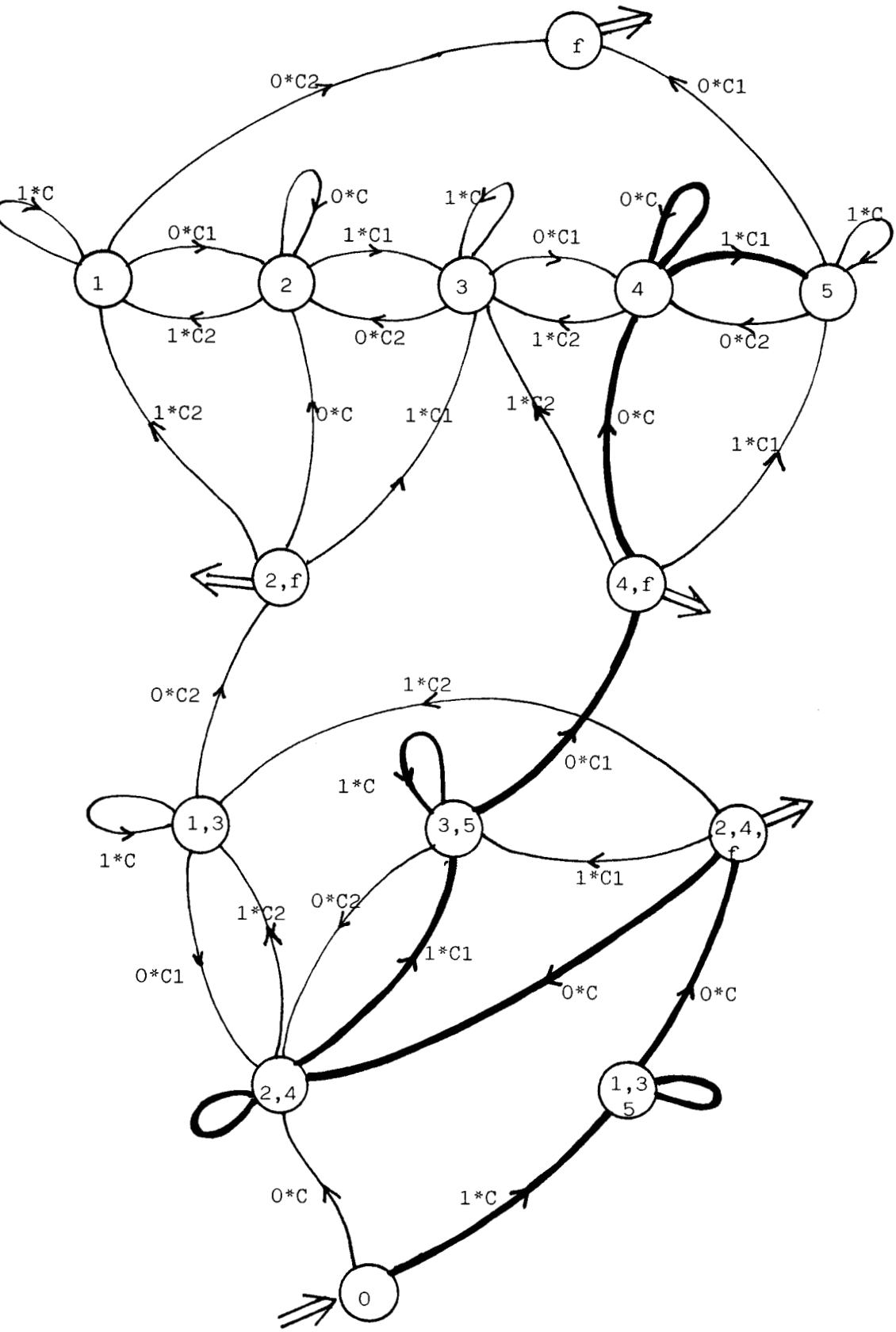
La séquence 1 * c1, 0 * c1, 1 * c1, 0 * c1, 1 * c1 est une phase de localisation. Ceci est lié aux trois occurences de 1 * c1 dans la séquence. Cette séquence est en trait fort dans la figure 6.5b.



- figure 6.4 -

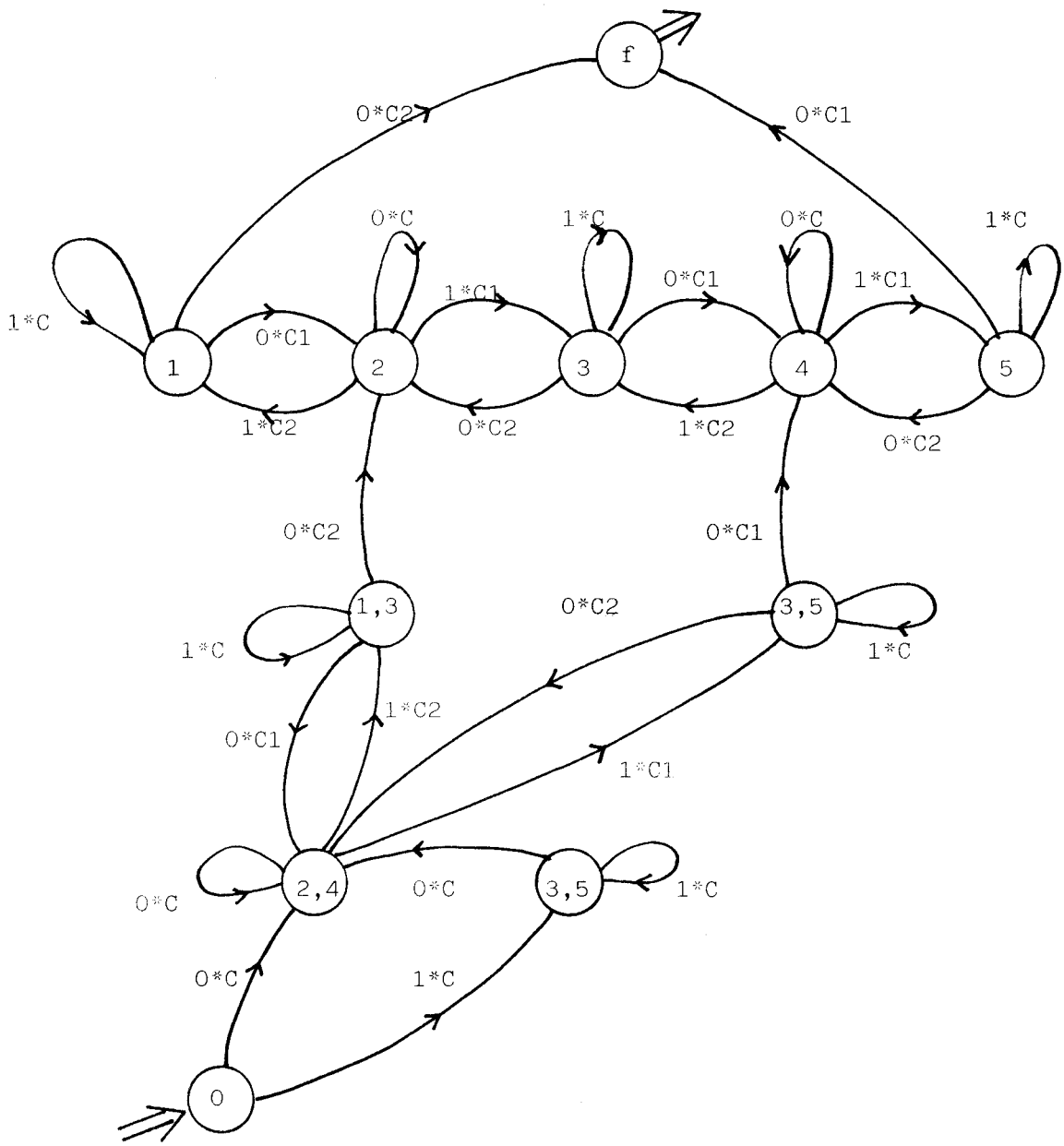


- figure 6.5a -


f
0*C2
0*C1
1*C
0*C
1*C
0*C
1*C
1
2
3
4
5
0*C1
1*C1
0*C1
1*C1
1*C2
0*C2
1*C2
0*C2
1*C2
0*C
1*C1
1*C2
0*C
1*C1
2,f
4,f
0*C2
1*C2
0*C1
1*C
1,3
3,5
2,4,
f
1*C
1*C1
0*C2
1*C2
1*C1
0*C1
0*C
0*C
2,4
1,3
5
0*C
1*C
0

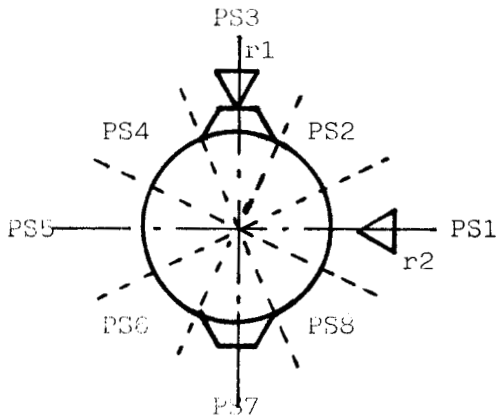
- figure 6.5b -

Par convention, nous limitons les éléments finaux aux seuls sommets non ambigus de $\mathcal{D}^1$. Cette régle permet de rendre le test décidable. Il est alors possible de fusionner certains états ambigus. La figure 6.6 donne le graphe obtenu après fusionnement des états fusionnables de l'automate de la figure 6.5b.



- figure 6.6 -

Dans le cas d'un capteur incrémental placé sur une trajectoire fermée (Figure 6.7), la condition de localisation n'est pas satisfaite. Dans l'automate déterministe (Figure 6.9), il est impossible de définir un sous graphe G^2 commun à l'automate non déterministe (Figure 6.8) en dehors de l'ensemble terminal (réduit ici à qf).

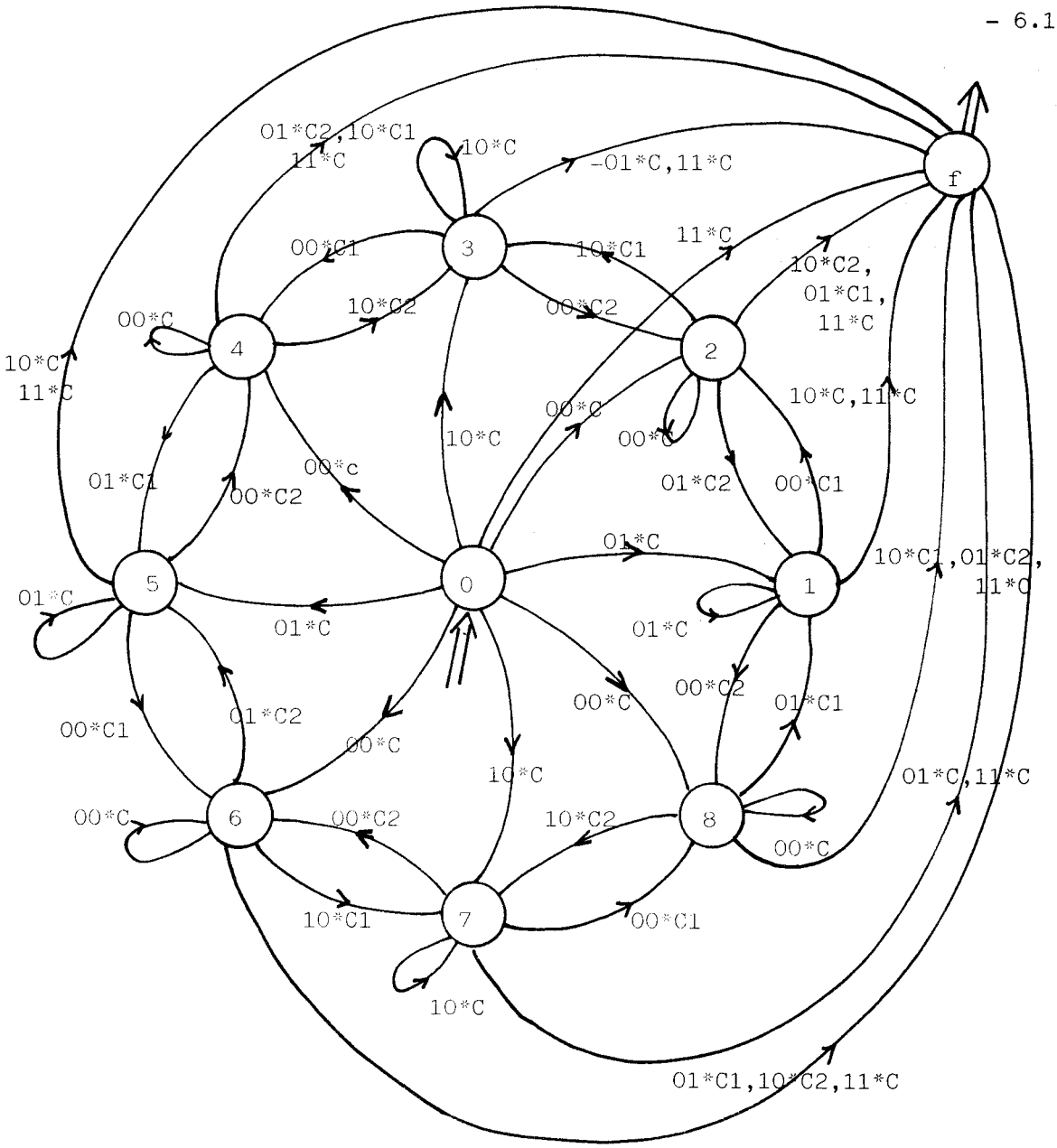


- a -

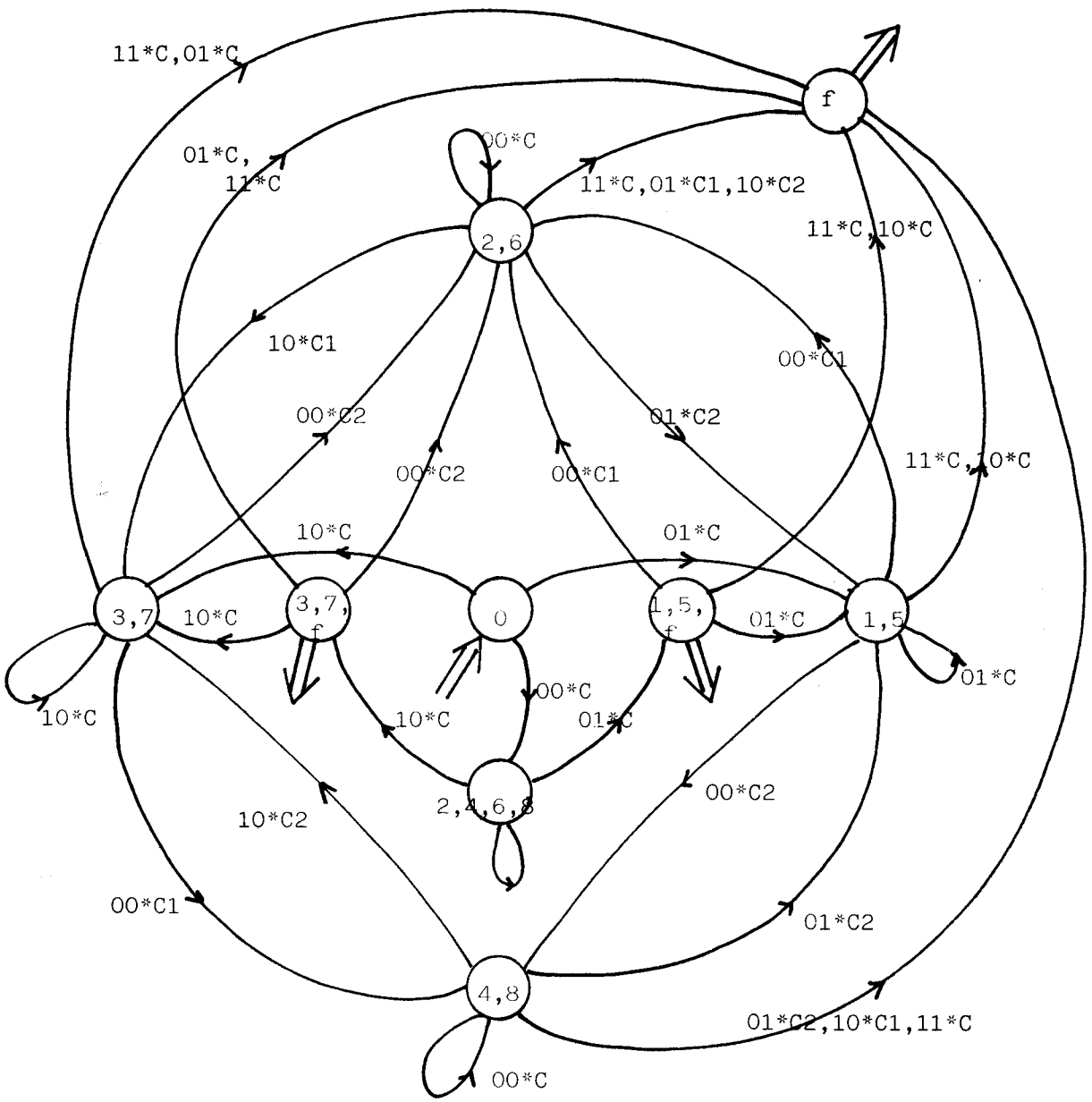
	r1	r2
PS5,PS1	0	1
PS6,PS2	0	0
PS7,PS3	1	0
PS8,PS4	0	0

- b - Codage des points singuliers.

- figure 6.7 -


01*C2,10*C1
11*C
10*C
-01*C,11*C
f
00*C1
3
10*C1
11*C
10*C2,
01*C1,
11*C
00*C
10*C2
00*C2
4
2
10*C
11*C
10*C,11*C
00*C
10*C
00*C
01*C1
00*c
00*C2
01*C2
00*C1
01*C
10*C1,01*C2,
11*C
5
0
1
01*C
01*C
01*C
00*C
00*C2
01*C1
00*C1
01*C2
00*C
10*C
01*C,11*C
00*C
6
00*C2
10*C2
8
00*C
10*C1
7
00*C1
10*C
01*C1,10*C2,11*C

- figure 6.8 -



- figure 6.9 -

Il est possible d'observer,sur cet exemple, l'existence d'une phase de speudo localisation. Cette remarque est justifiée par la présence du graphe 1,5 - 2,6 - 3,7 - 4,8 qui ne comprend aucun sommet ambigu, et pour lequel il n'existe aucun arc de sortie.

6.3 INFLUENCES DES LIMITES DES GRAMMAIRES REGULIERES SUR LE TEST

La prise en compte du contexte de la commande permet d'obtenir un modèle déterministe

- en permanence, si nous faisons l'hypothèse d'un cycle de travail répétitif;
- à l'issue de la phase de localisation, si la commande est une information connue.

Les avantages d'une telle propriété au niveau du test sont importants:

- le temps de latence d'une erreur est limité,
- le risque de masquage d'une erreur liée aux séquences ambiguës est supprimé.

Ces avantages ont été développés au long de ce chapitre et notamment dans le paragraphe 1.2.

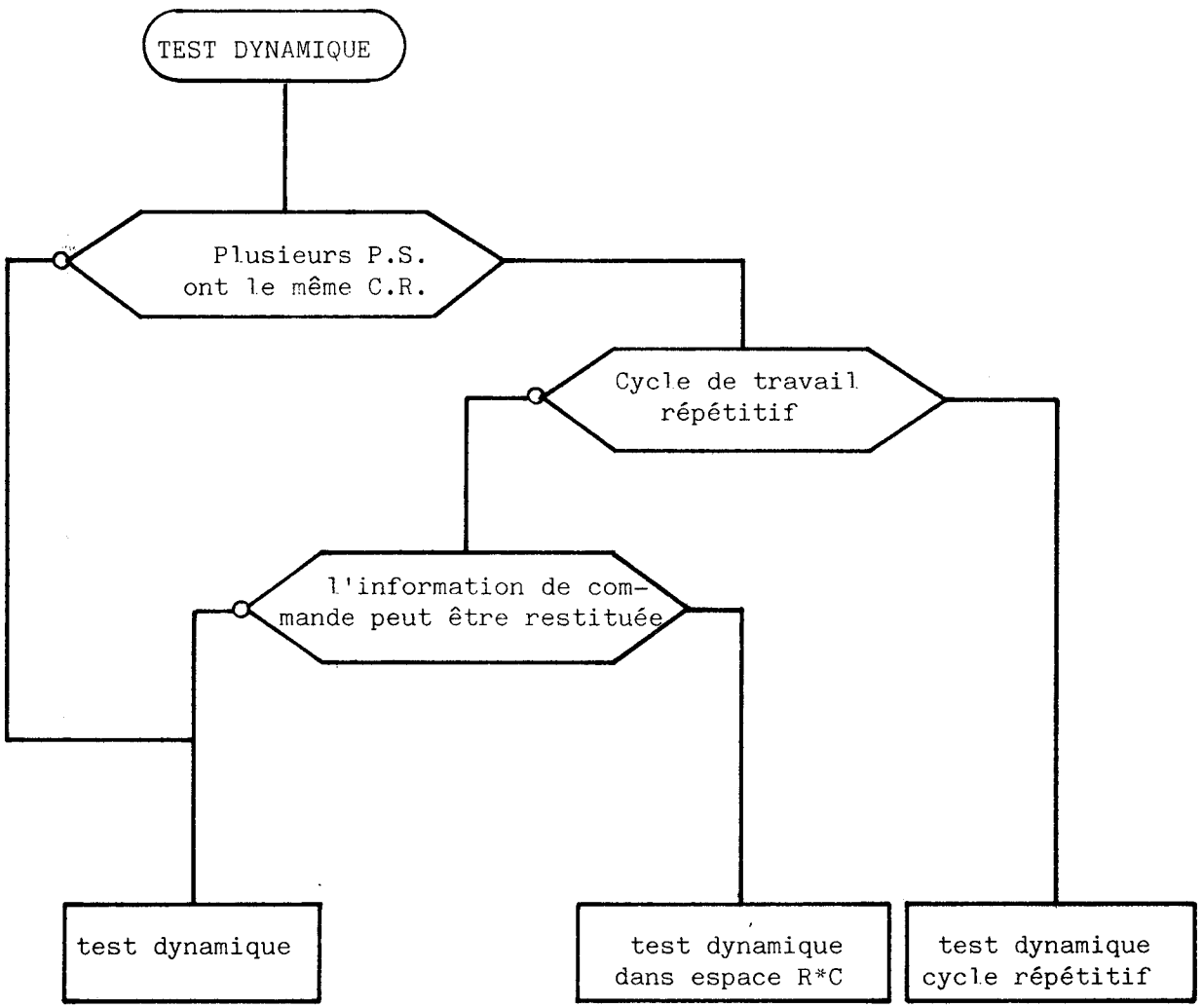
Le non déterminisme du modèle étant introduit par le codage, il est clair que la prise en compte du contexte de la commande ne se justifie que lorsque plusieurs points singuliers ont le même C.R.. S'il en est autrement, le test dynamique hors contexte apporte les mêmes performances.

L'hypothèse du cycle de travail répétitif est une restriction du domaine d'application. Il est clair que la prise en compte de la commande implique un domaine plus vaste, mais au prix de deux difficultés:

- la nécessité d'avoir une information sur la commande,
- la vulnérabilité du test dans la phase de localisation.

Ce dernier point montre l'intérêt d'une phase de localisation de courte durée. Pour ce qui est de la prise en compte de la commande, la question est de savoir si cette information est observable ou si elle peut être reconstituée à partir de grandeurs observables (état du grafcet de commande, des ordres, ...). Ce point sera développé dans la partie réalisation.

Si l'utilisation d'un mécanisme de détection de défaillances est jugée nécessaire et si le test statique offre des performances insuffisantes, le recours à un test dynamique s'impose. La figure 6.10 propose un algorithme de choix parmi les trois tests dynamiques présentés.



- figure 6.10 -

Quelque soit le modèle adopté, le codage de l'indicateur de défaillance noté dj a été obtenu de façon à ce que tout élément de R, accessible compte tenu de la géométrie des trajectoires et éventuellement de la commande appliquée, soit accepté. Une telle convention contient en elle-même une limitation de performance, car il est certain que l'arrivée d'un tel C.R. peut être considéré comme anormal en dehors de certaines limites de temps. En d'autres termes, les anomalies conduisant à un blocage ou à une variation de la vitesse d'exécution d'une action, ne sont pas décelables. L'introduction de "chiens de garde" dans le modèle devrait permettre de mettre en évidence certaines de ces anomalies. Ceci fait l'objet du chapitre suivant.

CHAPITRE VII

MODELES TEMPORISES

L'introduction de temporisations dites de "chien de garde" dans le modèle revient à considérer le temps de maintien dans un état.
En d'autres termes, il faut donner au modèle la possibilité de compter les itérations d'arcs qui traduisent la persistance de cet état. La référence à une grammaire régulière utilisée jusqu'ici nous interdit cette pratique. Nous devons donc nous tourner vers d'autres grammaires.

7.1 GRAMMAIRES NON REGULIERES

Il existe de nombreuses grammaires et notamment celles qui sont représentables par des automates stockastiques ou des automates flous |MIC-84|. L'objectif de ces grammaires est d'évaluer la confiance que l'on peut accorder à une affirmation du style: "cette phrase fait partie du langage de l'automate". Dans les deux cas, il est associé à chaque arc une métrique qui permet cette évaluation. De tels modèles sont exploitables pour le test hors ligne. Ils permettent de diminuer la longueur des séquences de test en fonction d'un taux d'échec admissible. De même, ils permettent de guider le choix d'une règle de production parmi un ensemble de règles applicables dans le cadre des systèmes experts.
Pour le test en ligne, de telles méthodes présentent le défaut de rejeter toute évolution normale, mais peu fréquente, sans apporter les limites temporelles attendues.
Nous nous rapprochons alors des grammaires programmables présentées dans |FU - 74|.

Notons X l'alphabet du langage et V un alphabet auxiliaire.
Dans une grammaire programmée, une règle de production s'écrit:

soit $x \in X$, A, B, $C \in V$

$A \longrightarrow xB \mid xC \mid x$

Le choix entre les différentes réécritures de A est fait par application d'une règle complémentaire, généralement basée sur l'algèbre des prédicats. Nous allons définir une telle grammaire que nous appliquerons ensuite au test en ligne.

7.2 AUTOMATE PONDERE

7.2.1 DEFINITION ET PROPRIETES D'UN AUTOMATE PONDERE

Définition

Soit un alphabet X et un graphe G(X, Q, δ) dont les étiquettes sont des éléments de X.

A chaque sommet qi $\in$ Q est associé un nombre entier positif appelé poids et noté pi.

Soit qi, qj $\in$ Q et xij $\in$ X tels que (qi,xij,qj) = α ik $\in \delta$.

A tout arc tel que α ik, nous associons un intervalle de valeurs possibles pour pi noté Iik et une application V(α ik) appelée prédicat.

V(α ik) : pi $\longrightarrow$ v(α ik) $\in \{ 0, 1 \} = B$

telle que v(α ik) = 1 si pi $\in$ Iik

v(α ik) = 0 si pi $\notin$ Iik

Un graphe pondéré Gp(X, V, Q, δ p) est un quadruplet

où X est l'alphabet d'entrée,

V un ensemble de prédicats,

Q un ensemble de sommets

δp un ensemble d'arcs.

Propriété

Le graphe pondéré Gp(X, V, Q, δ p) se comporte comme un graphe G(X, Q, δ) tel que l'arc α ik = (qi,xij,qj) $\in \delta$ p est aussi un élément de δ si et seulement si v(αik) = 1.

Configuration d'un automate pondéré

Une configuration est connue lorsque le poids affecté à chaque élément de Q est fixé.

Pour une configuration d'indice n, nous noterons G(n) le graphe G(X, Q, δ (n)) obtenu à partir de Gp(X, V, Q, δ p) par application de la règle ci-dessus.

Automate pondéré

Nous définissons un automate pondéré $\mathcal{A}$p=(X, V, Q, δ p, qo, T) par un sextuplet où X, V, Q, δ p sont les éléments du graphe pondéré Gp(X, V, Q, δ p),

qo un sommet initial; qo $\in$ Q,

T un ensemble de sommets terminaux; T $\subset$ Q.

Automate de référence

L'automate de référence $\mathcal{A}$ ref relatif à $\mathcal{A}$p = (X, V, Q, δ p, qo, T) est le sextuplet $\mathcal{A}$ref = (X, Q, δ , qo, T) obtenu en faisant $v(\alpha ik) = 1$; $\forall \alpha ik$.

Pour une configuration donnée, nous obtenons l'automate fini $\mathcal{A}(n)=(X, Q, \delta(n), qo, T)$.

Propriété

Un automate pondéré $\mathcal{A}$p génére une famille d'automates finis $\mathcal{A}(n) = (X, Q, \delta(n), qo, T)$ par modification de la configuration. Si l'automate $\mathcal{A}(n)$ est déterministe pour toutesles configurations possibles, nous dirons que $\mathcal{A}$p est déterministe.

Proposition

Si, pour l'ensemble des prédicats, l'intersection des intervalles Iik correspondants aux arcs de même sommet q, de même origine qi et de même étiquette rij $\in$ R est vide, l'automate pondéré est déterministe.

Soit $\alpha i1, \alpha i2, \ldots, \alpha in_i$ l'ensemble des arcs tels que $\alpha ik = (qi, rik, qk)$ avec $k = 1, 2, \ldots, n_i$; $\forall k$; $rik = ri1$

Soit Iik l'intervalle fermé associé à αik.

La proposition ci-dessus s'exprime alors:

$$\forall qi \in Q', \bigcap_{k=1}^{n_i} Iik = \emptyset \longrightarrow \forall n; \mathcal{A}(n) \text{ est déterministe.}$$

$$\bigcap_{k=1}^{n_i} Iik = \emptyset \Rightarrow \forall j, k \in \{1 .. n_i\}, \nexists pi \text{ tel que } pi \in Iij, pi \in Iik$$

donc $v(\alpha ij) = v(\alpha ik) = 1$ est impossible.
Par construction de $\mathcal{A}(n)$, nous tirons, pour une configuration n quelconque $\nexists \alpha ij, \alpha ik \in \delta(n)$ ayant même étiquette.

L'hypothèse étant supposée vraie pour tout i, l'automate obtenu est déterministe pour toute configuration. $\mathcal{A}$ p est donc déterministe.

Remarque

Si l'automate de référence est déterministe, la proposition est toujours satisfaite puisqu'il n'y a, dans ce cas, qu'un seul arc d'origine qi et d'étiquette r.

Par contre, si l'automate de référence est non déterministe et si la proposition ci-dessus n'est pas satisfaite, alors $\mathcal{A}$p est non déterministe.

Transformation d'un automate pondéré non déterministe en automate pondéré déterministe

Cette transformation a été appliquée dans le cas des automates finis à $\mathcal{A} = (R, Q, \delta, qo, T)$ pour obtenir $\mathcal{A}' = (R, Q', \delta', qo, T')$ où Q' est contenu dans l'ensemble des parties de Q.
Cette transformation ne peut être applicable aux automates pondérés qu'à la condition de pouvoir définir une règle d'affectation des poids dans $\mathcal{A}'$p qui conserve les propriétés liées au poids dans $\mathcal{A}$ P.
Il faut donc que le poids soit conservatif dans cette transformation.

Chemin dans un automate pondéré

Pour une configuration donnée, l'automate pondéré se comporte comme un automate fini G(n). La notion de chemin et de séquence générée conserve la même signification, pour les automates pondérés.

Appelons sommet absorbant tout élément $qi \in Q$ tel que $\forall r \in R$ tout arc $(qi,r,qj) \notin \delta$ si $qj \neq qi$.

Chemin fatal

Tout chemin qui conduit à un sommet absorbant est fatal si ce sommet n'est pas un élément final. Pour qu'un tel chemin ne puisse pas être rencontré, il suffit qu'aucun élément de Q n'appartenant pas à T soit sommet absorbant.
Si nous définissons des règles de calcul de la pondération liées à la séquence engendrée, l'automate dans lequel nous évoluons est évolutif.

L'existence de chemins fatals est alors une éventualité à étudier.

Absence de chemin fatal

Notons $\mathbb{P}i$ l'ensemble des poids pi que peut prendre $qi \in Q$ pour tout chemin conduisant de qo (sommet initial) à qi. Cet ensemble est spécifié par les règles de calcul de pi.

Soit $\alpha i1$, $\alpha i2$, ... , αic l'ensemble des arcs de $\delta(p)$ d'origine qi et d'extrémité $qj \neq qi$ et (Ii1, Ii2, ... , Iic) les intervalles correspondants.

Proposition

Il n'y a pas de chemin fatal si:

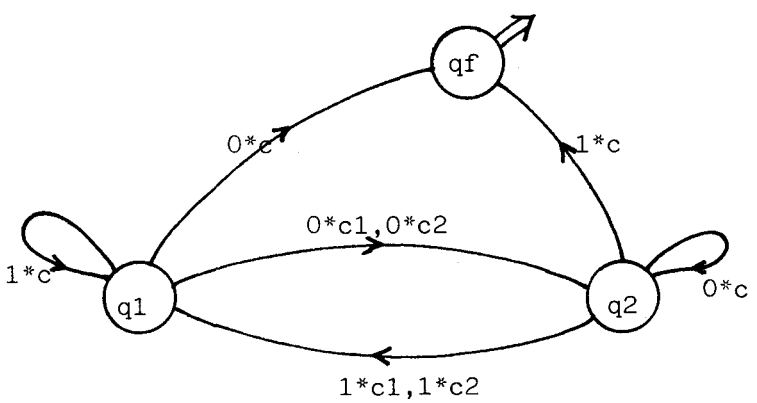
$\forall$ qi; $\bigcup_{k=1}^{c}$ Iik $\supset \mathbb{P}i$.

En effet, $\forall$ p $\in \mathbb{P}i$, $\exists$ k tel que p $\in$ Iik

Donc, V(α ik) = 1 et l'arc α ik d'origine qi et d'extrémité $qk \neq qi$ existe.

7.2.2 EXEMPLE D'UTILISATION

Reprenons l'exemple du capteur incrémental sur une trajectoire ouverte (Figure 4.4). Soit N le nombre d'incréments. Supposons que la trajectoire se termine par la fermeture de C comme dans cet exemple (ici N = 3, mais des valeurs nettement supérieures sont à envisager).
En dehors de la phase de localisation, cette P.O. peut être représentée par un automate pondéré Dp dont Dref est donné figure 7.1.



- figure 7.1 -

Soit c1, c2, c3 les commandes avant, arrière, arrêt (c = c1 + c2 + c3). Notons $\alpha 1$, $\alpha 2$ = (q1, 0*(c1, c2), q2) les arcs obtenus en donnant à la commande successivement les valeurs c1, c2.

$\alpha 3$, $\alpha 4$ = (q2, 1*(c1, c2), q1)
$\alpha 5$, $\alpha 6$, $\alpha 7$ = (q1, 1*(c1, c2, c3), q1)
$\alpha 8$, $\alpha 9$, $\alpha 10$ = (q2, 0*(c1, c2, c3), q2
$\alpha 11$, $\alpha 12$, $\alpha 13$ = (q1, 0*(c1, c2, c3), qf)
$\alpha 14$, $\alpha 15$, $\alpha 16$ = (q2, 1*(c1, c2, c3), qf)

Adoptons les règles de pondération suivantes:

Poids p2 = 1 = constante
p1 = p1 + p avec
p = +1 pour toute évolution selon $\alpha 3$ = (q2, 1*c1, q1)
p = -1 pour toute évolution selon $\alpha 4$ = (q2, 1*c2, q1)
p = 0 dans tous les autres cas.

Nous supposons connue la valeur initiale de p1 comprise dans $\{1 .. N\}$. Les intervalles choisis sont alors:

I3 = I4 = I8 = I9 = I10 = I16 = $[1]$
I14 = I15 = $[0]$

I1 = $[1 .. N-1]$; I11 = $[N, +\infty]$
I2 = $[2 .. N]$; I12 = $[0, 1]$
I13 = $\mathbb{N}$

I5, I6, I7 = $\mathbb{N}$

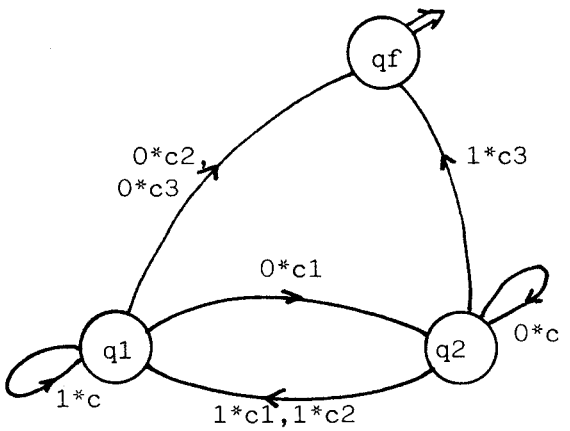
Nous donnons figure 7.2 les graphes obtenus pour les configurations correspondantes à p1 = 1, 1 < p1 < N, p1 = N

Il est clair que du point de vue réalisation, ce modèle correspond à l'association d'un automate fini et d'un compteur-décompteur dont l'incrémentation est obtenue à chaque transition q2 ⟶ q1. Cette évolution se traduit par un passage de 0 à 1 du capteur. Le problème de la localisation n'a pas été abordé. En fait, dès que le compteur décompteur initialisé à "0" atteint la valeur $\pm$ N/2, le modèle peut être calé par utilisation de la correspondance suivante

L: pi = -N/2 ⟶ pi = 0
L: pi = +N/2 ⟶ pi = 1

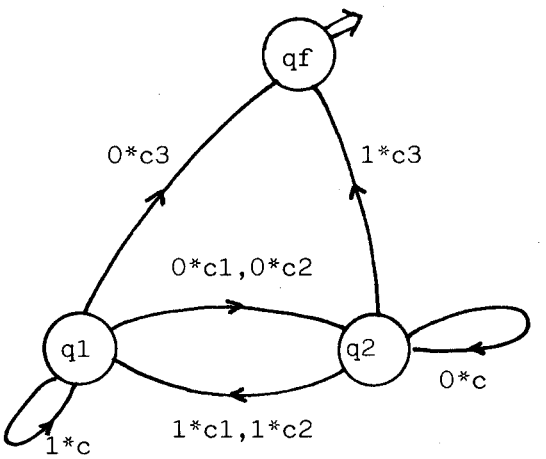
sinon pi est inchangé.

Ce problème est traité ici en guise d'exemple d'utilisation d'un automate pondéré. De nombreuses applications sont envisageables. Dans le cadre du test, la pondération introduite a une signification tout à fait différente de celle que nous avons adopté dans cet exemple.



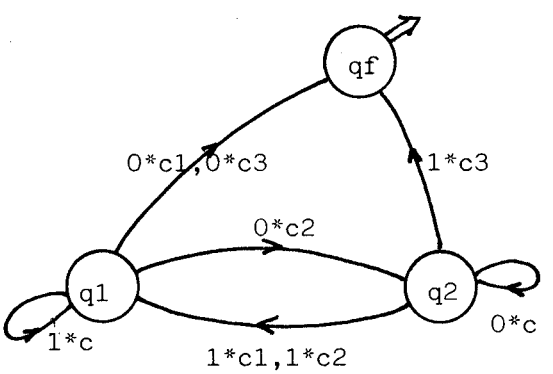
p1=1 ; p2=1

- a -



1 < p1 < N ; p2=1

- b -



p1=N ; p2=1

- c -

- figure 7.2 -

7.3 MODELE PONDERE DANS LE CADRE DU CYCLE DE TRAVAIL REPETITIF

7.3.1 ETABLISSEMENT DU MODELE; CHOIX DES REGLES DE PONDERATION

Nous utilisons un automate de référence B ref = (R, Q, δ^r, qo, qf) bâti à partir de l'automate B = (R, Q, δ, qo, qf) défini au paragraphe VI-2. L'automate B est modifié de façon à ce que pour tout sommet qi $\in$ Q, différent de qo, tout élément r $\in$ R soit étiquette d'un arc (qi, r, qf).

B ref est donc *non* déterministe. Il conserve néanmoins la propriété de B suivante:

le sous graphe G = (R, Q', δ^r) extrait de G ref = (R, Q, δ^r) tel que Q' est l'ensemble des éléments de Q autres que qf est déterministe.

Ceci entraîne:

$$qj \neq qk \longleftrightarrow rik \neq rij \mid (qi, rij, qj), (qi, rik, qk) \in \delta^r \quad ; qi, qk \neq qf$$

Règles de détermination des Iik utilisées dans la modélisation

A chaque arc α ik, αih, αiv de B'ref est alors associé un intervalle Iik, Iih, Iiv $\subset \mathbb{N}$ tel que

a) si α ik = (qi,rij,qj) et qj $\neq$ qi, qj $\neq$ qf

$\longrightarrow$ Iik = | Min pik, Max pik |

dans ce cas $\exists \alpha$'ik = (qi,rif,qf) tel que rif = rij

Pour cet arc, nous avons I'ik = $\mathbb{N}$ - Iik

b) si α ii = (qi,rii,qi), qi $\neq$ qf

$\longrightarrow$ Iii = | 1, Max (Max pik) |

dans lequel Max (Max pik) représente la plus grande valeur de la borne supérieure de l'intervalle Iik obtenu par application de la règle a.

$\exists \alpha$'ii = (qi, rif, qf) avec rif = rii auquel est associé I'ii = $\mathbb{N}$ - Iii

c) $\nexists \alpha$ iv = (qi,rij,qj) avec qj $\neq$ qf

alors $\exists \alpha$'iv(qi,rif,qf) avec rif = rij tel que I'iv = $\mathbb{N}$

L'ensemble de ces règles spécifie l'automate pondéré Bp(R, V, Q, δ p, qo, qf) recherché.

Calcul du poids pi du sommet qi ϵ Q

- Toute évolution selon un chemin de longueur 1 (donc réduit à un arc) modifie le poids pi affecté à qi de la façon suivante:

  $\forall$ qi, qj $\neq$ qf pour (qi, r, qi) $\longrightarrow$ pi est incrémenté

  pour (qi, r, qj); qj $\neq$ qi $\longrightarrow$ pi est mis à 0

  pour (qj, r, qi); qj $\neq$ qi $\longrightarrow$ pi est mis à 1.

  Dans tous les autres cas pi est inchangé.

- A la création d'un chemin d'origine qo (correspondant à une initialisation), la configuration initiale est définie par:

  $\forall$ i qi = qo $\longrightarrow$ po = 1

  qi $\neq$ qo $\longrightarrow$ pi = 0

Procédure de test

La procédure de test reste inchangée, une erreur est révélée par l'existence d'une séquence non acceptable générée par un chemin passant par qf.
La différence avec la méthode précédente provient de ce que l'automate utilisé est modifié au fur et à mesure de la génération de la séquence. Nous devons donc vérifier que le test est décidable et qu'il n'y a pas de chemin fatal.

7.3.2 VALIDITE DU TEST

Domaine de variation $\mathbb{P}$i du poids affecté au sommet qi

Soit (qo, r(1 , n), qi) un chemin de longueur n conduisant à qi tel que si (qo, r(1 , n-1), qj) avec r(1 , n) = r(1 , n-1) r(n) alors qj $\neq$ qi

Calculons la valeur pi après application de ce chemin.
En quittant un sommet, le poids qui lui est affecté est mis à 0. Donc, seul le sommet atteint par le chemin considéré a un poids différent de 0.

Si (qo, r(1, n-1), qj); qj ≠ qi ⟶ pi = 0

Le passage à qi se fait alors par un chemin unitaire (un arc)

αji =(qj, r(n), qi)

ce qui implique qu'à l'issue du chemin (qo, r(1 ,n), qi) le poids pi = 1.
pi est incrémenté à chaque chemin unitaire (qi,r,qi).
La valeur maximale de pi est donc Max (Max pik); (règle b)
donc, $\mathbb{P}$i = | 1, Max (Max pik)|

Il n'y a pas de chemin fatal

Comme dans B, il n'existe aucun sommet absorbant en dehors de l'élément terminal qf, il existe au moins un arc α ik = (qi, rij, qj) qj ≠ qi
et un arc α'ik = (qi, rif, qf) avec rij = rif.

D'après les règles précédentes, la réunion des intervalles

Iik $\cup$ I'ik = $\mathbb{N}$

Pour l'ensemble des c arcs d'origine qi et d'extrémité qj ≠ qi, nous avons donc: $\bigcup_{k=1}^{c}$ Iik $\supset \mathbb{N}^+ \supset \mathbb{P}$i

D'après la proposition du paragraphe 1.2, il n'y a donc pas d'évolution fatale.

Le test est décidable

Les règles a, b, c précédentes montrent que:

$\forall$ qi, qj $\in$ Q

Si à l'arc (qi, rik, qj) est associé Iik, il existe un arc (qi, rik, qf) auquel est associé l'intervalle $\mathbb{N}^+$ -Iik.

Comme il n'y a pas d'autre arc d'origine qi et d'étiquette r,
Iik $\cap$ I'ik = $\emptyset$, ce qui entraîne, d'après la proposition du paragraphe 1.2 que tous les automates de la famille générée par Bp sont déterministes.

Le test à partir des séquences de C.R. non acceptables, obtenu pour une séquence de travail répétitive, est donc décidable.

7.3.3 INTERPRETATION DE LA PONDERATION

Il est clair que le calcul de pi correspond au nombre d'itérations de rii $\in$ R que l'on peut observer si la P.O. est dans l'état qi. A la période d'échantillonnage près, pi représente la durée de maintien dans l'état, caractérisé par le temps écoulé entre deux variations de C.R.. Les valeurs Min et Max utilisées pour définir les prédicats sont donc des tolérances admissibles sur la mesure de ces temps. Cette tolérance est nécessaire pour tenir compte de l'influence de perturbations.
Tout changement d'état vers qi provoque une initialisation des poids telle que pi = 1 et pj = 0 si qj $\neq$ qi. Ceci revient à considérer tout changement d'état comme un instant de "recalage" du modèle. Si cette proposition est réaliste pour la grandeur mesurée qui change de point singulier, elle n'a pas de sens physique pour les autres grandeurs. L'application des règles de pondération proposées sont donc valables, s'il n'y a pas (ou peu) d'actions simultanées.

En cas de parallélisme, il faut pouvoir décomposer la P.O. en sous ensembles modélisables tels que le parallélisme y soit inexistant.

Remarque

La solution qui consisterait à prendre comme origine de temps le départ de qo afin d'éviter la décomposition est à rejeter. En effet, l'absence de recalage va provoquer une dérive du modèle par rapport à la réalité, correspondant à la somme des incertitudes.

Lorsqu'il y a parallélisme, il peut y avoir, pour certaines configurations, plusieurs arcs d'origine qi qui ne conduisent pas à qf.

Soit $\alpha 1, \ldots, \alpha n$ l'ensemble des arcs d'origine qi et d'extrémité différente de qf. S'il existe deux arcs αj, αk de cet ensemble pour lesquels $Ij \cap Ik \neq 0$, alors pour toute configuration pour laquelle $pi \in Ij \cap Ik$, il existe au moins deux substitutions acceptables.

L'amélioration apportée au test repose sur l'hypothèse de reproductibilité du comportement de la P.O. soumis à une séquence de travail elle-même répétée.

Etude qualitative des performances

L'hypothèse de la séquence de travail répétitive a permis de générer la séquence de C.R. à partir d'un automate déterministe.
L'introduction de la pondération permet de détecter les erreurs dues à une variation notable de la vitesse d'évolution d'une grandeur mesurée. En validation de mesure, ceci est connu sous le terme de cohérence de gradient.

Par ces ajouts, nous avons étendu le domaine du test en ligne aux défaillances de l'ensemble des éléments de la P.O. susceptibles de modifier notablement les C.R. dans un cycle de travail répétitif.
Le temps de latence est, de plus, diminué, ce qui réduit le risque d'erreurs multiples.

Les performances seront étudiées de façon plus précises dans la suite de ce mémoire. Nous allons maintenant envisager dans quelles conditions nous pouvons obtenir un niveau de performances comparable, en nous affranchissant de la contrainte de cycle répétitif.

7.4 PRISE EN COMPTE DU TEMPS DANS LES RELATIONS CAUSALES COMMANDE - COMPTE RENDU

Il est tentant d'introduire un modèle pondéré à partir de l'automate $\mathcal{D}$ ' à l'image de la démarche faite dans le cadre de la séquence de travail répétitive. Malheureusement, la règle de calcul du poids adoptée, qui introduit en fait la notion de temps passé dans un état, ne peut pas être reprise ici.
En effet, tout changement d'état correspond au passage d'un point singulier à un autre pour une grandeur mesurée. Ceci ne constitue pas un événement pour les autres grandeurs, il ne peut donc être question ici de considérer un changement d'état, dans un modèle global, comme un instant de recalage pour l'ensemble de la P.O..

7.4.1 DECOMPOSITION DE LA P.O. - INTERACTION ENTRE GRANDEURS MESUREES

L'adoption d'un automate pondéré comme modèle nécessite la définition d'une règle de calcul du poids. Ceci implique l'existence et la connaissance d'une relation entre la commande appliquée et l'évolution constatée.

L'élaboration du poids se trouve facilitée si nous pouvons admettre, comme précédemment, que chaque modification d'état est un instant de recalage. A partir de la notion d'état adoptée,une telle évolution correspond à un changement de point singulier pour une grandeur mesurée.

Nous adopterons alors l'hypothèse de séparabilité des grandeurs mesurées. Dans ce cadre, chaque trajectoire est modélisée séparément; un changement de point singulier, donc d'état, peut alors être considéré comme un instant de recalage.
Cette hypothèse est restrictive, mais certains types d'interactions entre grandeurs mesurées sont envisageables, sans remettre en question la clause du point de régénération au changement d'état. Nous citerons:

- le partage de capteurs entre plusieurs trajectoires,
- la modification d'une trajectoire par une autre grandeur mesurée dans certaines conditions que nous analyserons ultérieurement.

7.4.2 MODELISATION D'UNE TRAJECTOIRE PAR UN AUTOMATE PONDERE

Il a été défini au chapitre V un automate $\mathcal{A}' = (R, Q, \delta, qo, qf)$ comme modèle de la trajectoire $\mathcal{E}_j$ relative à la jème grandeur mesurée.

Comme pour l'automate B, défini dans le cadre du cycle de travail répétitif, le codage de l'indicateur de défaut dj est repris.
En fait, $\forall\, r \in Rj$ et $\forall qi \in Qj$, $qi \neq qjf$, le codage de dj est tel que $\exists\, (qi\ r\ qjf) \in \delta$.
Nous obtenons alors un automate $\mathcal{A}$ ref de référence pour lequel nous définissons des règles de calcul de poids.

A chaque arc $\alpha ik = (qi, rik, qk)$ est affecté un prédicat $V(\alpha ik)$ tel que:

$$V(\alpha ik) : \quad pi \longrightarrow v(\alpha ik) \in \{0,1\}$$

où pi est le poids affecté au sommet qi,

$v(\alpha ik) = 1$ si $pi \in Iik$

$v(\alpha ik) = 0$ sinon.

Ici toutefois, l'intervalle Iik est pris dans l'ensemble des réels.
Comme précédemment, seuls les arcs tels que $v(\alpha ik) = 1$ sont conservés.

Cette règle est applicable uniquement pour des sommets dont le poids est déterminé. Nous imposons $v(\alpha ik) = 0$ lorsque pi n'est pas déterminé.

Détermination du poids pi associé à qi $\in$ Qj

Soit un arc $\alpha jk = (qj, rjk, qk) \in \delta j$. Toute évolution selon cet arc αjk modifie le poids pi affecté au sommet qi de la façon suivante:

a) si qj $\neq$ qi; qk $\neq$ qi alors pi est inchangé;

b) si qj = qi; qk $\neq$ qi alors pi indéterminé;

c) si qk = qi, il existe une expression $f(\alpha ik)$ telle que
$f(\alpha ik)$: pi, c $\longrightarrow$ pi; avec pi = pio si, de plus, qj $\neq$ qi.

Dans ces expressions, c représente la commande; pio représente une valeur initiale.
$f(\alpha ik)$ est une expression de récurrence.

A l'initialisation, seul le poids du sommet initial est déterminé.

Interprétation physique du poids

Le poids représente une évaluation (interpolation) de la valeur prise par la grandeur mesurée entre les extrémités du point singulier correspondant à qi. Cette estimation est faite en valeur relative par rapport à une origine que nous choisissons arbitrairement au centre du point singulier.
pio représente la distance qui sépare l'entrée du point singulier à son centre.

Remarques

- pi est indéterminé lorsque l'on quitte le sommet qi. La valeur de pi n'a alors aucune importance pour le test.
- L'interpolation par $f(\alpha ik)$ correspond à l'hypothèse du fonctionnement en régime établi. Ceci suppose que le régime est établi ou que le temps d'établissement de ce régime est faible, vis-à-vis de la durée nécessaire pour parcourir le point singulier.
L'influence des transitoires est alors prise en compte dans l'incertitude représentée par les intervalles Iik utilisés dans les prédicats, au même titre que les perturbations.

- Une modélisation faisant appel aux équations différentielles qui relient commandes et grandeurs mesurées est incompatible avec l'hypothèse du sommet considéré comme point de régénération. Ceci constitue en fait, la limite du test par analyse syntaxique. En dehors de cette hypothèse, il faut recourir à des méthodes de simulation des systèmes continus.

- Cette interprétation physique suppose également que les modifications de la commande à l'intérieur du point singulier sont en nombre limité. En effet, dans le cas contraire, l'incertitude sur la valeur estimée peut conduire à une incohérence sur le test.

Exemple

Soit (pi)1, (pi)2, ..., (pi)n les n accroissements du poids pi obtenus sous n commandes distinctes. En supposant que l'incertitude relative sur la localisation soit constante et notée $\delta = \frac{\Delta pi}{pi}$.

Nous avons pi = pio + (pi)1 + (pi)2 + ... + (pi)n et

$$\Delta pi = \Delta(|(pi)1| + |(pi)2| + ... + |(pi)n|).$$

Si les $(pi)_j$ sont alternés par exemple, alors $\frac{\Delta pi}{pi}$ augmente et peut rendre pi non significatif.

- Recaler le modèle sur le système réel est une nécessité pour tout mécanisme de test utilisant un modèle de référence. Lorsque cette hypothèse n'est pas faite, il faut prévoir un mécanisme qui permette de compenser l'inévitable décalage entre modèle et P.O.. On trouve dans |RAU-84| une proposition interessante dans le cas des systèmes continus.

Détermination des Iik

Nous nous limitons ici aux arcs d'origine différente de qo. Le problème de la localisation est traité à part.

a) Soit un arc $\alpha ik = (qi, r, qk)$ avec qk ≠ qf
Iik est de la forme | Min ik, Max ik |

Cet intervalle représente l'incertitude admise sur l'interpolation.
Il est représenté par la proposition suivante:

Si L est la distance relative de l'extrémité du point singulier vers laquelle le système évolue et ΔL l'incertitude admise, alors Iij s'exprime:

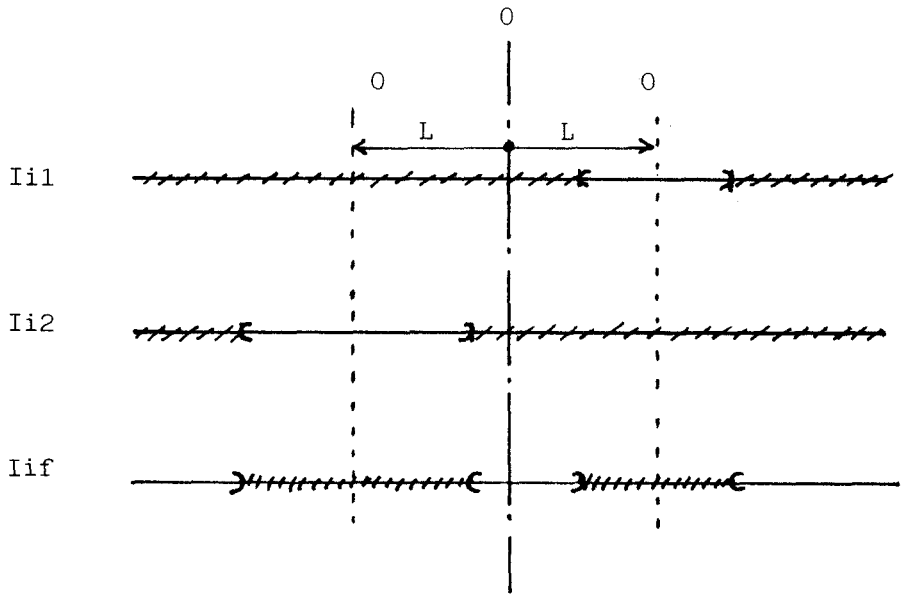
$$Iik = | L - \Delta L, L + \Delta L |.$$

Ceci peut être généralisé à l'ensemble des arcs d'origine qi et d'étiquette r.

Compte tenu du codage et de la définition des points singuliers, nous savons qu'il existe au plus deux arcs d'origine qi, de même étiquette r, et d'extrémités différentes autres que qf.

Soit $\alpha i1 = (qi,r,q1)$ et $\alpha i2 = (qi,r,q2)$ ces arcs et Ii1, Ii2 les intervalles correspondants.
Il existe également, après modification de σj, un arc $\alpha if = (qi,r,qjf)$ de même étiquette.
Nous définissons l'intervalle Iif par (figure 7.3):

$$Iif = \int_{\mathbb{R}} (Ii1 \cup Ii2).$$



- figure 7.3 -

b) Soit $\{\alpha i1, \alpha i2, \ldots, \alpha ip\}$ l'ensemble des arcs d'origine qi qui conduisent à un sommet autre que qi ou qf. Soit ri1, ... , rip les étiquettes correspondantes.
$\forall j \in [1,p]$, et $j \neq i$, s'il existe un arc (qi,rii,qi), son étiquette rii est différente de rij. Ceci découle de la définition des points singuliers.
Nous avons alors:

$$Iii = \left[\begin{matrix} \text{Min (Min ij)}, & \text{Max (Max ij)} \\ j=1 \longrightarrow p & j=1 \longrightarrow p \end{matrix} \right]$$

Cette proposition stipule que, tant qu'il est impossible d'affirmer que la grandeur mesurée n'est plus dans le point singulier, le C.R. noté rii est acceptable.

$\exists \alpha if = (qi, rii, qjf)$ pour lequel

$$Iif = \complement_{\mathbb{R}} Iii.$$

c) Pour tout $r \in Rj$ pour lequel il n'existe pas d'arc d'origine qi et d'extrémité différente de qf, il existe un arc $\alpha if = (qi,r,q f)$ tel que $Iif = \mathbb{R}$

N'ayant pas défini de règles de pondération pour les arcs d'origine qo, nous pouvons définir un sous graphe pondéré de l'automate recherché:
$Gp = (R, Q2, V, \delta)$ où $Q2 = Q - \{qo\}$.
Nous étudions les propriétés de ce graphe.

7.4.3 PROPRIETES DU GRAPHE PONDERE

Proposition

Il n'y a pas de chemin fatal.

Soit $\{\alpha i1, \alpha i2, \ldots, \alpha in\}$ l'ensemble des arcs d'origine qi, de même étiquette r. Compte tenu de la définition de l'automate de référence $\mathcal{A}$ ref et des prédicats retenus, nous pouvons dire qu'il existe un arc $\alpha if = (qi,r,qf)$ tel que $(\bigcup_{j=1}^{n} Iij) \cup Iif = \mathbb{R}$.

Décidabilité du test

Rappel: Pour que ce sous graphe Gp soit déterministe, quelque soit la configuration, il suffit que l'intersection de tous les intervalles correspondants aux arcs d'origine qi et de même étiquette r soit vide.

Cette propriété doit être vérifiée pour tout qi $\in$ Q2.

Cette proposition a été démontrée dans le cadre général des automates pondérés.

Elle est obligatoirement satisfaite en cas d'application des règles b et c. Dans le cadre de la règle a, il faut:

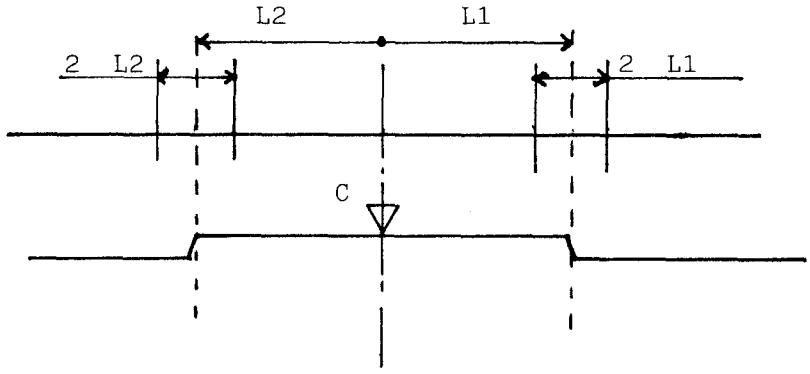
$$Iif \cap Ii1 \cap Ii2 = \emptyset$$

Par définition de Iif, cette condition est satisfaite si

$$Ii1 \cap Ii2 = \emptyset$$

Proposition

De tout automate généré, il est possible d'extraire un sous graphe déterministe défini sur Q2, si la valeur moyenne des incertitudes est inférieure à la "longueur" du point singulier (Fig 7.4).



- figure 7.4 -

D'après la figure, il faut $L1 + L2 > \frac{2\,\Delta L1 + 2\,\Delta L2}{2}$

S'il n'en n'est pas ainsi, il faut effectuer une transformation du graphe telle qu'elle a été présentée précédemment. Ceci introduit des sommets ambigus dont l'existence est néfaste pour la qualité du test. Nous supposons, dans la suite de l'exposé, que cette condition est satisfaite.

Remarque

Il est possible de s'affranchir de cette contrainte en définissant, à partir de Iij, deux intervalles $I^{+}ij$, $I^{-}ij$ et en introduisant la notion de sens de déplacement lié à la commande.
Si l'on considère qu'il est impossible de se déplacer dans un sens opposé à celui imposé par la commande, alors l'un de ces intervalles est toujours nul.
Si l'on considère $I^{+}i1$, $I^{-}i1$, $I^{+}i2$, $I^{-}i1$, nous avons pour un déplacement de sens positif:

$$I^{+}i1 \neq \emptyset, \quad I^{+}i1 = \emptyset \quad \text{donc} \quad I^{+}i1 \cap I^{+}i2 = \emptyset$$

Pour le sens opposé, nous avons de même $I^{-}i1 \cap I^{-}i2 = \emptyset$.

Cet aménagement permet d'éviter l'hypothèse simplificatrice ci-dessous. Dans les deux cas, cela interdit les phénomènes oscillatoires d'amplitude suffisante pour provoquer des changements de points singuliers.

7.4.4 LOCALISATION DU MODELE

Nous avons introduit la pondération au niveau du sous graphe de $\mathcal{A}$ ref, limité aux sommets de l'ensemble Q2 formé des éléments de Q autres que qo. Comme précédemment, nous devons préciser l'évolution du modèle à partir d'une initialisation.
Compte tenu de l'hypothèse faite sur l'incertitude dans l'interpolation de la position, nous avons vérifié que tout sous graphe, limité aux éléments de Q2 généré à partir du graphe pondéré, est déterministe. L'intérêt d'une telle propriété pour le test a déjà été discuté. Nous pouvons donc étudier les conditions pour que le modèle de la P.O. soit localisable. Nous adoptons la définition suivante.

Modèle localisable

Le modèle pondéré de la P.O. est localisable s'il existe une séquence qui puisse être générée par une P.O. saine de façon exclusive par des chemins de $\mathcal{A}$ ref, d'origine qo et de même extrémité qi $\in$ Q2 et s'il est possible d'affecter à qi un poids pi.

Cette définition impose un certain nombre de transformations du modèle.

Constat

Après une initialisation du modèle, la position à l'intérieur du point singulier ne peut être déduite par simple observation des C.R.. Nous devons donc admettre la règle suivante:

Conséquence

Toute évolution dans l'automate pondéré, à partir d'un arc d'origine qo ne permet pas de déterminer un poids pi au sommet atteint qi, qui soit une interpolation de la mesure. Le poids du sommet atteint reste donc indéterminé.

De plus, si le codage introduit un non déterminisme dans $\mathcal{A}'$, l'évolution est ambiguë pour tout chemin d'origine qo. La transformation de $\mathcal{A}'$ en graphe déterministe se fait en regroupant des états de $\mathcal{A}'$.
Il est alors impossible d'admettre de façon générale un poids unique, représentant la distance de l'entrée du point singulier par rapport au centre, pour un tel groupement de sommets correspondant à des points singuliers distincts.

Régle: Seules les évolutions selon un arc dont l'extrémité qj et l'origine qi sont des éléments de Q2 permettent d'affecter un poids significatif au sommet atteint.

Pour compléter le modèle précédent de façon à y inclure la phase de localisation, nous faisons appel à l'étude faite dans le cadre de la modélisation avec prise en compte de la commande.
Nous supposons qu'il est possible de créer l'automate déterministe $\mathcal{D}' = (R * C, Q', \delta, qo, qf)$ défini sur l'alphabet R * C vu au chapitre précédent, mais limité ici à la modélisation de la jème trajectoire, qui est l'objet de l'étude.
Ce graphe inclut la phase de localisation. La trajectoire est donc supposée localisable au sens du chapitre VI.

Pour transformer cet automate en un automate pondéré $\mathcal{D}^P = (R, V, Q', \delta^P, qo, qf)$ ayant le même comportement, il suffit d'adopter les règles suivantes:

a) Les commandes sont classées en fonction du sens de l'évolution qu'elles imposent sur la trajectoire.

b)-Toute évolution selon un arc de δ dont l'étiquette correspond à une commande positive (sens d'évolution positif) impose un poids égal à +1 au sommet extrémité.

- Si la commande est de sens opposé, le poids est de -1. Dans tous les cas, le sommet origine prend le poids 0 s'il est distinct de l'extrémité.

c) A tout arc dont l'étiquette correspond une commande positive est associé un intervalle | +1 |. Dans le cas contraire, cet intervalle est | -1 |.

L'automate pondéré $\mathcal{E}^p$ = (R, V, P, γ, qo, qf) recherché est obtenu à partir de $\mathcal{D}^p$ et du sous graphe pondéré réduit aux sommets de Q2 tiré de $\mathcal{E}^p$.

Soit G^p = (R, Vg, Q2, δ) le sous graphe pondéré limité aux éléments de Q2 $\in$ Q

Soit $\mathcal{D}^p$ = (R, Vd, Q', δ^p, qo, qf) l'automate de localisation défini ci-dessus.

Par hypothèse, qf $\in$ Q' et qf $\in$ Q2

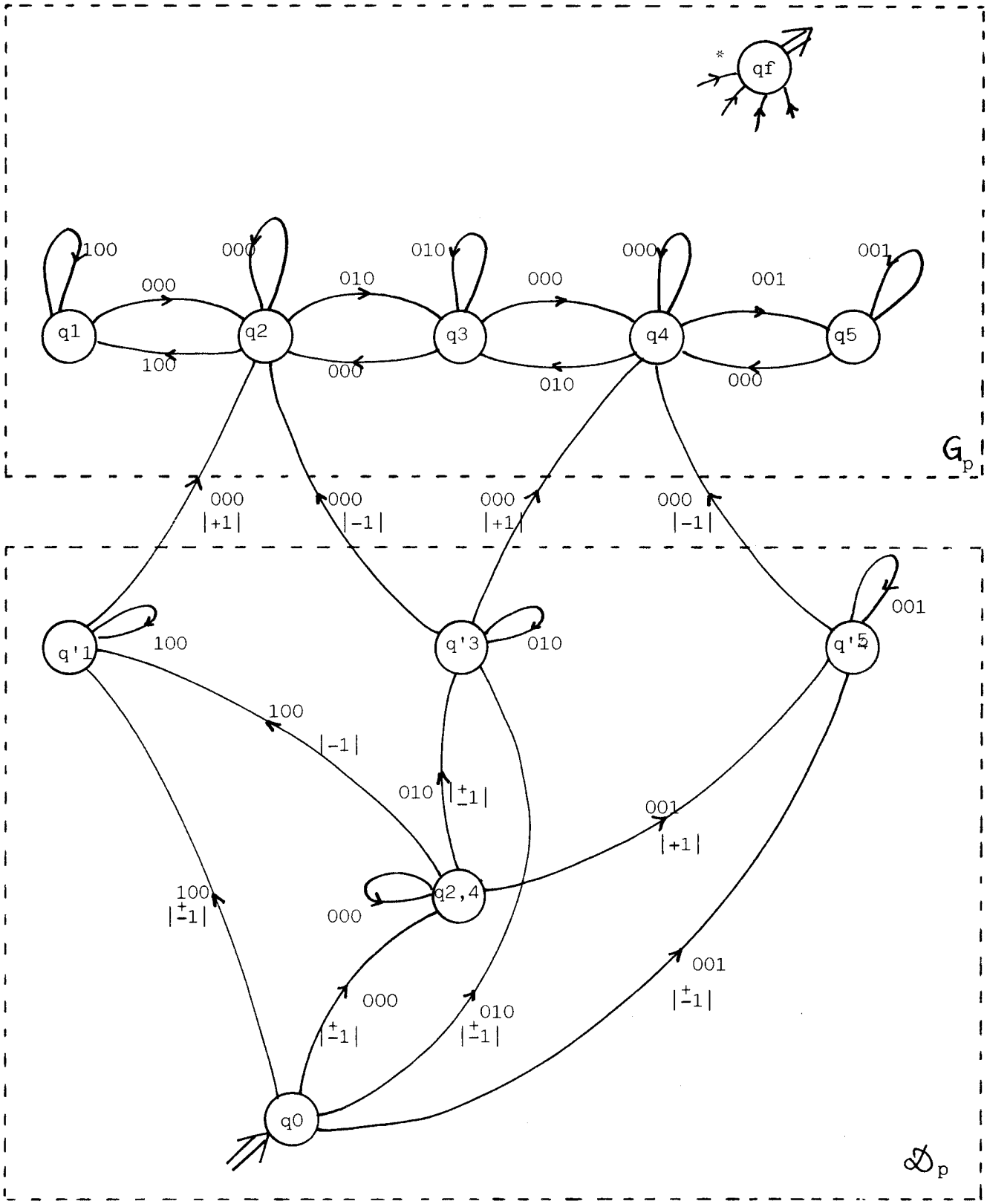
L'automate pondéré $\mathcal{E}^p$ = (R, V, P, γ, qo, qf) qui modélise la P.O. est construit comme indiqué ci-dessous.

- L'ensemble P des sommets est Q' $\cup$ Q2.
- $\mathcal{D}$ étant supposé localisable, il existe pour chaque point singulier un sommet de Q' qui représente ce point singulier et lui seul. Par analogie avec Q2, nous notons Q'2 cet ensemble.
- Notons qi $\in$ Q2 et q'i $\in$ Q'2 les représentants, dans chaque graphe, du point singulier i.

Les connexions entre les graphes sont réalisées de la façon suivante:

tout arc α'ij = (q'i,r,q'j) $\in \delta^p$ tel que q'i, q'j $\in$ Q'2 est remplacé par un arc αij = (q'i,r,qj) $\in \gamma$. Le prédicat V(α'ij) est affecté à ce nouvel arc.
Un tel arc permet d'affecter à l'extrémité qj un poids pjo significatif des "dimensions" du point singulier atteint.
Tout chemin d'origine qo qui empreinte un tel arc constitue une phase de localisation.

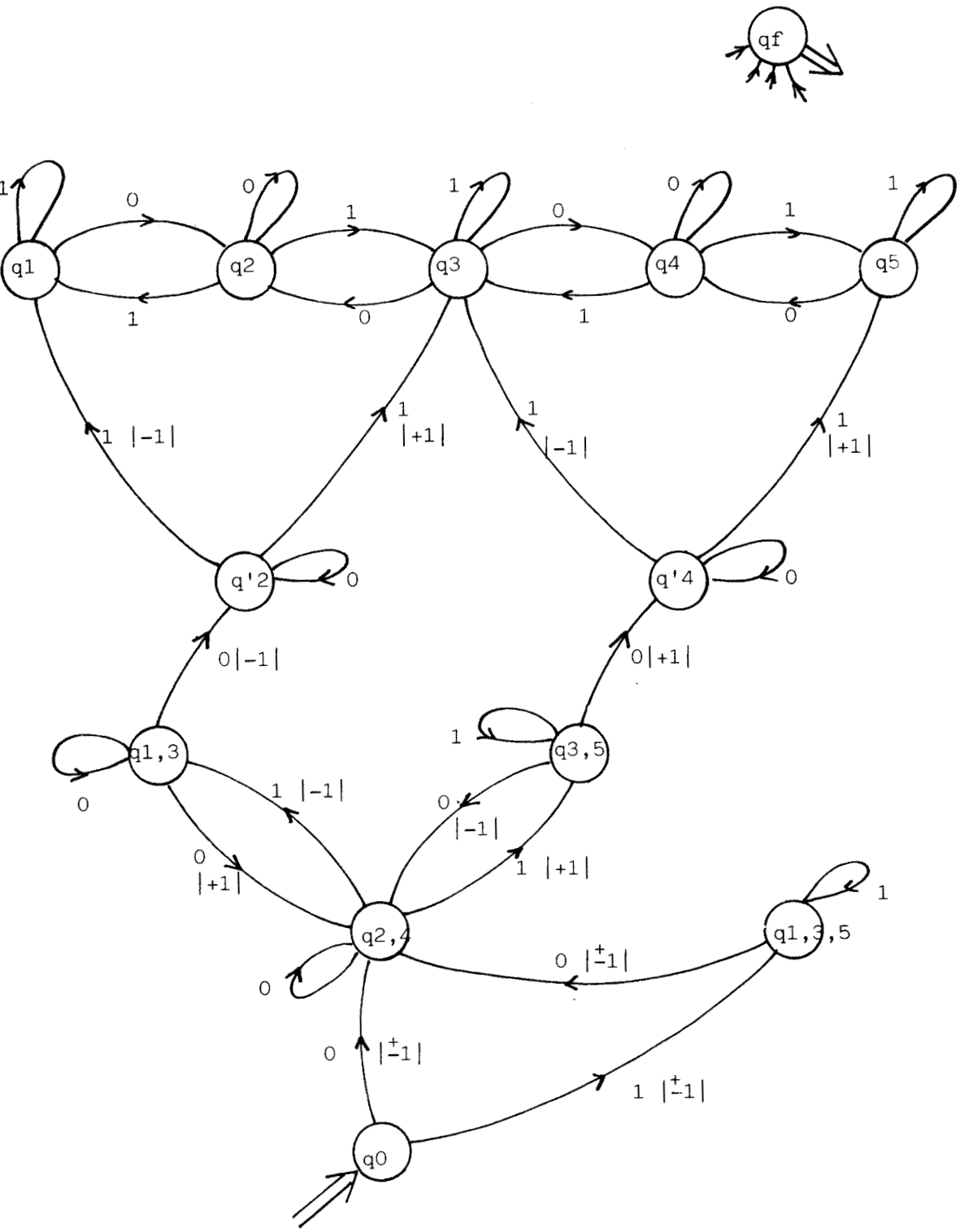
Nous donnons ci-après deux exemples de réalisations correspondants à ceux des figures 5.3 et 6.4.



Automate pondéré modélisant la trajectoire d'une came ponctuelle devant trois capteurs (fig. 5.3).

* Les arcs vers qf ne sont pas représentés pour alléger le graphisme.

- figure 7.5 -



Capteur incrémental (à trois incréments), en trajectoire ouverte.

- figure 7.6 -

Remarque

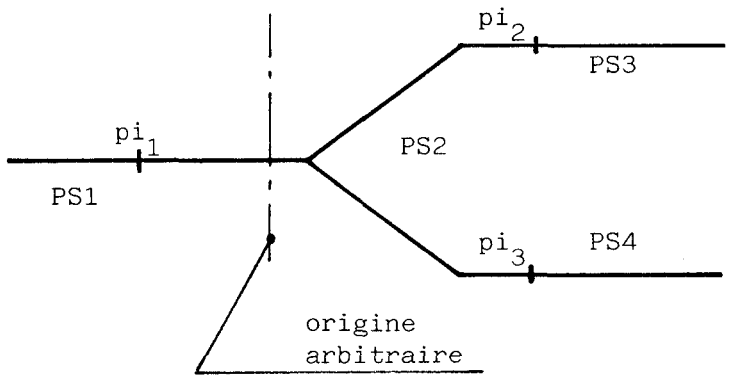
Si le modèle n'est pas localisable, au sens où il est impossible de trouver un automate $\mathcal{A}'$ tel que Q'2 $\subset$ Q', il est alors à priori impossible de créer l'automate pondéré $\mathcal{E}^p$ tel que nous l'avons défini.

Si nous considérons l'automate relatif à la came rotative (Fig. 6.6), nous constatons qu'un modèle pondéré est acceptable si l'on considère le modèle localisé par les sommets q1,5; q2,6; q3,7; q4,8.

En fait, cette hypothèse est justifiée par l'existence d'une symétrie. En toute rigueur elle ne l'est plus si les bossages de la came sont décalés.

Remarque

Nous avons considéré des points singuliers ayant une entrée à chaque extrémité. Il est clair que ceci peut être étendu à des éléments de trajectoire comportant des bifurcations (Fig 7.7).



- figure 7.7 -

Il suffit alors de définir les valeurs initiales du poids en fonction de l'entrée concernée.

7.5 MODELE TEMPORISE

Le modèle précédent s'avère difficile à mettre en oeuvre, essentiellement à cause des règles de calcul du poids. Nous proposons une transformation visant à remplacer l'interpolation de la mesure par une information temporelle, qui peut faire l'objet d'une acquisition par apprentissage.

Signification du poids

L'objectif est d'utiliser un modèle pour lequel le poids pi corresponde à la durée de maintien dans l'état qi.
Cette représentation se heurte à deux problèmes essentiels qui sont:

- le temps ne peut pas rendre compte des évolutions réelles sur la trajectoire,
- il ne dépend pas de l'extrémité par laquelle le point singulier a été atteint.

Cette constatation nous conduit à adopter l'hypothèse suivante.

Hypothèse de commande maintenue

La commande est maintenue pendant toute l'évolution interne à un point singulier. Toute modification de commande est supposée intervenir à l'instant du changement de point singulier.

Cette hypothèse est tout à fait cohérente avec le comportement normal de la commande. Les ordres sont effectivement modifiés à l'issue d'un changement de C.R..

L'utilisation du temps en guise d'interpolation conduit également à une difficulté liée à la suppression de la référence spatiale.

Soit un sommet qi accessible avec des poids initiaux -pio et +pio selon l'extrémité par laquelle le point singulier est atteint.
Soit c une commande maintenue telle que pi évolue par valeur croissante.

Si qi avait été atteint avec la valeur +pio, avant application de c,

$$pio \leqslant pi \leqslant pio + \Delta pi$$

Si qi avait été atteint dans les mêmes conditions avec pi = -pio, alors

$$-pio \leqslant pi \leqslant +pio + \Delta pi$$

L'utilisation du temps nous conduit à considérer la valeur initiale du poids comme origine de comptabilisation de celui-ci.
Dans le permier cas, il est donc normal de quitter qi sous l'effet de la commande c avec un poids compris entre 0 et + Δpi.
Dans le deuxième cas, l'intervalle associé à l'arc qui fait quitter qi est $[2pio - \Delta pi, 2pio + \Delta pi]$.

Pour chaque arc, nous avons deux intervalles possibles selon qu'il y a eu ou non changement de sens de déplacement sur le point singulier.

Par contre, la valeur de pi à l'entrée dans qi est dans tous les cas pi = 0.

Perte de déterminisme

Soit un sommet qi tel qu'il existe dans $\mathcal{A}'$ deux arcs de même étiquette conduisant à deux sommets qj, qk distincts.
Par définition, ces deux sommets sont situés de part et d'autre de qi. Ils sont accessibles, à partir de qi, par deux commandes distinctes. Les intervalles Iij et Iik sont respectivement centrés sur -pio, +pio et ils sont disjoints par hypothèse. La suppression de la référence absolue rend alors les nouveaux intervalles susceptibles d'être non disjoints.

Pour conserver à l'automate pondéré son déterminisme (au niveau du sous graphe bâti sur Q2), il suffit de tenir compte de la commande dans la définition de l'intervalle.

Pour une commande c1 telle que l'évolution selon (qi,r,qj) est possible, nous posons:

$$Iij,c1 = Iij$$

Pour une commande c2 correspondant à une évolution dans l'autre sens nous prendrons:

$$Iij,c2 = \emptyset$$

Le problème de la localisation ne subit aucune modification puisque dans cette phase, la pondération n'a pas la même signification.

Introduction du temps

Notons $vc = \frac{d\,pi}{dt}$ la vitesse d'évolution dans le point singulier PSi sous l'effet de la commande c.
Nous avons alors:

$$2\,pio = \int_0^{Ti} vc\,dt \qquad \text{de même} \quad \Delta pi = \int_0^{+\Delta t} vc\,dt$$

Ti représente alors le temps mis pour "traverser" le point singulier sous l'effet de c et Δt l'incertitude sur ce temps.

Pour chaque vitesse c, nous avons alors deux intervalles de temps

$$I^1ik,c = [0,\ \Delta t] \ ; \quad I^2ik,c = [T - \Delta t,\ T + \Delta t] .$$

De même, pour tout arc $\alpha ij = (qi\ r\ qi)$ entraînant une modification de poids

$$f(pi, c) \longrightarrow pi,$$

nous pouvons écrire, si vc est une vitesse moyenne sur PSi

$$pi(n+1) = pi(n) + vc*t \quad \text{avec } t = 1 \text{ unité de temps.}$$

Nous obtenons alors en posant

$$ti = \frac{pi}{vc}$$

$$ti(n+1) = ti(n) + 1$$

relation indépendante de la vitesse.

Récapitulatif

A chaque sommet est affecté un poids ti tel que:

- l'arrivée en qi charge ti = 1;
- le départ de qi met ti à 0;
- toute évolution selon un arc (qi,r,qi) provoque l'incrémentation de ti.

La règle de calcul de poids est donc identique à celle utilisée pour le modèle pondéré adopté dans le cadre du cycle répétitif.

A chaque arc (qi,r,qj) avec qi ≠ qj; qj ≠ qf et pour chaque commande c correspond deux intervalles:

$$I^1_{ij,c} = \left[Min^1\ ij,c;\ \ Max^1\ ij,c \right]$$

$$\text{et } I^2_{ij,c} = \left[Min^2\ ij,c;\ \ Max^2\ ij,c \right]$$

Le permier est utilisé s'il n'y a pas eu de changement de sens d'évolution lors de l'arrivée sur qi. Dans le cas contraire, c'est $I^2_{ij,c}$ qui est retenu.
Le modèle retenu est donc un automate fini associé à une temporisation de chien de garde.

Conclusion

Quel que soit le moyen adopté pour la prise en compte de la commande, l'introduction d'un chien de garde améliore les performances du test en réduisant l'intervalle pendant lequel un C.R., acceptable compte tenu de la commande et de la géométrie de la trajectoire, est accepté.

Les blocages, les départs intempestifs et les variations notables de la vitesse d'évolution sont détectables.

Ces tests, incluant la prise en compte de la commande constituent donc une amélioration des procédés proposés au chapitre VI.

La décomposition de la P.O. en ses trajectoires, jugée nécessaire ici, n'est pas une contrainte vraiment spécifique. En effet, il est peu raisonnable d'espérer définir pratiquement un modèle direct de la P.O. dès le test dynamique. Même dans le cas de la P.O. prise dans le cadre du cycle de travail répétitif sans temporisation, la décomposition peut s'avérer nécessaire s'il y a un niveau de parallélisme certain, pour éviter une croissance exponentielle du modèle.

La taille du modèle relatif à une sous P.O. croît comme le produit du nombre de P.S. par le nombre de commandes.
Ceci a une implication directe, au moment de la réalisation, sur la place mémoire occupée.
Nous avons vu également que le passage d'un P.S. à un autre doit être effectué à commande maintenue et que l'influence du transitoire doit être négligeable (ou inclu dans l'intervalle associé).
Toutes ces restrictions montrent que cette méthode de test, mal adaptée aux applications relevant de la régulation des systèmes continus, est plus particulièrement orientée vers les systèmes logiques à évolution séquentielle.

Il nous reste maintenant à étudier de façon plus précise les performances attendues ainsi que les réalisations possibles. Il est certain que la gestion de ces modèles ne présente pas de difficultés majeures. Leur élaboration est par contre, plus délicate.

Cette deuxième partie de notre étude propose différents modèles pour lesquels le niveau de connaissance de la P.O., représenté par la quantité d'informations nécessaires, va croissant.

Le premier niveau, appelé test statique, est un simple test d'appartenance au code. Le taux de couverture, plus que modeste pour certains types de défauts, peut être amélioré par l'introduction d'informations supplémentaires. Cette approche, relevant de l'utilisation des codes détecteurs d'erreurs, a le défaut d'être pénalisante pour la fiabilité.

Le deuxième niveau, dit test dynamique, consiste à modéliser la P.O. par une grammaire régulière. La représentation des règles de production par un automate d'états finis permet de mettre en évidence les ambiguïtés qui résultent du codage des différents points de trajectoire. Pour rendre le test décidable, nous avons été obligés d'accepter un certain risque de masquage de pannes. Ces deux premiers niveaux sont présentés dans le chapitre 6.

L'introduction et l'utilisation d'un alphabet de commande permet d'inclure des informations sur le contexte de la P.O., tout en gardant la simplicité des grammaires régulières. Ce troisième niveau, appelé test dynamique dans le cadre de la commande, permet d'éliminer les ambiguïtés du modèle précédent en dehors de la phase de localisation, correspondant à la séquence de synchronisation (Homing Sequence).

Le dernier modèle, correspondant au test dynamique pondéré, fait appel à une grammaire programmée. Sa représentation utilise des automates pondérés dont la structure évolue en fonction d'un programme qui prend en compte à la fois la commande appliquée et le temps normalement nécessaire pour atteindre un nouveau compte rendu. Ce modèle, évidemment le plus performant, nécessite un niveau de connaissance de la P.O. important.

Les modèles statique et dynamique révèlent uniquement les défaillances des composants qui élaborent les comptes rendus (capteurs, transducteurs). La prise en compte du contexte permet de mettre en évidence des divergences entre l'évolution attendue d'une grandeur mesurée et son évolution réelle. Les deux derniers modèles assurent donc réellement la surveillance de l'ensemble des éléments hors automate et notamment de la P.O.. L'utilisation du test dynamique pondéré est limitée par deux contraintes qui doivent être respectées: la première suppose que le passage d'un point singulier à un autre s'effectue à commande maintenue et que le régime transitoire éventuel a une influence sur la durée de cette évolution. La seconde impose que la partie opérative soit décomposable en sous parties opératives indépendantes, correspondant à chaque grandeur mesurée.

Après cette étude théorique, nous envisageons l'intégration du mécanisme de test à l'automatisme et à l'évaluation quantitative de la sûreté de fonctionnement obtenue.

TROISIEME PARTIE

REALISATION ET PERFORMANCE D'UN DISPOSITIF DE TEST DE LA P.O.

INTRODUCTION A LA TROISIEME PARTIE

Nous abordons les aspects relatifs à la mise en oeuvre et aux performances du test en ligne de la P.O. .

La sûreté de fonctionnement dépend à la fois du taux de couverture du mécanisme de test et de l'architecture marérielle retenue. Ces deux aspects sont abordés respectivement dans les chapitres 8 et 9. Le dernier chapitre aborde le problème de l'exploitation temps réel du modèle retenu, mais aussi celui de la création de ce modèle.

CHAPITRE VIII

EVALUATION DES PERFORMANCES DES DIVERS MECANISMES DE TEST EN LIGNE

Il est difficile de mesurer l'efficacité des différentes méthodes de test en ligne proposées. Cette difficulté est liée, pour l'essentiel, au fait que nous ne sommes pas maître de la séquence de test.
En effet, l'évolution de la P.O. défaillante reste liée au cycle de travail imposé par la P.C., qui est eventuellemnt perturbé lui-même par la présence de l'erreur induite par la défaillance.

Nous nous proposons d'analyser pour différents types de défaillances:
- les possibilités de masquage;
- les évolutions qui permettent de révéler une erreur et le temps de latence qui en découle.

Ces évolutions doivent être envisagées simultanément dans le modèle utilisé et pour la P.O. réelle.

1 LES DEFAILLANCES ENVISAGEES

.1 TYPES DE DEFAILLANCES

Les défaillances peuvent être classées en deux catégories qui sont:
- les défaillances qui affectent le C.R. sans altérer l'évolution de la P.O.;
- les défaillances qui modifient la séquence de points singuliers, mais dont l'évolution réelle (donc anormale par hypothèse) est fidèlement traduite par le C.R..

Nous posons comme postulat qu'il n'existe pas de pannes simples qui induisent simultanément ces deux types de défaillances.

Remarque

Il va de soi qu'une défaillance peut, par réaction de la P.C., conduire à une évolution réelle différente de celle attendue. Cela doit être pris en considération et ne constitue pas une exception au postulat ci-dessus.

Défaillances qui affectent le C.R.

Ces défaillances affectent les capteurs et les interfaces capteurs-automate et capteurs - P.O. (cames, clavettes ...).
Ceci conduit à ce qu'il estconvenu d'appeler collage à "1" ou collage à "0" en ce qui concerne les capteurs tout-ou-rien. Il s'agit en fait ici d'un blocage.
Dans le cas d'une grandeur digitalisée (codeur de position, utilisation d'un convertisseur ...), le blocage peut être limité à un ou plusieurs bits. Ce blocage partiel perturbe malgré tout, globalement, la mesure.

Les pannes non catalyptiques telles que: modification du biais, du facteur d'échelle, introduction d'une dérive ... sont considérées comme faisant partie du deuxième type. En effet, elles conduisent à une modification des dimensions apparentes des points singuliers. Par cette remarque, nous admettons par avance que seuls les modèles pondérés permettent la détection de telles anomalies.

Les défaillances qui modifient la séquence de points singuliers

Ces défaillances prennent différentes formes qui sont:

- la substitution de commande,
- la modification de trajectoire.

Dans le premier cas, la commande réellement appliquée est différente de celle attendue. Une telle panne peut conduire à une modification du module et/ou du sens de la vitesse d'évolution. Ceci inclut les blocages et les départs intempestifs d'actionneurs. Les éléments concernés sont les préactionneurs et les actionneurs ainsi que toutes les liaisons entre ces éléments et l'automate ou la P.O..
Les ruptures d'organe d'entraînement comme leurs glissements peuvent être rangés dans cette catégorie de défaillances.

La modification de trajectoire regroupe les pannes dont l'effet s'apparente à une modification des dimensions des points singuliers. Ceci correspond aux pannes non catalyptiques des capteurs, recensées ci-dessus. Nous y trouvons notamment les glissements des cames, codeurs de position ..., dus à une défaillance des liaisons mécaniques.

.1.2 DEFAILLANCE ET ERREUR

Nous avons défini l'erreur comme la substitution d'au moins un élément de C.R., par rapport au C.R. qu'aurait généré la P.O. saine, placée dans les mêmes conditions.
Ceci nous amène à distinguer le temps de latence de l'erreur, mesuré à partir de l'instant d'apparition de la substitution, du temps de détection de la défaillance, mesuré à partir de l'instant où la panne est effective. Ce temps de latence est une caractéristique importante d'un mécanisme de test. Toutefois, il est possible qu'une erreur n'ait pas d'influence immédiate sur la commande. Ceci est particulièrement perceptible lorsqu'on utilise une machine réceptive pour décrire la commande. Une erreur latente ne correspond donc pas forcément à un fonctionnement réellement dangereux. Par contre, lorsque la défaillance affecte la séquence des points singuliers, il est possible d'avoir un fonctionnement dangereux sans qu'il y ait génération d'une erreur au sens rappelé ci-dessus. En effet, il est possible alors d'avoir substitution d'un élément de C.R. par un élément identique.
Aucune des solutions envisagées ne permet de s'affranchir de ce risque.

8.1.3 METHODE D'ANALYSE DES PERFORMANCES DU MECANISME DE TEST

L'évolution, dans l'automate utilisé pour le test, est liée à la séquence de comptes rendus observée. Une anomalie est détectée lorsque le chemin emprunté dans ce modèle passe par un élément terminal (noté qf $\in$ T).
Pour mettre en évidence le comportement du mécanisme de test en présence d'une faute, nous éliminons de l'automate utilisé les arcs dont l'étiquette est devenue irréaliste, compte tenu de la défaillance étudiée. Nous créons alors un modèle de comportement du test en présence de la faute considérée.

Cette méthode est à rapprocher de celle développée notamment dans |COU-72| pour définir les séquences de tests hors ligne des systèmes séquentiels.

Cette transformation, et l'analyse de performance qui en découle, est limitée ici au cas des pannes simples.

8.2 DEFAILLANCES AFFECTANT LE COMPTE RENDU

Nous posons par hypothèse que l'évolution réelle de la P.O. correspond à la séquence d'ordres générée par la P.C.. Seuls les C.R. reçus sont perturbés par la faute envisagée.

8.2.1 DEFINITIONS DES OUTILS D'EVALUATION DES PERFORMANCES

Le type de défaillance envisagée ici correspond au maintien à une valeur constante d'une variable binaire (ou éventuellement d'un ensemble de variables binaires). Ce défaut est désigné par collage à "1" ou à "0" de la variable correspondante.
Une telle défaillance correspond à une application d de l'espace des C.R. dans lui-même, définie par:

$\forall$ rj $\in$ R; $\exists$ rk $\in$ R tel que d: rj $\longrightarrow$ rk.

8.2.1.1 Modèle de comportement

Nous traitons ici du cas des automates non pondérés. L'introduction de la pondération dans le modèle de comportement est traitée ultérieurement.

a) Automate défaillant

Soit $\mathcal{A}$ = (R, Q, δ, qo, T) un automate.
Nous créons un automate défaillant $\mathcal{A}$d = (R, Q, δd, qo, T) de la façon suivante.

Soit un arc (qi, rj, qj). Compte tenu des propriétés des automates utilisés, et en particulier de leur déterminisme, il existe un arc et un seul (qi, rk, qk) dans δ pour lequel rk soit l'image par d de rj. Pour tout arc (qi, rj, qj) de δ tel que qi $\notin$ T, il est introduit dans δd un arc (qi, rk, qk).

Si qk $\in$ T, la défaillance envisagée est immédiatement détectable à partir de qi.

Si qk $\notin$ T, la défaillance est à l'origine d'une erreur non révélée. Ces arcs, dits arcs erronés, sont repérés sur les figures par un double trait. Tout passage par qi fait alors courir un risque de propagation de l'erreur que le mécanisme de test est incapable d'éliminer. De plus, une telle situation correspond à une perte de localisation du modèle.

b) Automate de révélation

Pour tenir compte du phénomène de perte de localisation, nous définissons un automate de révélation $\mathcal{A}$r = (R, Q, δr, qo, T).

δr est obtenu en ajoutant à δd les arcs obtenus par application de la règle ci-dessous.

Soit (qi, rk, qk) un arc erroné de δd. Tout arc de δd tel que (qi, rn, qn) d'origine qi donne naissance à un arc (qk, rn, qn) introduit dans δr.

Simplification des automates défaillant et de révélation

Parmi les arcs qui conduisent à un élément de T, il en est qui correspondent à deux événements simultanés. L'un est relatif à une évolution normale de la P.O., l'autre est lié à une défaillance.
La probabilité d'occurence de ces deux événements est considérée comme négligeable. Nous éliminons de δd et δr tous les arcs d'origine qi, conduisant à un sommet terminal dont l'élément de C.R., utilisé dans l'étiquette, n'est pas adjacent à l'élément rii de l'arc (qi, rii, qi) du modèle initial.

Remarque:
Cette simplification ne doit être appliquée à δd qu'après obtention de δr.

2.1.2 Evaluation qualitative

Séquence de révélation

Nous appelons séquence de révélation, pour la panne considérée, à partir de l'état qi, toute séquence générée par un chemin sans cycle, d'origine qi et d'extrémité qf $\in$ T, pris dans l'automate de révélation.

A un tel chemin correspondent une ou plusieurs évolutions réelles de la P.O.. Tant que la défaillance ne génére pas d'erreur, au sens envisagé ci-dessus, la séquence de commande reste normale.

Panne masquable

Une panne est masquable, s'il n'existe pas de séquence de révélation. Plusieurs cas peuvent se produire:

- il existe un sommet puits qi non final; tout chemin conduisant à ce sommet entraîne le masquage;
- il n'existe aucun arc conduisant à un élément qf de T. Dans ce cas, la panne est totalement masquée;
- il n'existe aucun chemin permettant d'atteindre les sommets origines des arcs d'extrémité qf. Il y a masquage si l'état de la P.O. ne correspond pas à ces sommets à l'instant d'apparition de la défaillance.

Evolution réelle

A partir d'une séquence de points singuliers (cohérente avec la géométrie des trajectoires) que nous appelons évolution réelle, il est possible de suivre le comportement du modèle contenant la défaillance.

Toute évolution générant une séquence incluant une séquence de révélation, augmentée éventuellement de cycles, conduit à une détection de défaillance.

Chaque fois que cette évolution dans le modèle emprunte un arc erroné, il y a génération éventuelle d'une erreur. Cette erreur est propagée si la commande est alors réceptive ou sensible à la variable erronée.

8.2.1.3 Evaluation quantitative

Probabilité d'être dans un état à l'apparition de la défaillance

Au moment où apparait un événement normal ou anormal, le modèle est dans un état supposé connu.
A chaque P.S.i, nous pouvons associer une probabilité P^*i correspondant à la fréquence de passage dans ce P.S. sur une période d'observation T. Cette probabilité peut être évaluée de façon assez précise là où les séquences de travail sont connues ainsi que les fréquences d'apparition de ces séquences. Une évaluation approchée peut être obtenue en prenant l'hypothèse que les P.S. sont atteints un même nombre de fois. Le problème de fréquence étant ainsi éludé, il reste à connaître la durée relative de maintien dans chaque P.S.. Pour les modèles qui incluent la commande, il est souhaitable de faire une évaluation séparée pour chaque commande.

Compte tenu de la correspondance entre les P.S. et les sommets $q \in Q$ du modèle, il est alors possible d'attribuer à qi une probabilité P_i exprimée à partir des P^*_i supposées connues.

P_i correspond alors à une probabilité asymptotique d'être dans l'état qi, pour le modèle choisi, en l'absence de défaillance.

Nous écrivons alors $P_i = \frac{ti}{T}$

où ti représente le temps moyen de présence dans qi.

T est tel que $\sum_{\substack{\text{ensemble} \\ \text{du modèle} \\ \text{localisé}}} P_i = 1$

L'étude porte alors sur le modèle localisé (réduit donc aux éléments du sous graphe G2 défini dans le chapitre précédent).

Le temps ti n'est pas sans rapport avec l'intervalle $I(\alpha_{ii})$ associé à l'arc (qi, rii, qi) dans le modèle temporisé.

Taux de couverture de la défaillance

Nous appelons taux de couverture Pcd la probabilité notée

Pcd= $\mathcal{P}$(défaillance soit détectée dés son apparition).

Elle est égale à la somme des probabilités associées à chaque sommet origine d'un arc d'extrémité $qf \in T$ dans l'automate défaillant.

Il peut être considéré comme un taux de couverture dans la mesure où il est divisé par la probabilité de l'événement certain (par hypothèse, il existe une défaillance à l'instant t).

Taux de couverture de l'erreur

Nous appelons erreur agissante, une substitutuion de C.R. qui entraine une transition vers un état du modéle défaillant n'appartenant pas à T.

Le taux de couverture de l'erreur, noté Pce, est représenté par:

Pce= $\mathcal{P}$(erreur agissante soit détectée dés son apparition).

Cette évaluation diffère de la précedente dans la mesure où le système peut continuer d'évoluer normalement si la défaillance n'entraine pas d'erreur agissante. Pce évalue en particulier, les risques de propagation de l'erreur vers la P.C..

Pce est égal au quotient de Pcd par la somme des probabilités associées aux sommets origines des arcs contenant une erreur autre que (qi, rii, qi). Cette exclusion est due au fait que cet arc ne correspond pas à une substitution de C.R. visible par la P.O.. Le calcul est fait à partir du modèle défaillant sans tenir compte des arcs dit "normaux", c'est-à-dire non affectés par la défaillance.

Taux de détection de l'erreur

Ce taux s'exprime par Pd=$\mathcal{P}$ (qu'une erreur soit détectable).
Après avoir calculé les probabilités d'être dans chacun des états non finaux de l'automate de détection, Pd est calculé en faisant:

Pd= somme des probabilités attachées aux sommets origines d'au moins un chemin d'extrémité qf T.

Ce taux évalue les probabilités de non masquage de l'erreur, en dehors de toute considération de temps. Le temps de latence n'est donc pas pris en considération.

Introduction de la pondération dans le calcul des taux de couverture

Par construction de l'automate défaillant, tout arc $\alpha ij=(qi,rj,qj)$ est transformé (ou confondu) en l'arc $\alpha ik=(qi,rk,qk)$ par la défaillance d: rj $\longrightarrow$ rk. A l'arc αik est associé, dans l'automate, l'intervalle $I(\alpha ik)$.
L'automate défaillant de référence est complété par l'integration de l'arc $\alpha' ik=(qi,rk,qf)$ auquel est associé $I(\alpha' ik)$ tel que:

$$I(\alpha' ik) \cup I(\alpha ik) = \mathbb{R} .$$

Soit pi le poids de qi au moment de l'apparition de la panne (ou de l'erreur selon le taux de couverture recherché).
Si pi $\in I(\alpha ik)$, l'erreur n'est pas détectée, il y a évolution selon l'arc erroné αik. La probabilité pour que cet événement se produise est donnée par

$$P_i^e = \frac{\Delta t(\alpha ik)}{ti}$$

où $\quad t(\alpha ik) = Max\ I(\alpha ik) - Min\ I(\alpha ik).$

La probabilité d'avoir évolution selon cet arc erroné est donc évaluée par

$$\frac{\Delta t\ (\alpha ik)}{ti} \times \frac{ti}{T} = \frac{\Delta t\ (\alpha ik)}{T}$$

La probabilité de détection immédiate à partir de qi est:

$$P_i^d = \frac{ti - \Delta t(\alpha ik)}{ti} = 1 - P_i^e \quad ;$$

Pour la période d'observation T, cette probabilité s'exprime par

$$P^d = Pi - \frac{\Delta t\ (\alpha ik)}{T} \quad .$$

Remarque: Notons δ et δ_d l'ensemble des arcs de $\mathcal{A}$ et $\mathcal{A}_d$.
d est une application de $\mathcal{A}$ dans $\mathcal{A}_d$ telle que

$$d: (qi,rj,qj) \longrightarrow (qi,rk,qk).$$

Si $qk \in T$ alors $\Delta t\ (\alpha ik) = 0$; donc $P_i^d = Pi$.
Si $qk = qi$ alors $\Delta t\ (\alpha ik) = ti$; ce qui entraine $P_i^d = 0$.

Ce calcul montre que la probabilité de détection instantanée d'un blocage est impossible, ce qui est évident.
Par contre, l'introduction de la pondération permet de détecter ce blocage car l'arc d'extrémité qf introduit un chemin dans l'automate de révélation.

Remarque

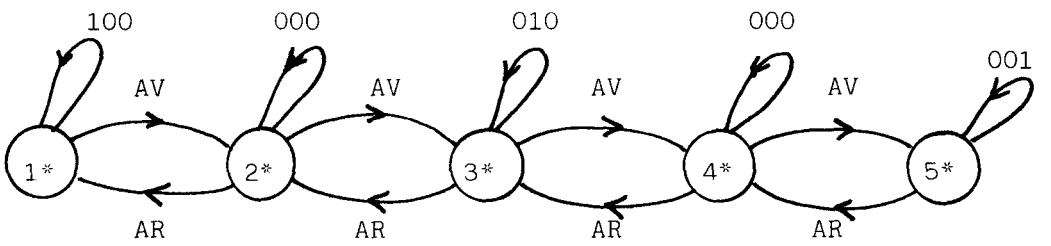
L'étude est faite pour une P.O. localisée. Toutefois, les mêmes calculs peuvent être faits pour une phase de localisation. Le modèle transformé est alors limité aux éléments correspondants. Les probabilités associées à chaque arc du modèle sont alors établies, en fonction des probabilités attribuées aux points singuliers, dans l'hypothèse précédente.

8.2.2 EXEMPLE

Nous examinons la P.O. formée de la came se déplaçant devant trois capteurs C1, C2, C3,utilisée dans les chapitres précédents.
Nous étudions le comportement de chacun des tests face à un collage à 1 puis à 0 de C1. Les commandes sont:

- V1 qui provoque une évolution positive sur la trajectoire ;
- V2 qui commande une évolution en sens inverse;
- V3 qui correspond à l'arrêt de la came.

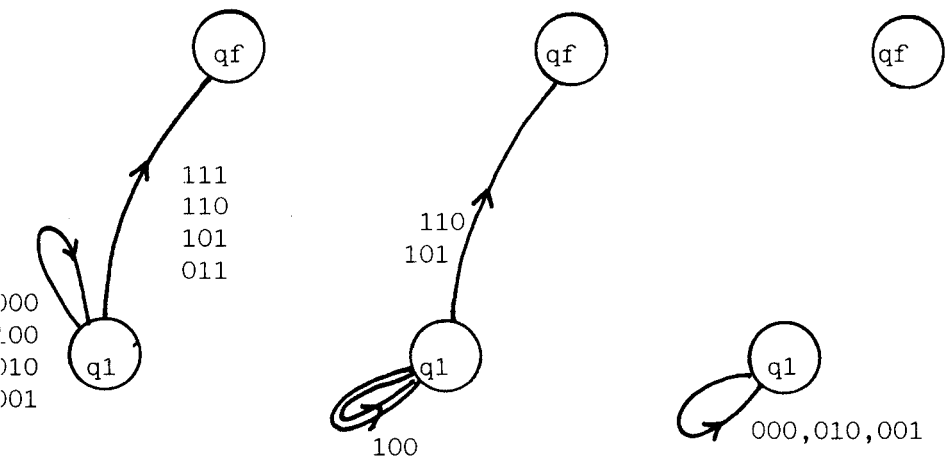
La figure 8.1 représente la trajectoire des P.S. $\mathcal{T}$ de cet actionneur. L'élément de compte rendu qui caractérise chaque transition est porté entre parenthèses. Sur chaque arc, nous indiquons le sens d'évolution réel sur la trajectoire. Nous notons "AV" l'évolution dans le sens positif arbitraire, "AR" le déplacement inverse.
Les probabilités réelles d'être dans qi sont notées Pi^* .



- figure 8.1 -

2.2.1 Test statique

La figure 8.2 traite du test statique. Par souci d'unification de la représentation, nous avons adopté comme modèle un graphe à deux états (sain et défaillant), bien que cela ne s'impose pas. Quelque soit le point singulier réellement occupé, le modèle est dans l'état 1 s'il n'y a pas d'erreur détectée. Donc, $P1^{*} = 1$.



- a - - b - - c -

- figure 8.2 -

La figure 8.2a représente l'automate utilisé pour le test statique. Les figures 8.2b et 8.2c correspondent respectivement à l'automate défaillant pour un collage à "1" d'une part, à "0" d'autre part du capteur C1.

L'examen de la figure 8.2c montre que le collage à 0 n'est pas détectable. D'après la figure 8.2b, il y a deux séquences de révélation représentées ici par r1(1,1) = 110, r2(1,1) = 101 dont la longueur unité permet d'espérer une détection immédiate. Toutefois, la présence de l'arc (q1, 100, q1) laisse prévoir la possibilité d'obtenir une erreur non détectée.
Dans le cadre du test statique, l'erreur est immédiatement détectée si elle apparaît à partir des PS 3 et PS 5 pour lesquels le C.R. normal est respectivement 010 et 001. Les règles de calcul établies ne s'appliquent pas ici directement, car le modèle est un fusionnement d'un modèle dynamique. A partir des remarques ci-dessus, nous pouvons toutefois calculer:

$$Pcd(C1 = 1) = P^*3 + P^*5; \quad Pcd(C1 = 0) = 0$$

$$Pce(C1 = 1) = Pcd(C1 = 1); \quad Pce(C1 = 0) = 0$$

En cas de collage à 1, toute évolution conduisant vers PS 3 ou PS 5 génére une séquence de révélation. Le taux de détection Pd est de 1 pour le collage à 1 et de 0 pour le collage à 0.
Il est déjà possible de conclure que le collage à 1 de C1 est toujours détectable quelque soit le modèle utilisé, si la séquence de travail utilise toutes les commandes.
Il n'est fait aucune hypothèse sur le temps de latence qui peut être long.

8.2.2.2 Test dynamique

Le modèle utilisé est défini au chapitre 5 (figure 5.9).

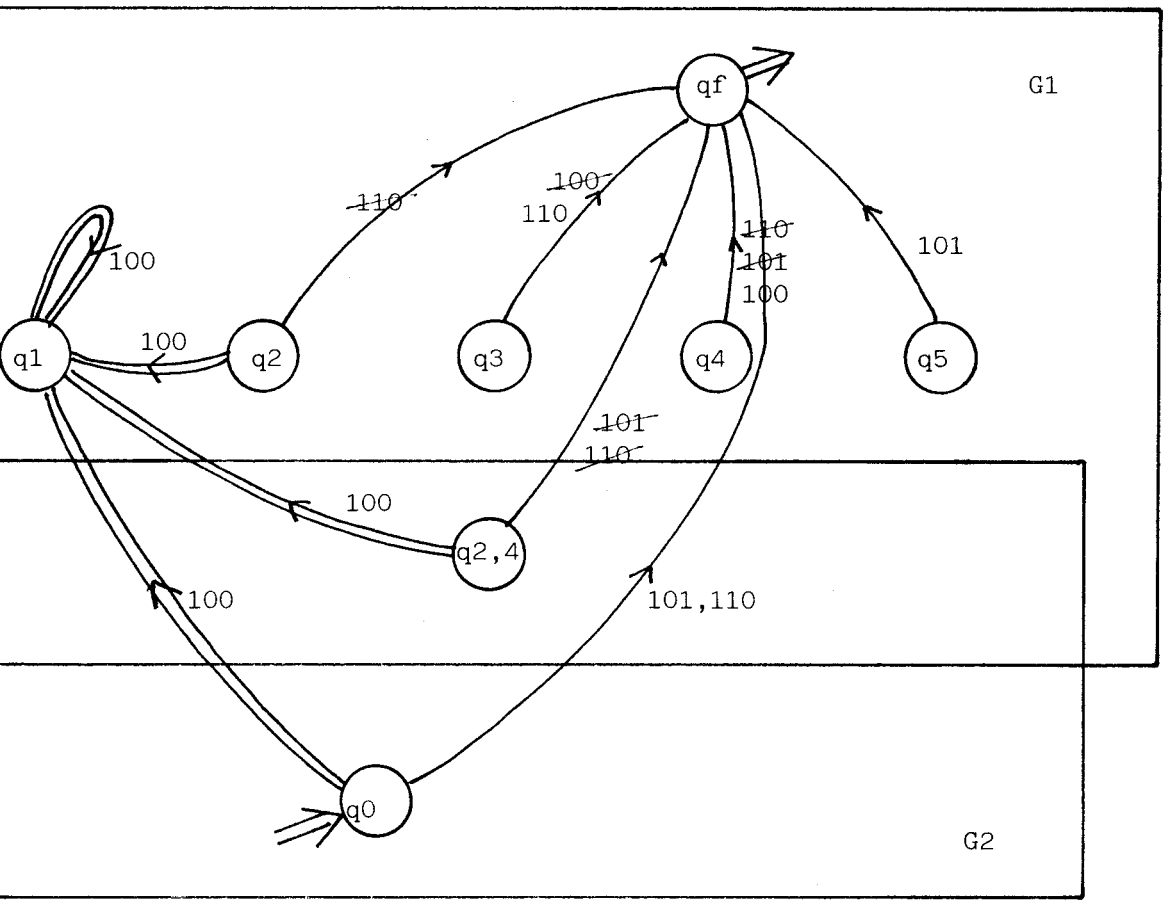
Nous évaluons $P1 = P^*1$; $P3 = P^*3$; $P5 = P^*5$

$$P2 = \frac{1}{2}P^*2; \quad P4 = \frac{1}{2}P^*4$$

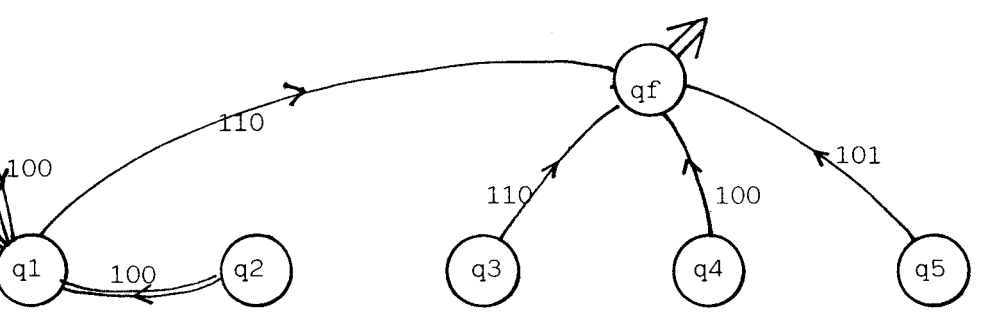
$$P2,4 = \frac{1}{2}(P^*2 + P^*4)$$

La probabilité d'être en P2 ou P2,4 lorsque la P.O. se trouve en PS2 dépend du P.S. précédemment occupé.

Les figures 8.3a et 8.3b donnent les deux graphes obtenus en incluant le collage à 1 dans le modèle.
D'après 8.3 , tout chemin qui, dans le modèle, passe par les états q1; q2 ou q2,4 entraîne une erreur. Compte tenu des arcs qui ne satisfont pas aux conditions d'adjacence (arcs rayés sur la figure 8.3a), nous évaluons les taux de couverture à:



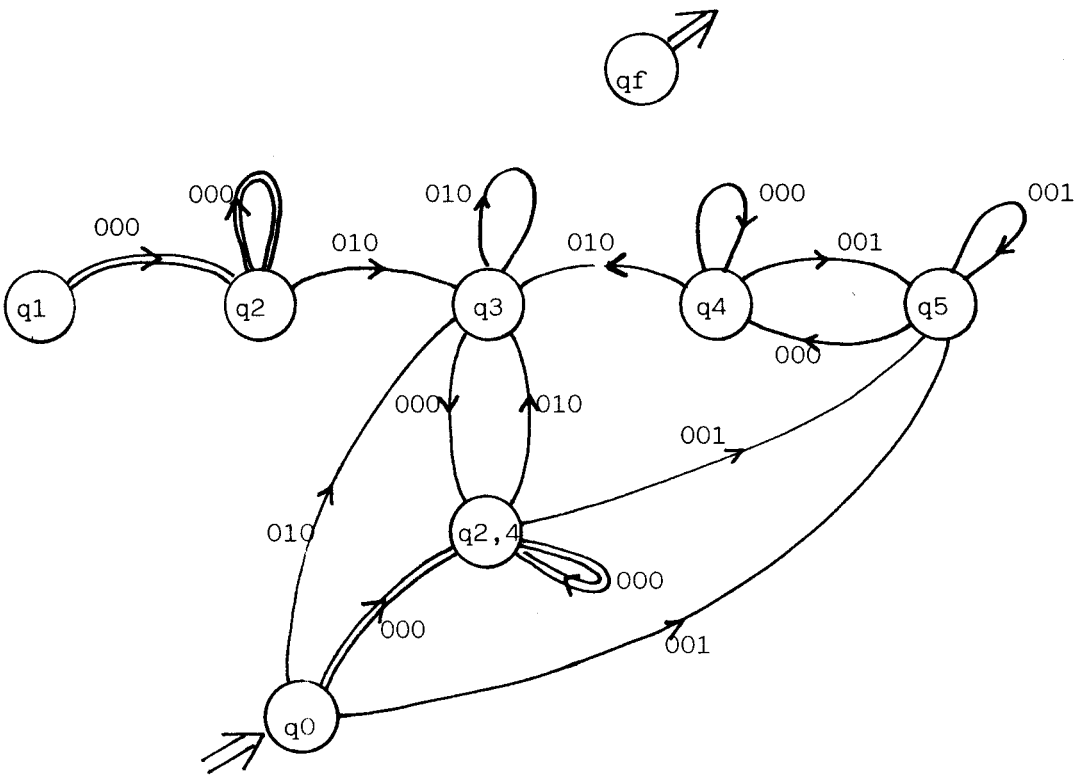
- a - Automate de défaillance



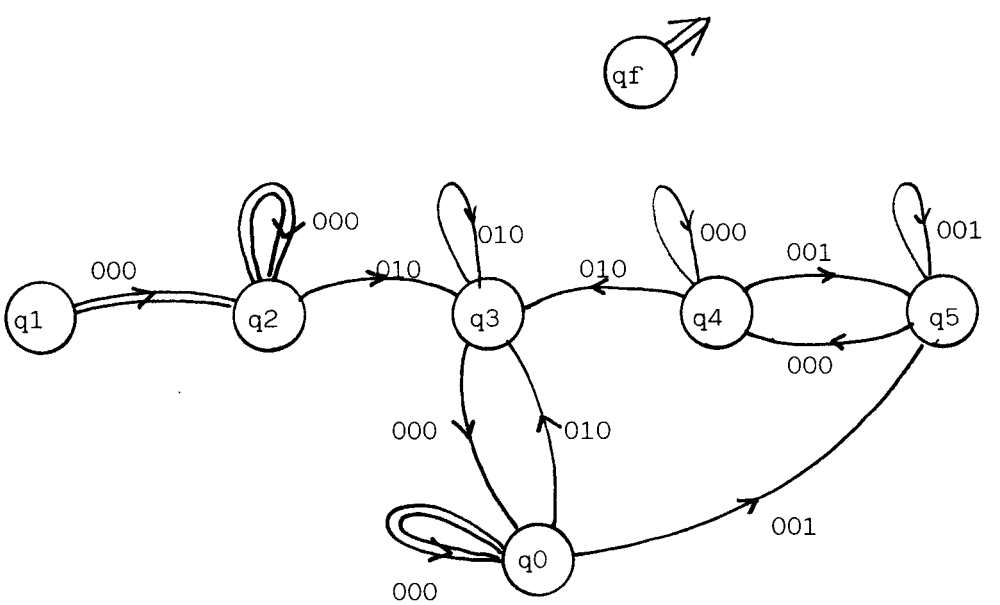
- b - Automate de révélation hors phase de localisation

Collage à 1 de C1

- figure 8.3 -



- a - Automate de défaillance



- b - Automate de révélation

Collage à 0 de C1

- figure 8.4 -

$$Pcd(C1 = 1) = P3 + P4 + P5$$

$$= P*3 + P*5 + \frac{1}{2} P*4$$

$$Pce(C1 = 1) = 1 - \frac{P*2 + \frac{1}{2} P*4}{P*2 + P*3 + P*4 + P*5}$$

L'amélioration par rapport au test statique, provient de la distinction entre les états relatifs à PS 2 et PS 4. Toutefois, ces états restent indiscernables lorsqu'ils sont atteints à partir de PS 3, comme l'atteste l'état q2,4.

Les figures 8.4a et 8.4b relatives au collage à 0 de C1 montrent que ce défaut reste non révélé par la méthode de test simplement dynamique.

En phase de localisation, on trouve:

$Pcd(C1 = 1) = P*3 + P*5$; $Pcd(C1 = 0) = Pce(C1 = 0) = 0$.

$Pce(C1 = 1) = P*3 + P*5$.

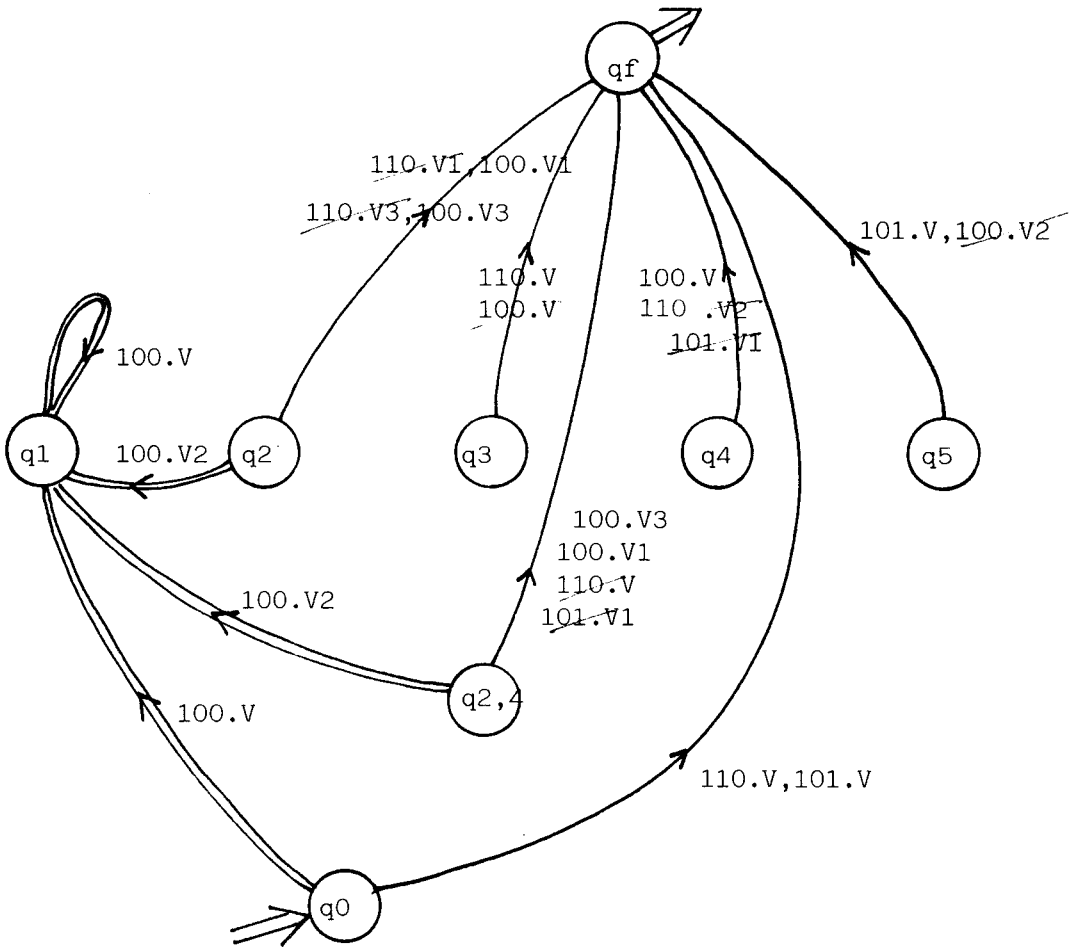
2.2.3 Test dynamique avec commande

Nous reprenons l'automate défini au chapitre VI (figure 6.3), pour lequel les sommets ambigus sont fusionnés avec les sommets non ambigus respectifs ($q1,f \longrightarrow q1$; $q3,f \longrightarrow q3$; $q5,f \longrightarrow q5$).
Nous évaluons $Pi = P*i$ $\quad i = 1, \ldots, 5$.

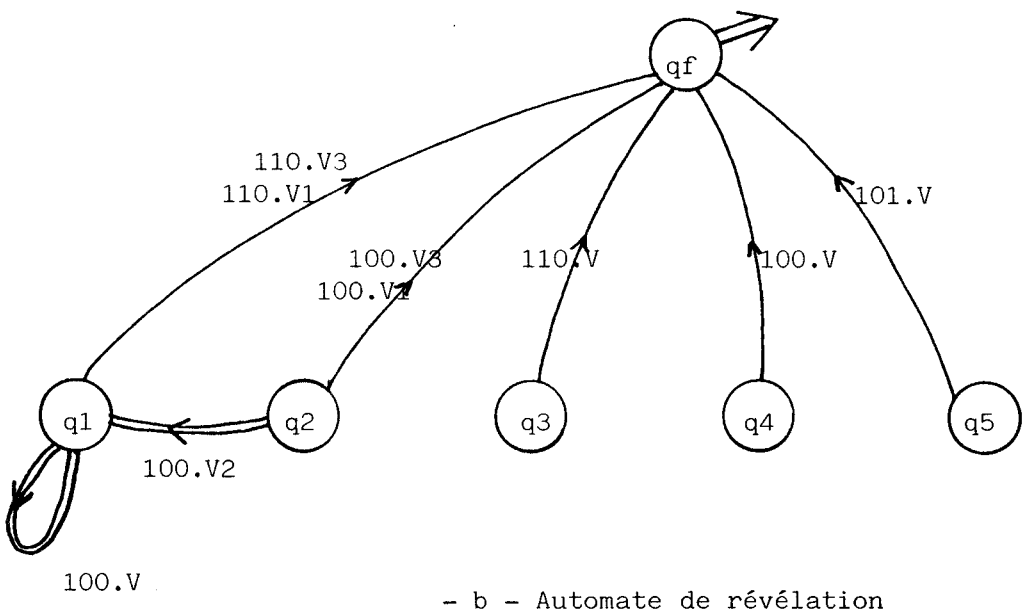
La figure 8.5 représente les automates défaillants relatifs au collage à "1" de C1. Les possibilités de détection du défaut sont maintenant liées à la commande appliquée. Ceci est très net lorsque le modèle est dans l'état q2 au moment de la défaillance.
Les taux de couverture doivent être définis séparément pour chaque commande.

Toute modification de C.R. sous V3 est anormale.

La comparaison avec le test dynamique simple (figure 8.4) montre l'existence d'arcs 100 V1 qui entraînent la transition vers qf. De plus, l'état q2,4 n'est pas à prendre en considération en dehors de la phase de localisation.



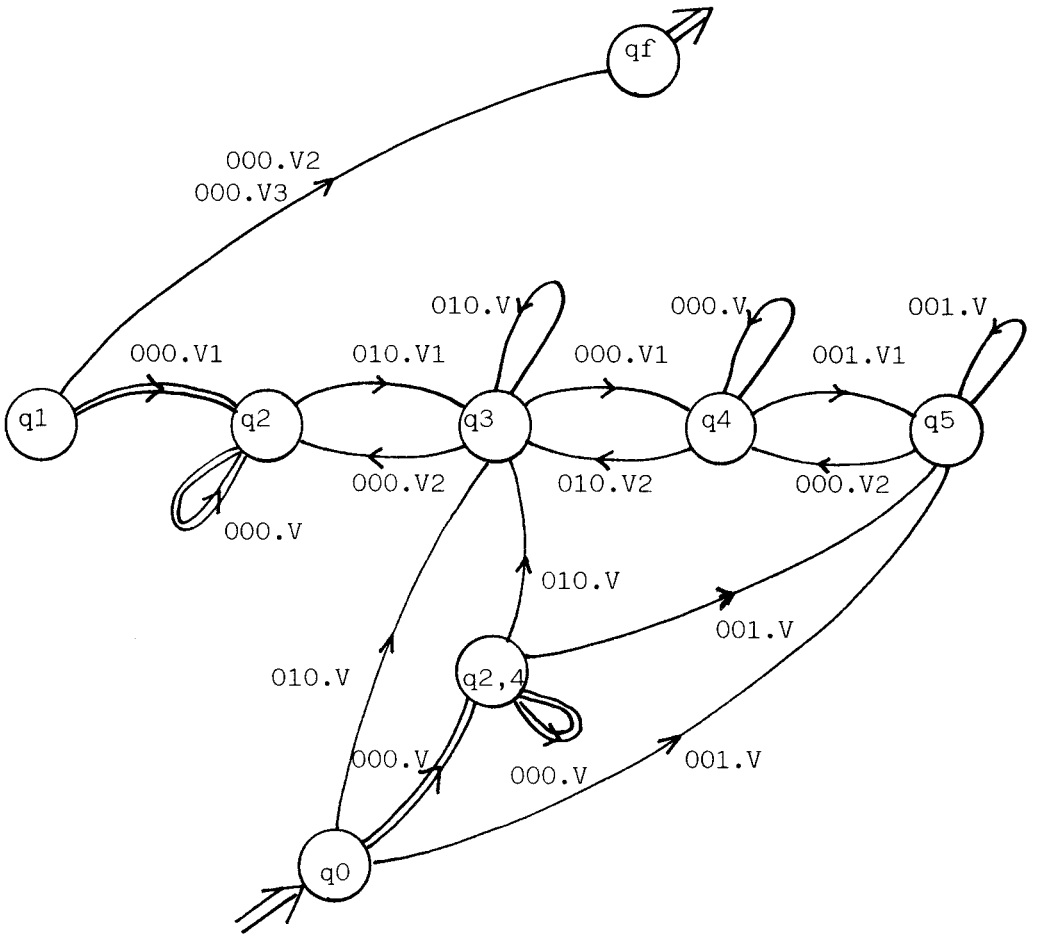
- a - Automate de défaillance



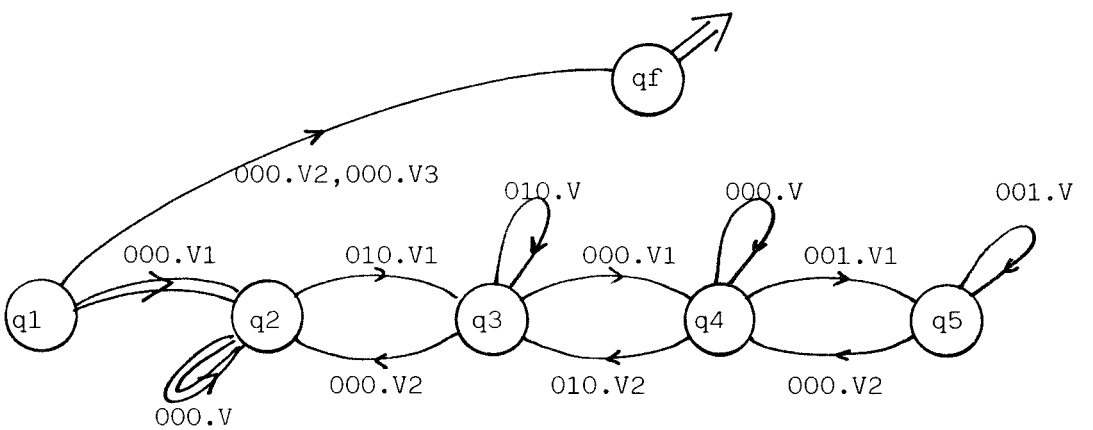
- b - Automate de révélation

Collage à 1 de C1

- figure 8.5 -



- a - Automate de défaillance



- b -Automate de révélation

Collage à 0 de C1

- figure 8.6 -

Les taux de couverture sont:

Pour la commande V1 $Pcd = 1 - P*1$

$Pce = 1$

Pour la commande V2 $Pcd = P*3 + P*4 + P*5$

$= 1 - P*1 - P*2$

$$Pce = \frac{1 - P*1 - P*2}{1 - P*1} = 1 - \frac{P*2}{1 - P*1}$$

Pour la commande V3 $Pcd = 1 - P*1$

$$Pce = \frac{1 - P*1}{1 - P*1} = 1$$

D'après la figure 8.5b, nous trouvons Pd = 1.

Pour le collage à 0 (figure 8.6), nous obtenons, comme taux de couverture de l'erreur, après localisation:

pour la commande V1 $Pcd(C1=0) = Pce(C1=0) = 0$

pour la commande V2 $Pcd(C1=0) = P*1$; $Pce(C1=0) = 1$

pour la commande V3 $Pcd(C1=0) = P*1$; $Pce(C1=0) = 1$

Pendant la phase de localisation, ces mêmes taux sont nuls.

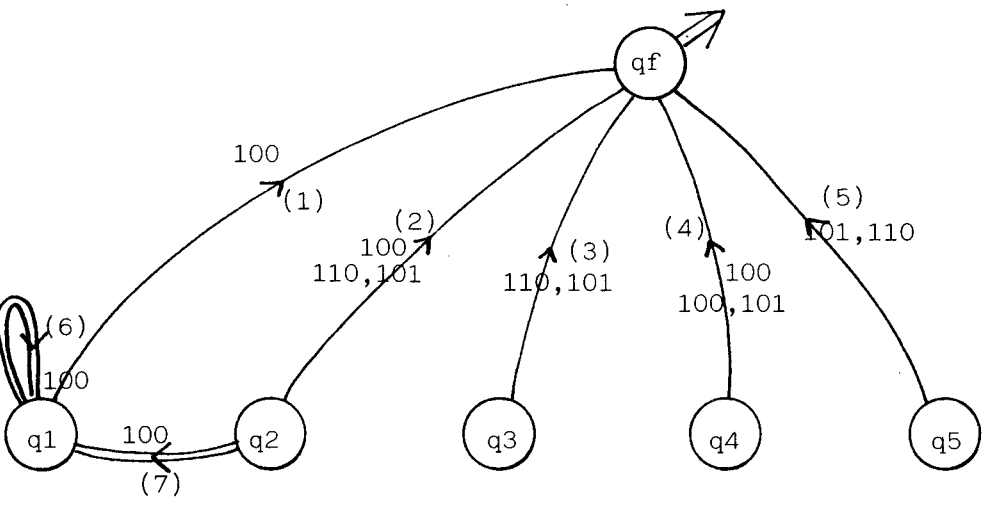
Cet exemple montre assez l'insuffisance de ces chiffres pour spécifier complètement les performances du test. En effet, sur la figure 8.6, nous constatons que le seul chemin, capable de générer une séquence de révélation, a pour origine q1. En d'autres termes, le collage à "0" ne peut être révélé que si l'état du modèle, à cet instant, est représenté par q1. Dans tous les autres cas, cette panne est masquée. Ce phénomène est bien mis en évidence sur la figure 8.6b correspondant à l'automate de révélation.
Nous en déduisons:

$$Pd = P*1, V2 + P*1, V3$$

où P*1, V2 et P*1, V3 sont les probabilités d'être au PS 1 sous la commande V2 (resp. V3) au moment de l'apparition de la défaillance.

8.2.2.4 Test dynamique par automate pondéré

Le modèle défaillant est tiré de la figure 7.5. En dehors de la phase de localisation, les automates relatifs au collage à "1" et à "0" de C1 sont donnés respectivement par les figures 8.7 et 8.8. Les valeurs des taux de couverture par arc sont regroupés dans les tables 8.1 et 8.2.

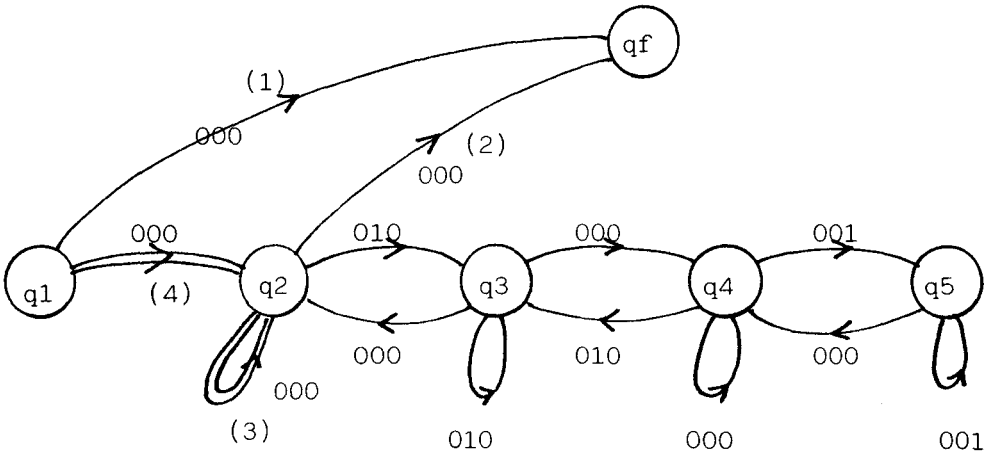


- figure 8.8 -

n° arc	C.R.	Taux de couverture			extrémité
		V1	V2	V3	
1	100	0	X	X	qf
2	100	P2*	$P2^*(1-\frac{\Delta t}{t2})$	P2*	
3	110	P3*	P3*	P3*	
4	100	P4*	P4*	P4*	
5	101	P5*	P5*	P5*	
6	100	P1*	X	X	≠ qf
7	100	X	$\frac{\Delta t}{t2}$ P2*	X	

X : probabilité inchangée par la défaillance.

- tableau 8.1 -



- figure 8.8 -

n° arc	C.R.	Taux de couverture			remarque
		V1	V2	V3	
1	000	P1* - $\frac{\Delta t}{T}$	P1*	P1*	erreur
2	000	X	0	X	détectée
3	000	X	0	X	erroné
4	000	$\frac{\Delta t}{T}$	0	0	

- tableau 8.2 -

L'évaluation donne pour le collage à 1

Pour la commande V1 $Pcd = 1 - P^*1$

$Pce = 1$

Pour la commande V2 $Pcd = 1 - P^*1 - \frac{\Delta t}{T}$

$Pce = \frac{Pcd}{1-P^*1} = 1 - \frac{\Delta t}{T} \cdot \frac{1}{1-P^*1}$

Pour la commande V3 $Pcd = 1 - P^*1$

$Pce = 1$

La même évaluation pour le collage à 0 donne:

Pour la commande V1 $Pcd = P^*1 - \frac{\Delta t}{T}$

$Pce = \frac{Pcd}{P^*1} = 1 - \frac{\Delta t}{T} \cdot \frac{1}{P^*1} = 1 - \frac{\Delta t}{t1}$

Pour les commandes V2 et V3 $Pcd = P^*1$

$Pce = 1.$

Mais le résultat le plus spectaculaire correspond au taux de révélation Pd = 1. Ceci correspond au fait qu'il existe toujours un chemin qui mène à q2 donc à qf, par (q2, 000, qf). La détection est alors liée à l'existence de la temporisation qui borne la durée de blocage en q2 sous la commande V1.

Les taux de couverture relatifs à notre exemple sont regroupés dans les tableaux 8.3 pour le collage à 1, et 8.4 pour le collage à 0 de C1. Le tableau 8.5 donne à titre indicatif, les valeurs numériques des estimateurs pour les différents types de tests étudiés.

Ces valeurs ont été calculées en fonction d'une présence équiprobable en chaque P.S. et pour des valeurs de $\frac{\Delta t}{ti} = 0{,}1$. Dans la pratique, ces probabilités peuvent être évaluées sur des critères objectifs, voire même déterminés statistiquement si l'automate existe. Ces probabilités sont différenciées en fonction de la commande si cette grandeur est prise en compte dans le modèle.

L'examen de la table 8.4 montre une amélioration certaine de performances pour le test avec automate pondéré. Ceci est surtout visible en cas de collage à 1.

Certains chiffres peuvent surprendre, ils doivent être vus en fonction de ce qu'ils sont censés mettre en évidence.

Type test	commande	Pcd	Pce	Pd	Observations
Statique		P3*+P5*	P3*+P5*	1	seul le passage par PS3 ou PS5 permet de révéler l'erreur.
Dynamique		P3*+P5*+1/2 P4*	$1- \frac{P1*+ 1/2\ P4*}{P2*+P3*+P4*+P5*}$	1	révéler dés apparition de l'erreur à partir de PS3, PS4, PS5.
Dynamique+ commande	V1	1-P1*	1	1	
	V2	1-P1*-P2*	$1- \frac{P2*}{1-P1*}$	1	
	V3	1-P1*	1	1	
Dynamique pondéré	V1	1-P1*	1	1	
	V2	$1-P1*- \frac{\Delta t}{T}$	$1- (\frac{\Delta t}{T})/(1-P1*)$	1	
	V3	1-P1*	1	1	

$P1* + P2* + P3* + P4* + P5* = 1$

Collage à 1 de C1

- tableau 8.3 -

Type test	commande	Pcd	Pce	Pd	Observations
Statique		0	0	0	Masquée
Dynamique		0	0	0	Masquée
Dynamique + commande	V1	0	0	0	Masquée
	V2	P1*	1	P1*	
	V3	P1*	1	P1*	
Dynamique pondéré	V1	$P1^* - \frac{\Delta t}{T}$	$1 - \frac{\Delta t}{t1} = 1-(\frac{\Delta t}{T})/P1^*$	1	
	V2	P1*	1	1	
	V3	P1*	1	1	

Collage à 0 de C1

- tableau 8.4 -

Type de test	commande	Collage à 1 de C1			Collage à 0 de C1		
		Pcd	Pce	Pd	Pcd	Pce	Pd
Statique		0.4	0.4	1	0	0	0
Dynamique		0.5	0.5	1	0	0	0
Dynamique + commande	V1	0.8	1	1	0	0	0
	V2	0.6	0.75	1	0.2	1	0.2
	V3	0.8	1	1	0.2	1	0.2
Dynamique pondére	V1	0.8	1	1	0.18	0.9	1
	V2	0.78	0.98	1	0.2	1	1
	V3	0.8	1	1	0.2	1	1

Valeurs calculées en supposant que la présence en chaque point singulier est équiprobable.

Pour $\frac{t}{ti}$ nous avons pris 0.1 ce qui donne, avec l'hypothèse ci-dessus, $\frac{t}{T} = 0.02$.

Pour le collage à 0 de C1 par exemple, la relation $P_{cd} \ll P_{ce}$ traduit simplement le fait que la défaillance est tolérée sur la majeure partie de la trajectoire.
Il peut paraître plus surprenant encore d'avoir un taux de couverture de l'erreur de 1 et un taux de détection de 0,2 (modèle dynamique + commande). En fait, l'erreur n'est agissante qu'à partir de q1,1.
La détection est alors immédiate pour V2 et V3.
Parcontre, si la vitesse est V1,ou si le système est en un état autre que q1, l'erreur n'est plus détectable, mais il n'y aura pas non plus d'erreur agissante. Par contre, il y a blocage en q2.

3 DEFAILLANCES AFFECTANT LA VITESSE D'EVOLUTION

Nous avons vu au paragraphe 1 de ce chapitre que certaines défaillances affectent la vitesse d'évolution sur la trajectoire. Cette anomalie peut porter sur le module et/ou sur le sens de la vitesse.
Par principe, seuls les modèles qui intègrent une information de commande sont susceptibles de détecter l'anomalie.

Les commandes peuvent être classées en trois groupes qui sont:

- celles qui imposent une vitesse > 0;
- celles qui donnent une vitesse < 0;
- celles qui arrêtent l'évolution.

Dans tous les cas, l'instant d'apparition de l'erreur coïncide avec la substitution de vitesse.

3.3.1 TEST DYNAMIQUE AVEC COMMANDE

Nous supposons qu'il existe une défaillance telle que:

$$d : \quad v1 \longrightarrow v2^{*}$$

$v2^{*}$ représente la vitesse réelle, différente de celle attendue.
$v2^{*}$ peut toujours être rangée dans l'une des trois classes envisagées même si elle ne correspond à aucune commande normale.
Appelons $\mathcal{C}_{v1}$ et $\mathcal{C}_{v2}$ la classe d'appartenance de ces deux vitesses.

Comme précédemment, nous introduisons la défaillance dans le modèle.

S'il existe, en fonctionnement non défaillant, une commande v3 telle que $\varphi_{v3} = \varphi_{v2}$, alors tout arc (qi, rj*v1, qj) est remplacé par l'arc (qi, rj*v3, qk) auquel est associée l'étiquette rj*v1.
L'arc obtenu est alors (qi, rj*v1, qk).
Si qk $\notin$ T, alors cet arc est repéré comme erronné (double trait sur les figures).

Remarque

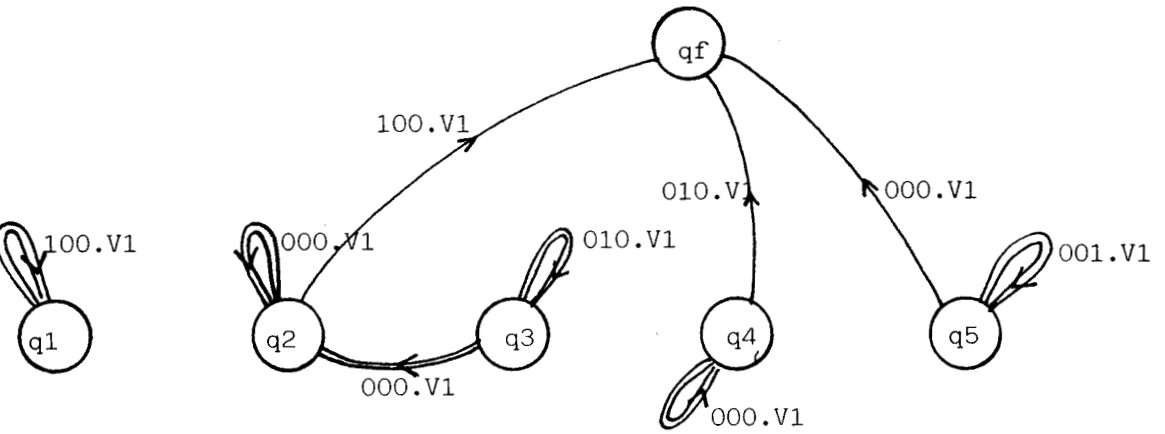
Si la commande v3, telle que $\varphi_{v3} = \varphi_{v2}$ n'existe pas, il n'est pas possible de faire cette transformation. Toutefois, nous rappelons que notre modification de graphe est un artifice permettant d'évaluer les performances. Elle ne correspond à aucune phase de mise en oeuvre du test. Il est toujours possible d'imaginer l'existence de v3 dans ce cas.

Les figures 8.9a, 8.9b, 8.9c représentent les automates défaillants relatifs à l'exemple choisi pour les différents types de substitutions interclasses, en dehors de la phase de localisation. Toute substitution de deux commandes de même classe est, par principe, ignorée avec un automate non pondéré.

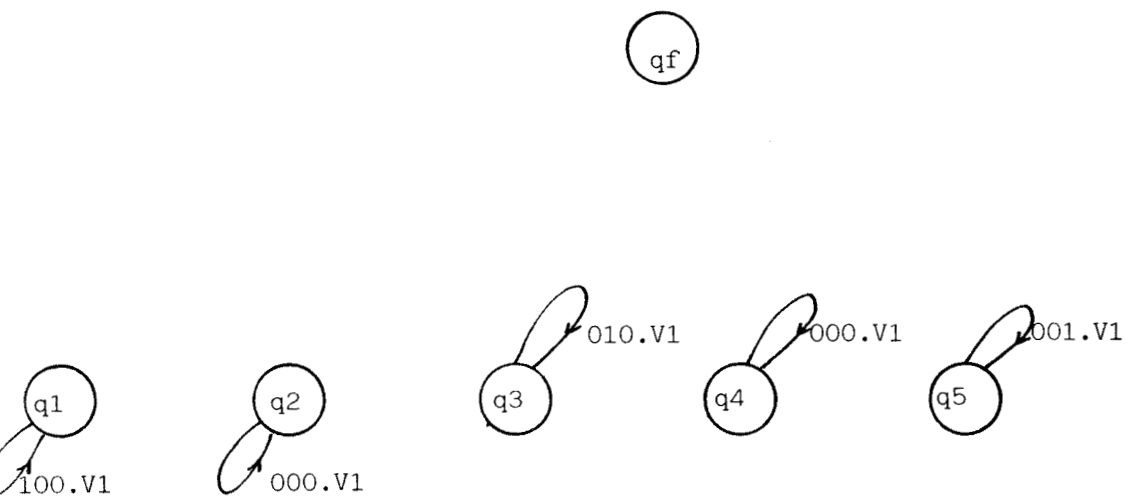
Comme nous l'avons déjà signalé, la probabilité de détecter la substitution de commande, donc l'erreur, au moment où elle se produit, est nulle. Par contre, lorsque cette anomalie provoque une substitution au niveau du C.R., il est important de détecter cette modification. Ceci permet d'éviter la propagation de l'erreur par la P.C..
En fait, il faut supposer que ce type d'anomalie a une probabilité de réalisation suffisamment faible pour être acceptée, et/ou qu'elle n'est pas catastrophique.
Nous appelerons donc taux de couverture de l'erreur Pce, la probabilité de détecter l'erreur de C.R., si une erreur de commande existe.

$$Pce = \mathcal{P}(\text{détecter la substitution de C.R. due à une substitution de commande}).$$

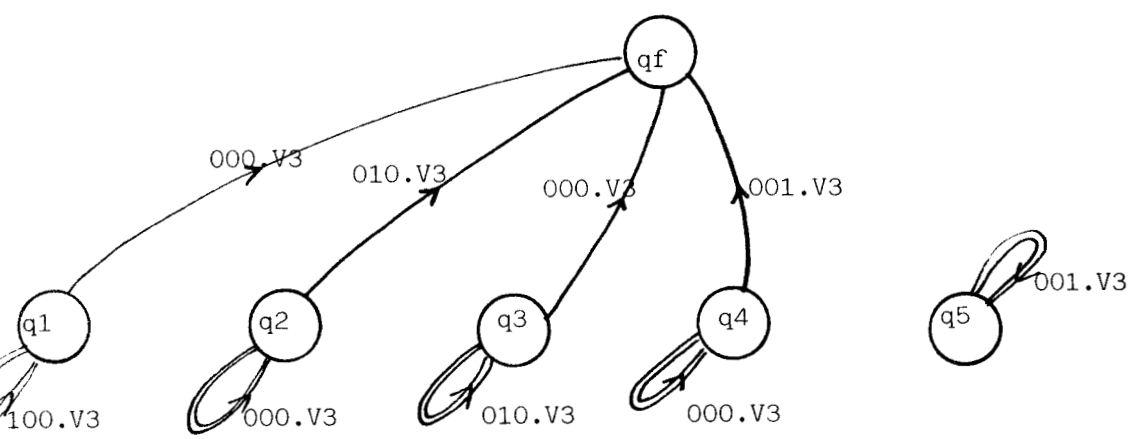
Pce est égal à la somme des probabilités P_i associées aux sommets qi origines d'un arc menant à qf $\in$ T.



- a - Substitution v1 —— v2



- b - Substitution v1 —— v3



- c - Substitution v3 —— v1

- figure 8.9 -

De même, nous évaluons la probabilité Pd de détecter l'erreur de C.R. liée à la substitution de commande par:

$$Pd = \mathcal{P}(\text{détecter erreur de C.R. liée à une substitution de commmande}).$$

Ce coefficient est évalué en faisant la somme des probabilités associées aux sommets origines d'un chemin menant à qf dans l'automate défaillant.

Les résultats correspondant à l'exemple traité sont regroupés dans le tableau 8.5.

	Pce	Pd	
v1 ⟶ v2	$1 - P_1^* - P_3^*$	$1 - P_1^*$	erreur de sens
v1 ⟶ v3	0	0	blocage
V3 ⟶ v1	$1 - P_5^*$	$1 - P_5^*$	départ intempestif
v1 ⟶ v'1	0	0	changement de module sans changement de classe

Tableau 8.5

Remarque

La transition 010 ⟶ 000, à partir de q3, ne permet pas de détecter l'erreur puisque le C.R. (ici 000) obtenu en quittant PS 3, est identique quelque soit le sens réel d'évolution.

Ceci met en évidence l'influence des chemins qui génèrent un "palindrome", c'est-à-dire ceux qui, pris dans un sens ou dans l'autre, génèrent la même séquence.

Appelons sommet pivot, l'état qui représente le point de symétrie. Si la substitution se produit alors que l'état de la P.O. est représenté par le sommet pivot, le temps de latence correspond alors au temps nécessaire pour sortir du palindrome si la commande est maintenue. Ce phénomène justifie le terme -P*3 dans l'expression de Pce dans la substitution v1 ⟶ v2 (tableau 8.5).

.3.2 TEST DYNAMIQUE PONDERE

La transformation précédente est reconduite. Mais pour évaluer les taux de détection, nous devons étudier spécifiquement l'influence de la pondération sur ces valeurs.

Soit:

- αij = (qi, rj*v1, qj) et αik = (qi, rj*v2, qk)
les arcs du modèle pondéré normal.

- α'ik = (qi, rj*v1, qk) l'arc du modèle défaillant pour lequel I (α'ik) = I (αik).

I (αik) évalue le temps que va mettre le système pour atteindre qk à la suite de la défaillance.

I (αij) permet de connaître les configurations pour lesquelles l'arc αij existe. Pour les autres configurations, αij est remplacé par (qi, rj*v1, qf).

Pour toute valeur du poids pi affecté à qi telle que pi $\in$ I(αik) $\cap$ I(αij), le modèle tolère la transition anormale vers qk, qu'il confond avec la transition normale vers qj.

Notons Δ(I(αik) $\cap$ I(αij)) la dimension de cette intersection.

Nous avons pi $\in$ |0, ti| avec par principe I(αik), I(αij) $\in$ |0, ti|.

Dans l'hypothèse où la défaillance peut apparaître de façon aléatoire dans l'intervalle |0, ti|, nous pouvons évaluer la probabilité d'une transition erronée à

$$Pei = \frac{\Delta(I(\alpha ik) \cap I(\alpha ij))}{ti}$$

Dans la pratique, la connaissance des intervalles permet de chiffrer avec précision cette incertitude. Dans tous les cas, il est possible d'encadrer cette valeur.
Dans le cas le plus défavorable, l'un des intervalles est inclus dans le plus grand. Dans ce cas:

$$Pei = \frac{Min(\Delta tik\ ,\ \Delta tij)}{ti} = \frac{\Delta ti}{ti}$$

où Δtik représente la dimension de l'intervalle I(αik) correspondant à Δtik = Max I(αik) - Min I(αik). Même chose pour Δtij.

Dans le cas le plus favorable, les intervalles sont disjoints.

Nous en concluons que:

$$0 \leqslant Pei \leqslant \frac{\Delta ti}{ti}$$

Le taux de couverture de l'erreur à partir de qi est donc compris dans

$$Pi * (1 - \frac{\Delta ti}{ti}) \ll Pci \leqslant Pi$$

Remarque

Lorsque la substitution de commande correspond à un changement de signe de la vitesse, nous avons:

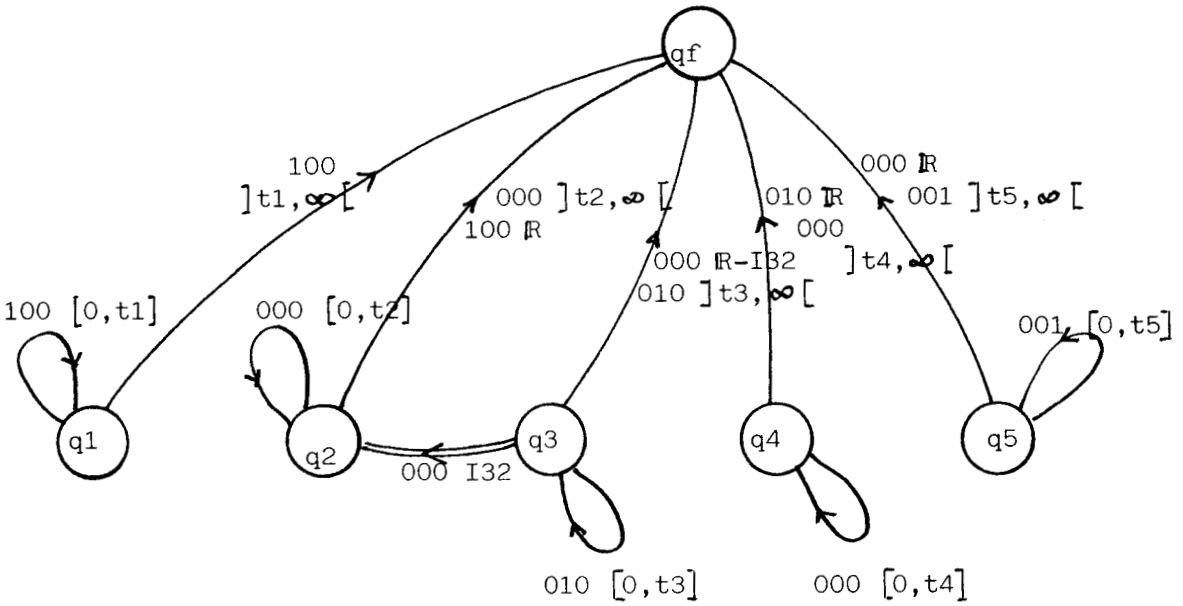
- l'intervalle correspondant à la traversée du point singulier pour un arc;
- celui relatif à une sortie du P.S. par le même côté pour l'autre.

Ces deux intervalles sont très souvent disjoints ou de faible recouvrement. La probabilité de ne pas détecter l'erreur dès la modification de C.R. est alors très faible.

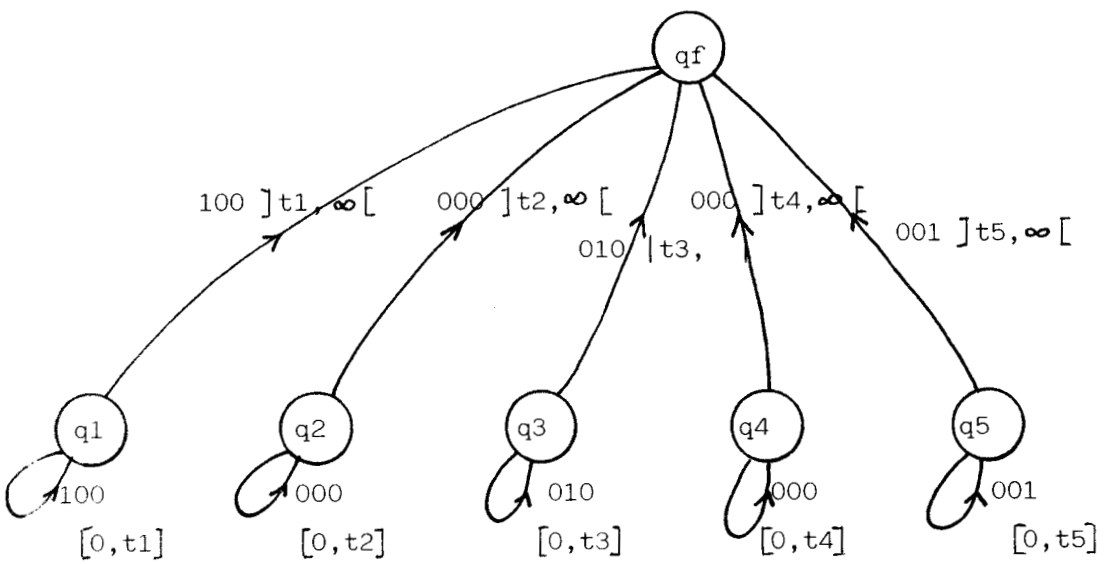
Nous donnons sur la figure 8.10 les automates défaillants de référence relatifs aux trois substitutions types. Le tableau 8.6 regroupe les évaluations des taux de couverture.

Substitution	Pce	Pd	
v1 ⟶ v2	$1-P*3 \leqslant 1-\frac{\Delta t3}{T} \leqslant Pce \leqslant 1$	1	si changement de sens $\frac{\Delta t3}{T}$ en général petit
v1 ⟶ v'1	idem	1	$\frac{\Delta t3}{T}$ peut être plus grand que ci-dessus
v1 ⟶ v3	1	1	blocage Le temps de latence est ici borné par Max(ti)
v3 ⟶ v1	1 - P*5	1 - P*5	départ intempestif

Tableau 8.6



- a - Substitutuion $\mathcal{C}_{v1} \longrightarrow \mathcal{C}_{v2}$



- b - Substitution $\mathcal{C}_{v1} \longrightarrow \mathcal{C}_{v3}$

Pour la substitution $\mathcal{C}_{v3} \longrightarrow \mathcal{C}_{v1}$, le graphe est identique à celui de la figure 8.9c.

- figure 8.10 -

L'amélioration apportée par la pondération est sensible dans la majorité des cas. Elle est surtout due à l'aptitude du modèle à détecter les blocages.
S'il reste impossible de détecter immédiatement l'erreur en cas de substitution de commande, la détection de l'anomalie dès la première modification de C.R. est très probable dans la majorité des cas.
L'influence du palindrome y est certainement atténuée.

S'il n'y a pas d'amélioration pour le départ intempestif, c'est que:

- d'une part, la pondération ne joue pas pour la commande $v3$ le temps d'arrêt est uniquement fixé par la durée d'application de v3,
- d'autre part, un départ intempestif vers la droite n'a plus d'effet sur le C.R. si la P.O. est déjà à l'extrémité droite de sa trajectoire.

Remarque

Le cas du test dynamique dans le contexte du cycle de travail répétitif n'a pas été étudié spécifiquement.
Dans la mesure où il est fait appel à un automate pondéré, les résultats obtenus sont comparables à ceux correspondant au test dynamique temporisé.

La procédure de test met en évidence les erreurs. Il est aussi important de localiser l'origine de l'anomalie. Cette localisation permet de réduire les temps de maintenance. Elle est surtout utile pour permettre une réaction sélective en fonction du défaut présumé, des risques pris... Elle permet, par exemple, de prendre des décisions du type "produire tout de même" ou de mettre en oeuvre des sécurités parfois pénalisantes voire destructives (systèmes parachutes). Elle est souvent utile à la reconfiguration du système.

8.4 LOCALISATION DE LA DEFAILLANCE

En fait, le système de test détecte une erreur de C.R., le problème est de déduire de cette observation, l'origine de la défaillance.

Une première constatation s'impose, pour la localisation de la défaillance, il faut au moins avoir détecté l'erreur.

Le fait d'avoir un sommet terminal par grandeur mesurée (ou sous P.O.) permet de cerner la défaillance aux matériels correspondants.

3.4.1 CORRESPONDANCES ERREURS - DEFAILLANCES

Les différentes erreurs constatées peuvent être rangées de la façon suivante:

- C.R. technologiquement impossible,
- C.R. impossible à partir de l'état considéré,
- C.R. incompatible avec la commande,
- C.R. hors des limites de temps retenues.

Cette classification est ordonnée. Nous considérons que la proposition à prendre en compte est la première de cette liste, qui soit satisfaite. Nous remarquons aussi que cette classification suit les différents niveaux de modèles utilisés.

Nous donnons parmi la liste des matériels concernés, celui qui est le plus probablement à l'origine de l'erreur, par utilisation des remarques ci-dessous:

a) C.R. technologiquement impossible
Cette erreur affecte de façon certaine les éléments qui forment le C.R.. A défaut d'informations supplémentaires, il est impossible de faire mieux. A partir de la connaissance du C.R. précédent, il est possible d'incriminer la dernière variable qui vient de changer de valeur.

b) C.R. impossible à partir du sommet qi occupé par le modèle quelque soit la commande
La défaillance affecte ici aussi un élément relatif à un capteur.
Supposons que l'état du modèle avant passage en qf soit qi.
Notons (qi, rii, qi) la boucle sur qi et r $\in$ R l'élément de compte rendu suspect.
Si r et rii sont adjacents, alors la défaillance porte sûrement sur les éléments qui élaborent cette variable.

c) C.R. incompatible avec la commande
Cela suppose que l'évolution du modèle prend en compte la commande.

Si le point (b) ci-dessus n'a pas été retenu, c'est que le C.R. obtenu est plausible à partir du sommet qi occupé avant le passage en qf. Si ce C.R. est toutefois inaccessible avec la commande appliquée, alors l'erreur est liée très certainement à une substitution de commande. L'élément défaillant porte sur l'ensemble préactionneurs - actionneurs.

d) C.R. hors limites temporelles

Aucune des propositions qui précédent n'ayant pu être retenues, il est impossible de localiser avec plus de précision l'origine de la défaillance.

Le dépassement de la limite haute peut correspondre:

- à un blocage du capteur, ou de tout autre élément, qui fournit la variable attendue,
- à un blocage ou un ralentissement mécanique,
- à une défaillance des préactionneurs (ou des actionneurs).

Une modification trop rapide de C.R. est attribuée à:

- une accélération de l'évolution liée à une diminution de charge,
- une défaillance de préactionneur ou d'actionneur,
- un capteur défaillant tel que le C.R. attendu pour sortir du P.S. est présent dès l'entrée dans le P.S..

Dans tous les cas, il peut aussi s'agir d'une modification de dimension des P.S. qui s'apparente aux erreurs de dérive, de biais, facteurs d'échelle etc...

8.4.2 LES ECHECS DU TEST ET DE LA LOCALISATION

Une première cause d'échec est liée au mauvais fonctionnement du système de test lui-même. Nous ne parlons pas ici des limites inhérentes aux modèles utilisés, ceci a été pris en compte dans le calcul des taux de couverture, mais des anomalies dues:

- à la défaillance matérielle ou logicielle du système de test,
- aux rejets liés au choix des intervalles de tolérances.

Le premier point va de soi. Le deuxième aspect est lié au dilemme entre l'amélioration des taux de couverture et la répétabilité de la P.O.. Prendre des Δt très grands, c'est laisser passer des pannes.

Si Δt est petit, la présence de perturbations importantes mais acceptables, ou l'influence de transitoires non éteints, entraîne une détection erronée et un diagnostic dénué de toute réalité.

Le choix de la dimension de l'intervalle associé à un arc dépend de la distribution des durées d'attente de cette transition et à la confiance accordée.
Nous ne développons pas cet aspect très classique.
Nous rappelons ci-dessous le résultat obtenu dans l'hypothèse d'une distribution des temps de transition selon la loi normale.

Notons $I = \overline{t} \pm \alpha \sigma$ l'intervalle associé à un arc
avec $\overline{t}$ temps moyen
σ écart type
α coefficient de confiance.
Pour un coefficient de confiance égal à 2, le nombre de fausses pannes prévisibles est de 4,6% des comptes rendus hors délais.

L'échec de la localisation sera réel lorsque le système de test, en présence d'une erreur non détectée, évolue vers un état qui ne correspond plus au P.S. réellement occupé. Ceci est possible lorsqu'il existe une séquence de révélation générée par un chemin passant par un arc erroné. Dans ce cas, la localisation étant faite à partir d'un état qi qui n'est pas le bon, la liste des éléments susceptibles d'être défaillants ne sera pas cohérente.

Exemple

Si l'on considère le collage à 1 de C1 dans le cadre du test dynamique (figure 8.3a), et si ce collage apparaît alors que l'on se trouve en q2,4 (correspondant au point singulier PS 2), le modèle passe en q1. Si la P.O. évolue vers PS 3, le C.R. 110 arrive. Le modèle détecte un C.R. impossible (fig. 8.3b). Le C.R. précédent étant 100, le diagnostic est:

collage à 1 de C2 (alors que c'est C1 qui est en défaut).

8.1.3 LES APPORTS DU TEST ET DE LA LOCALISATION

L'intérêt direct du système de détection est évident par son impact sur la sécurité. Un tel dispositif est, de plus, indispensable pour permettre à un éventuel système de réaction de prendre des décisions adaptées à la défaillance.

La localisation constitue également une aide précieuse à la maintenance curative. Mais l'apport du système de test à la maintenance préventive est, lui aussi, très important.
En permettant la constitution automatique d'un historique des pannes, il constitue une base de données très utile pour fixer la politique de maintenance.
En suivant l'évolution ou la dérive des durées (ou poids) associées à chaque P.S. en exploitation, il permet de mettre en évidence certains phénomènes de vieillissement affectant notamment la mécanique.

CONCLUSION

Dans ce chapitre, nous avons défini des paramètres estimateurs de performances et les moyens de les calculer. Ces paramètres ont été choisis de façon à être indépendants de la séquence de travail.
Ce choix nous a conduit à abandonner l'évaluation du temps de latence qui est, lui, très dépendant de la commande.
Notre évaluation est faite à partir du modèle utilisé pour le test.
Les informations supplémentaires nécessaires sont les fréquences de passage par les points singuliers et les valeurs des intervalles associés aux arcs dans le cas des modèles pondérés.
Ces valeurs peuvent être estimées, dans le cas d'une préétude, ou déduites statistiquement, à partir de mesures simples sur l'automatisme non défaillant, s'il existe déjà.
Dans tous les cas, une campagne de mesures statistique, fastidieuse, longue et coûteuse peut être évitée.

Ayant analysé les performances et les possibilités des différents modèles utilisés pour le test et la localisation de défauts en temps réel, il nous reste à envisager les réalisations possibles.
Nous retiendrons essentiellement le modèle temporisé, qui donne les meilleures performances. De plus, les autres modèles peuvent être considérés comme des sous-produits de celui-ci.

CHAPITRE IX

TEGRATION DU MECANISME DE TEST EN LIGNE DANS L'AUTOMATISME

Dans ce chapitre, nous présentons l'intégration matérielle du test en ligne dans les automatismes. Des choix d'architecture y sont présentés. Les problèmes de synchronisation dans les architectures multiprocesseurs y sont traités.

9.1 PRESENTATION DE DIFFERENTS MODES D'INTEGRATION DU MECANISME DE TEST

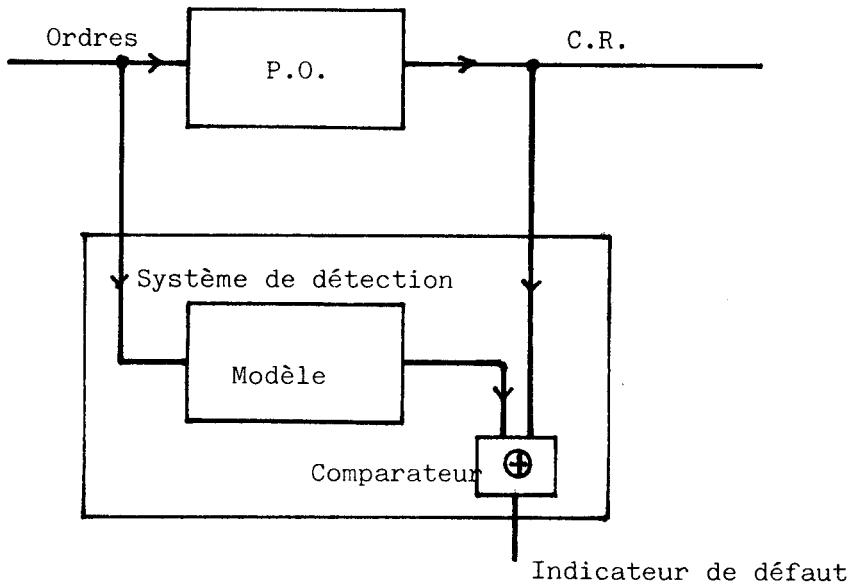
Diverses solutions sont envisagées pour introduire le test en ligne dans l'automatisme. Nous citerons:

- la méthode duplex,
- la méthode de l'inverse,
- la vérification de cohérence,
- le filtrage.

9.1.1 COMPARAISON A UN MODELE DE COMPORTEMENT

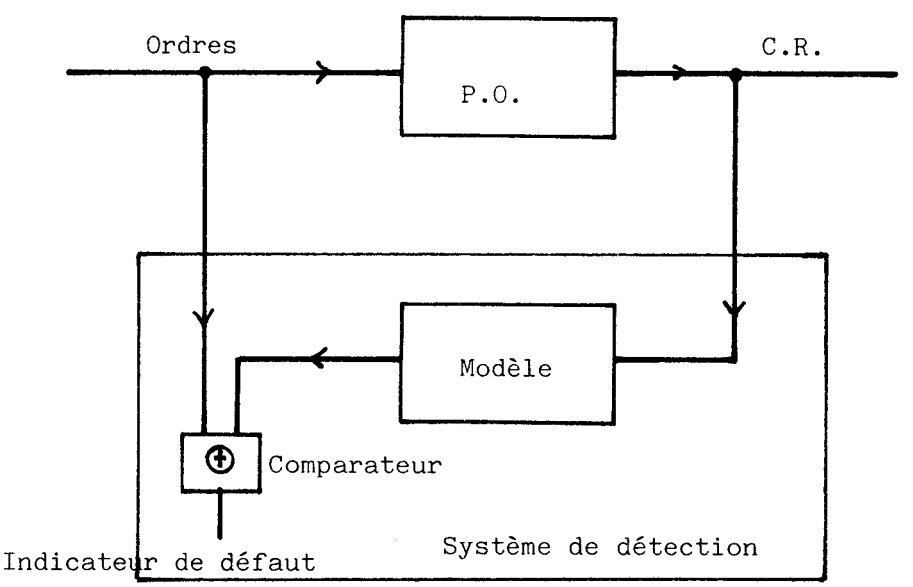
Sous cette rubrique, nous groupons les deux premières méthodes.
La méthode duplex (Fig. 9.1) correspond à une transposition des systèmes redondants par duplication. L'ensemble secondaire est ici remplacé par un modèle de l'ensemble primaire sous test.
La méthode de l'inverse présentée dans |DUR-74| a été proposée dans le cadre de la réalisation de circuits logiques autotestables (Fig.9.2). Ces deux méthodes nécessitent l'utilisation d'un circuit de comparaison difficilement autotestable.

Dans le cadre de la méthode duplex, notre modèle serait transformé de la façon suivante:



Méthode DUPLEX

- figure 9.1 -



Methode de l'INVERSE

- figure 9.2 -

Les étiquettes placées sur les arcs seraient dépendantes des seules commandes élaborées, à partir des ordres, par le modèle des préacteurs.
A ce modèle est associé un alphabet de sortie R* image de R.
A chaque sommet du modèle correspond un élément r* de R* qui est une transposition dans R* de l'élément r relatif à chaque arc (qi, r, qi).
Le comparateur met en évidence les discordances entre r(t) et r*(t) à l'instant t. La tolérance est introduite en masquant le signal d'erreur durant l'intervalle I(α ji) associé à l'arc qui amène le modèle en qi.

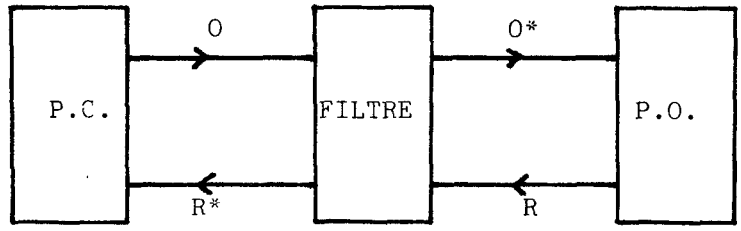
La méthode de l'inverse revient à étiqueter les arcs par les seuls éléments de R. Au modèle est alors associé un alphabet de commande C*, image de l'alphabet C. Le passage de l'état qj à l'état qi permet de choisir parmi les éléments de C* susceptibles de provoquer cette transition, l'élément c* le plus probable,compte tenu du temps nécessaire pour atteindre qi. L'application de C* au modèle inversé des préacteurs, génére alors une classe d'éléments d'ordres à laquelle doit appartenir l'ordre réel. La comparaison des deux solutions, rapidement présentée ici est faite dans |DUR-74|. La méthode de l'inverse est mise en oeuvre dans |MER-74| et |GIA-74|.

9.1.2 ANALYSE SYNTAXIQUE ET SEMANTIQUE

Cette méthode est celle que nous avons développé dans notre étude.
L'analyse syntaxique nous amène à créer dans le modèle un noeud puits (appelé souvent noeud poubelle) destiné à mettre en évidence les incohérences de syntaxe. L'introduction des temps correspond à l'apport sémantique lié à la prise en compte de la commande.
Cette solution présente l'avantage d'éviter l'introduction du comparateur. Cet élément constitue, en effet, un point dur pour les méthodes présentées en 9.1.1.Dans le contexte des automatismes lents, ce comparateur peut ne pas avoir d'existence physique. La fiabilité du logiciel de comparaison est alors peu dépendante du nombre de variables à tester.
Notre proposition englobe les méthodes utilisant les codes détecteurs d'erreurs qui correspondent au test statique.

9.1.3 FILTRAGE

Il ne s'agit pas ici véritablement d'une méthode de test différente car tous les types d'approche vus ci-dessus lui sont applicables. En fait, dans le cadre du filtrage, il y a réunion du mécanisme de détection et de réaction. La figure 9.3 illustre cette intégration. Le filtre est placé en transmission.



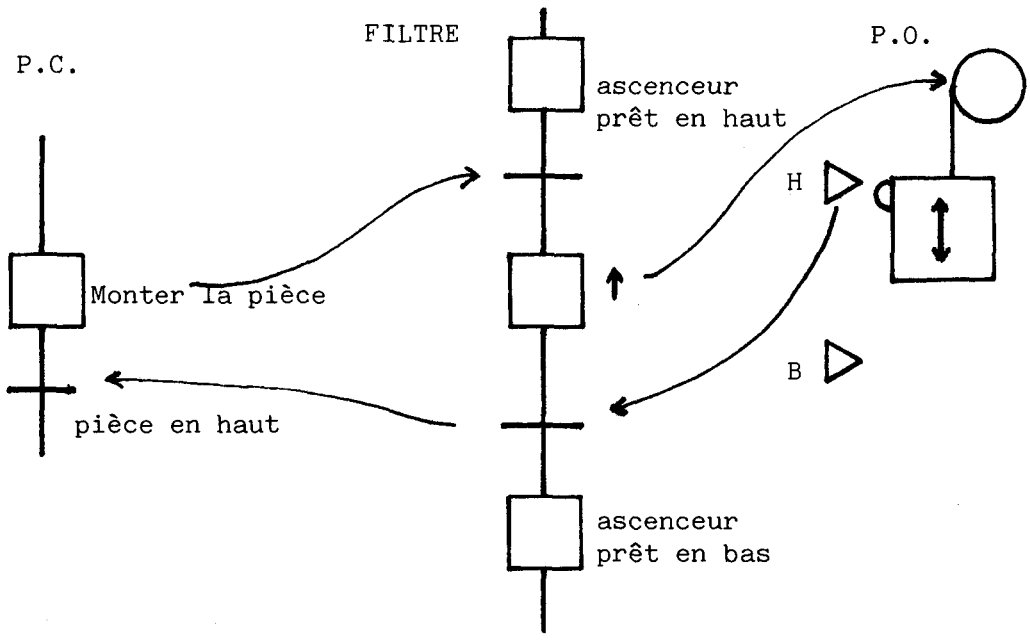
Méthode du FILTRAGE

- figure 9.3 -

Une telle configuration est proposée dans |ALA-84|; elle fait l'objet d'une étude du laboratoire d'automatique de Nancy et d'une publication
|
L'idée première est d'éviter la propagation des erreurs de la P.O. vers la P.C. et vice versa.
Notre modèle est tout à fait apte à remplir ce rôle, dans la mesure où c'est un modèle "synchronisé" sur la P.O..

Soit un alphabet R* image de R. Associons, comme au §1.1, à chaque sommet du modèle sans sommet final qf, un élément $r^* \in R^*$.
La P.C. reçoit les éléments de R*. Il est clair que seuls les éléments $r \in R$, émis par la P.O., compatibles avec l'état du modèle, influencent la P.C.. Si un tel défaut est fugitif, c'est-à-dire s'il disparaît avant que l'état défaillant ne soit accepté par le modèle, il devient non agissant.

Notre étude montre qu'un tel dispositif ne peut suffire, car de nombreuses défaillances conduisent à un blocage du modèle (chapitre VII). Le grafcet de commande se trouve alors lui-même bloqué. Ceci ne peut pas constituer un fonctionnement de sécurité.
L'introduction de chiens de garde sur les actions commandées est alors nécessaire.
De plus, l'existence de ce que nous avons appelé les arcs erronés, montre que le filtre est perméable à certaines anomalies, lorsque le codage introduit un non déterminisme. Cet aspect a été abordé dans ce que nous avons appelé les échecs de la localisation.
Les auteurs envisagent également un filtrage des ordres qui constituent une sécurité vis-à-vis des défaillances de la P.C.. Cet aspect n'entre pas dans le cadre de notre étude. Il est clair que la duplication de la P.C. pourrait donner le même résultat. En fait, l'approche originale qui est proposée constitue plus une méthodologie de description, qu'une couverture des défaillances de la P.C.. Elle permet une approche modulaire du problème, qui n'est pas sans intérêt.
La figure 9.4 illustre dans le cas simple d'un ascenseur la démarche proposée. L'ordre "monter pièce" ne sera suivi d'effet que si l'ascenseur est prêt (libre) et en haut. Cette démarche assure une sécurité vis-à-vis des erreurs de description du cahier des charges.
Le "filtre" est ici destiné à assurer l'interface entre un ensemble de process, essentiellement logiques, implanté dans la P.C., et un ensemble de ressources matérielles contenues dans la P.O..



- figure 9.4 -

Nous avons proposé une méthode de filtrage |DEF-79| complétée par un algorithme de localisation |TOU-83|. Cette méthode revient à associer à chaque transition du grafcet de commande, une expression régulière qu'il est nécessaire de vérifier pour passer à l'étape suivante. Cette expression correspond à un parcours dans le modèle dynamique simple.

Comme précédemment, toute anomalie dans la séquence de C.R. conduit à un blocage de la P.C.. Pour assurer la mise en fonctionnement de sécurité, il est alors impératif de temporiser la durée d'application de chaque action.

En fait, nous préférons séparer le mécanisme de détection et celui de réaction. Le problème de la reconfiguration et du comportement en cas d'erreur, n'a pas fait l'objet de notre étude. Néanmoins, il est possible d'envisager de réagir différemment selon la probabilité que l'erreur détectée affecte une grandeur pour laquelle la P.C. est réceptive ou sensible. Le problème est alors lié à l'efficacité de la localisation dont la procédure et les limites ont été étudiées au chapitre VIII.

9.2 MISE EN OEUVRE MATERIELLE ET LOGICIELLE

9.2.1 ARCHITECTURE MATERIELLE

La méthode de test retenue est donc celle de la vérification de cohérence sans filtrage. La figure 9.5 illustre cette implantation.

L'architecture ainsi présentée ne donne pas le meilleur compromis sécurité / fiabilité.

Si nous nous référons à l'étude faite au chapitre IV, nous avons pour l'ensemble primaire:

$$\lambda p = \lambda uc + ne\ \lambda e + no\ \lambda o + \lambda po$$
$$= \lambda pc + \lambda po$$

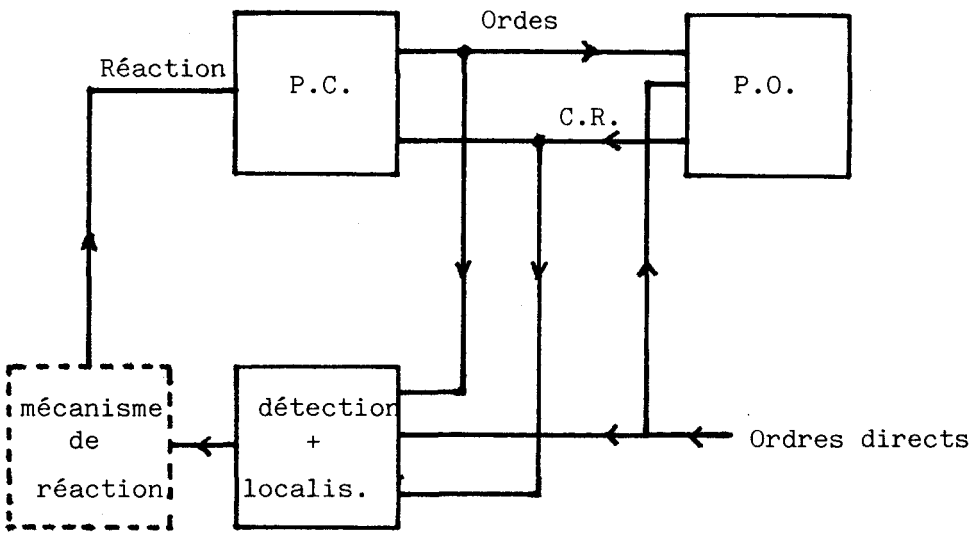
et pour l'ensemble secondaire:

$$\lambda s = \lambda ud + (ne + no)\ \lambda e \neq \lambda pc$$

où λud est le taux de panne de l'unité de détection que nous supposons du même ordre que λuc,

(ne + no) λe représente le taux relatif à l'ensemble des entrées de ce système.

λpo représente ici le taux de panne de l'ensemble de la P.O., y compris les capteurs, actionneurs, préactionneurs.



- figure 9.5 -

En l'absence d'autotest sur les unités de traitement, nous avons les taux de couverture suivants:

- $Pcp = Pct \frac{\lambda po}{\lambda p}$ où Pct est le taux de couverture du mécanisme de test de la P.O.

- $Pcs = \frac{\lambda s - (ne + no)\ \lambda e}{\lambda s}\ Pct$

Cette dernière proposition suppose qu'une panne d'E/S du dispositif de test a le même effet sur le mécanisme de réaction qu'une défaillance réelle de la P.O..

Nous avons évalué les performances en reprenant les éléments du chapitre IV. Nous rappelons que ces valeurs estimées et non évaluées statistiquement sont pessimistes.

Pour λ_{po}, nous avons adopté $\lambda_{po} = 0,8\ \lambda_p$ soit: $6\ 10^{-3}$ pannes / h alors que λ_{pc} est de $1,2\ 10^{-3}$ pannes / h. Ce chiffre semble plus réaliste que les 95% attribués à la P.O. par Siemens.

Pour le taux de couverture du système de test, nous adoptons 0,7 qui est une valeur qui peut être jugée réaliste pour le taux de couverture de l'erreur, compte tenu des valeurs évaluées dans le chapitre VIII.

Dans ces conditions, nous trouvons pour l'ensemble de l'automatisme:

pour le système sans test

MTFF = 166 h

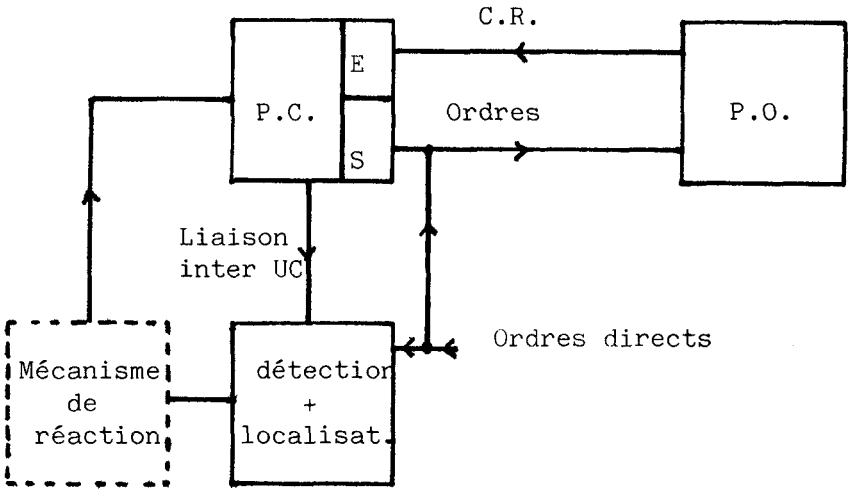
pour le système de la figure 9.5 ci-dessus, nous trouvons

MTFF = 150 h

MTFMF = 280 h

Le temps moyen entre deux interventions est de 138 h. Ce coefficient est relatif à la fiabilité inhérente.

La solution adoptée est celle qui est définie ci-dessous (figure 9.6).



- figure 9.6 -

Les ordres directs représentent les informations reçues de l'environnement, qui sont directement appliquées aux préactionneurs. Pour l'évaluation des performances, nous n'avons pas pris ces ordres en compte.

Le système de détection utilise alors les ordres et les C.R. mémorisés par l'automate. La liaison peut se faire, par exemple, par une liaison série. La valeur de λp est inchangée, mais pour l'ensemble secondaire, nous avons:

$$\lambda s \neq \lambda uc.$$

Cette évaluation est justifiée par le fait que le mécanisme de dialogue P.C. / unité de test nécessite peu de matériel et qu'il peut être facilement soumis à un mécanisme de test performant.
Si aucun test interne aux U.C., autre que la surveillance de la ligne de communication sus-visée, n'est mis en place, nous avons:

$$Pcp = Pct \frac{\lambda po}{\lambda p} \quad \text{comme précédemment}$$

et $Pcs = 0$.

Ceci conduit à:

MTFF = 166 h (valeur qui correspond au système primaire seul)

MTFMF = 360 h.

Le temps moyen entre deux interventions est ici de 158 h. Ces chiffres montrent l'intérêt, par ailleurs évident, de la deuxième architecture par rapport à la première.
Nous rappelons que nous avons supposé qu'une panne non détectée de l'ensemble secondaire ne pertube pas le fonctionnement. Cette hypothèse justifie le MTFF de 166 h ci-dessus, puisque dans cette architecture $Pcs = 0$. Ceci est évidemment optimiste, puisqu'une panne de cet ensemble peut simuler une erreur. Mais cette hypothèse a le même effet dans les deux cas.
L'introduction d'autotest dans les U.C. permettrait évidemment d'améliorer les performances globales. Le taux de panne de l'automate et de l'unité de test donne pour ces deux ensembles un MTFF de 660 h.
Cette architecture a été utilisée et présentée dans |NAI-83|. Elle était formée par un automate programmable PB 100 de Merlin Gérin associé à un microcalculateur.

9.2.2 MATERIEL COUVERT PAR LE TEST

Dans le cadre de l'architecture retenue (fig. 9.6), il est clair que les dispositifs E/S de l'automate sont considérés comme fai-

sant parti de la P.O.. Nous nous proposons de considérer ici l'automatisme dans son environnement. Nous avons envisagé l'existence d'ordres directement appliqués à la P.O. sans passer par la P.C.. La prise en compte de ces informations est indispensable au bon fonctionnement de la surveillance. Il reste à analyser la portée du test,dans le cadre des mécanismes de communication entre les automatismes ou avec les opérateurs.

Les différentes E/S de la P.C.

Nous distingons:

- les variables internes à l'automatisme;
- les grandeurs de communication.

Les variables internes correspondent aux grandeurs (ordres, C.R.), échangées entre P.O. et P.C.. L'absence de C.R. sur une évolution (utilisation de temporisations), supprime toute possibilité d'intervention du système de test.

Les grandeurs de communication permettent la synchronisation entre les automatismes, l'adaptation de leur tâche aux besoins de la production. Ces échanges se font en boucle ouverte ou par un dialogue de type appel-réponse (shake-hand). Ce dernier procédé permet un contrôle par le mécanisme de test. Deux cas sont envisagés.

La requête est issue de l'automatisme étudié, le système de surveillance contrôle le délais de retour de l'accusé reception. Les circuits E/S/cablage mis en place pour cet échange, sont couverts par le test.

L'appel constitu un événement auquel doit répondre l'automatisme, c'est en implantant un mécanisme de test sur le dispositif appelant, que les éléments précités seront testés. C'est donc au système qui a l'initiative qu'il appartient d'assurer le test de la procédure.

Cette remarque s'applique au besoin, aux opérateurs. Lorsque le conducteur prend l'initiative de la communication, le test est visuel. Cette démarche est courante. Lorsque l'automatisme est le demandeur, cette mission de surveillance doit être confiée au mécanisme de test mis en place. Ceci permet de vérifier le

bon état du matériel mis en jeu dans l'échange, comme la vigilance de l'opérateur! Cette proposition s'inscrit dans le cadre de la conduite de processus assistée par ordinateur.

Les communications en boucle ouverte (type monologue), sont à éviter dans la mesure où elles ne se prêtent pas au contrôle des éléments utilisé dans l'échange.

9.2.3 SYNCHRONISATION DANS UNE ARCHITECTURE BIPROCESSEUR

L'architecture envisagée correspond à la mise en place de deux processeurs spécialisés. Il faut donc résoudre les problèmes de synchronisation entre les deux machines et la gestion du flux de données.

Nous tirons de |COU-80| la représentation graphique suivante:

- chaque zone de données est représentée par un rectangle,
- chaque opérateur par un cercle,
- le cheminement des données est représenté par des arcs.

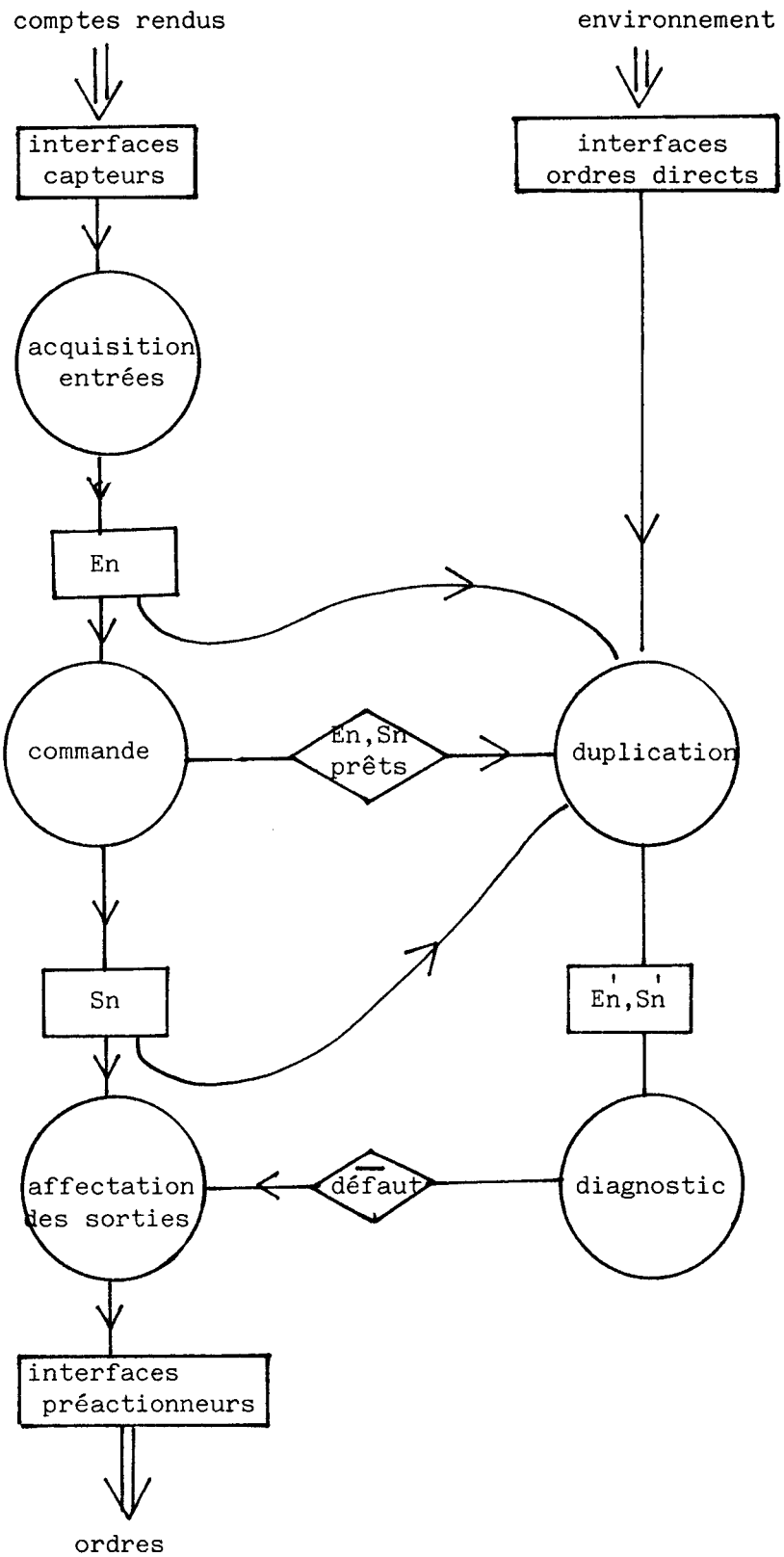
  Tout arc entre un rectangle et un cercle spécifie une donnée d'entrée de l'opérateur.

  Tout arc joignant un cercle à un rectangle correspond à une modification des données par l'opérateur.
- Les synchronisations entre opérateurs (pris ici au sens de tâches) sont représentées par un arc traversant un losange.

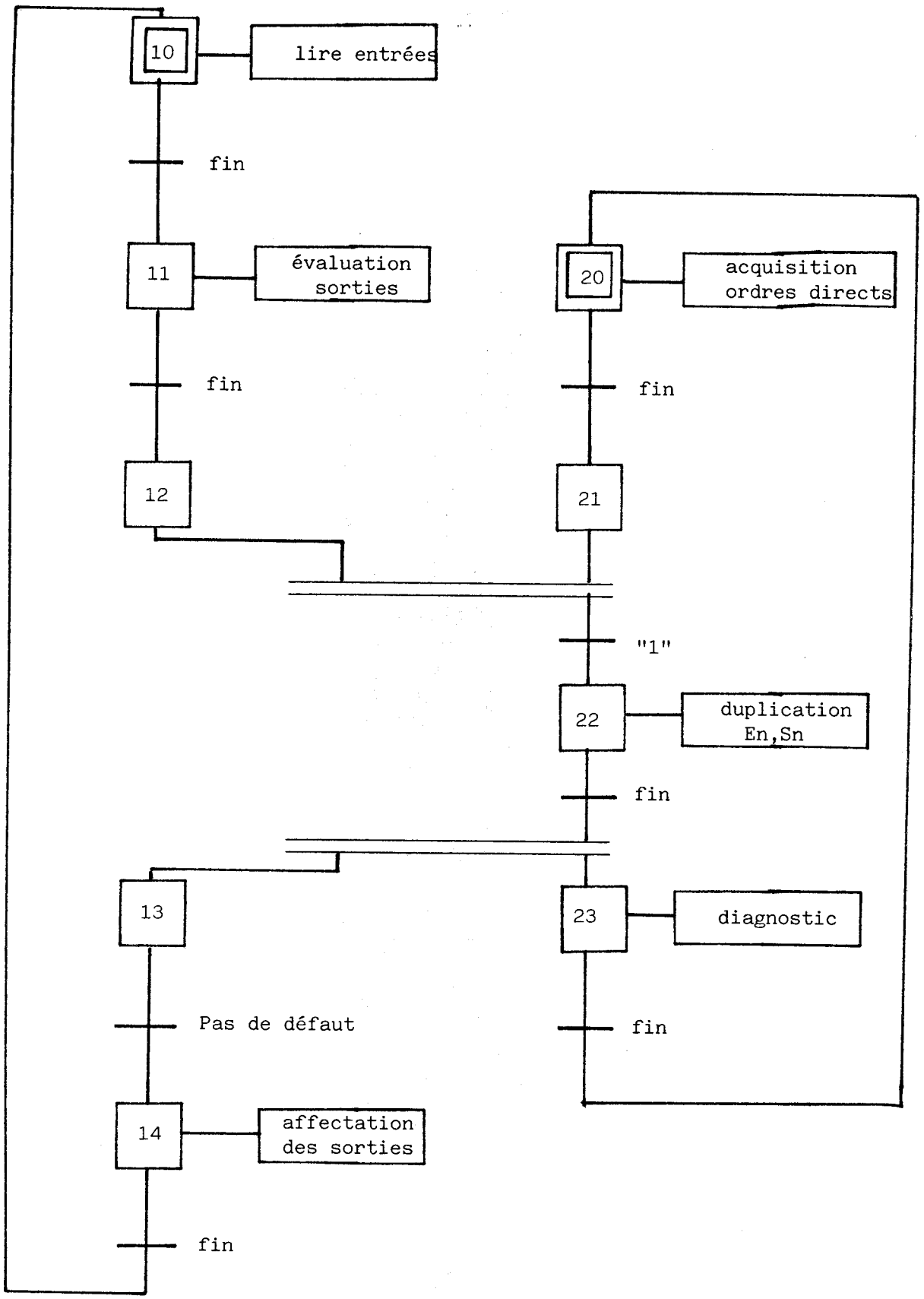
Les zones de données sont éventuellement des interfaces d'entrées / sorties repérées par des flèches doubles.
Les synchronisations peuvent évidemment se faire sur des événements extérieurs.

La figure 9.7 représente l'évolution des données au cours d'un cycle de traitement de l'automate.

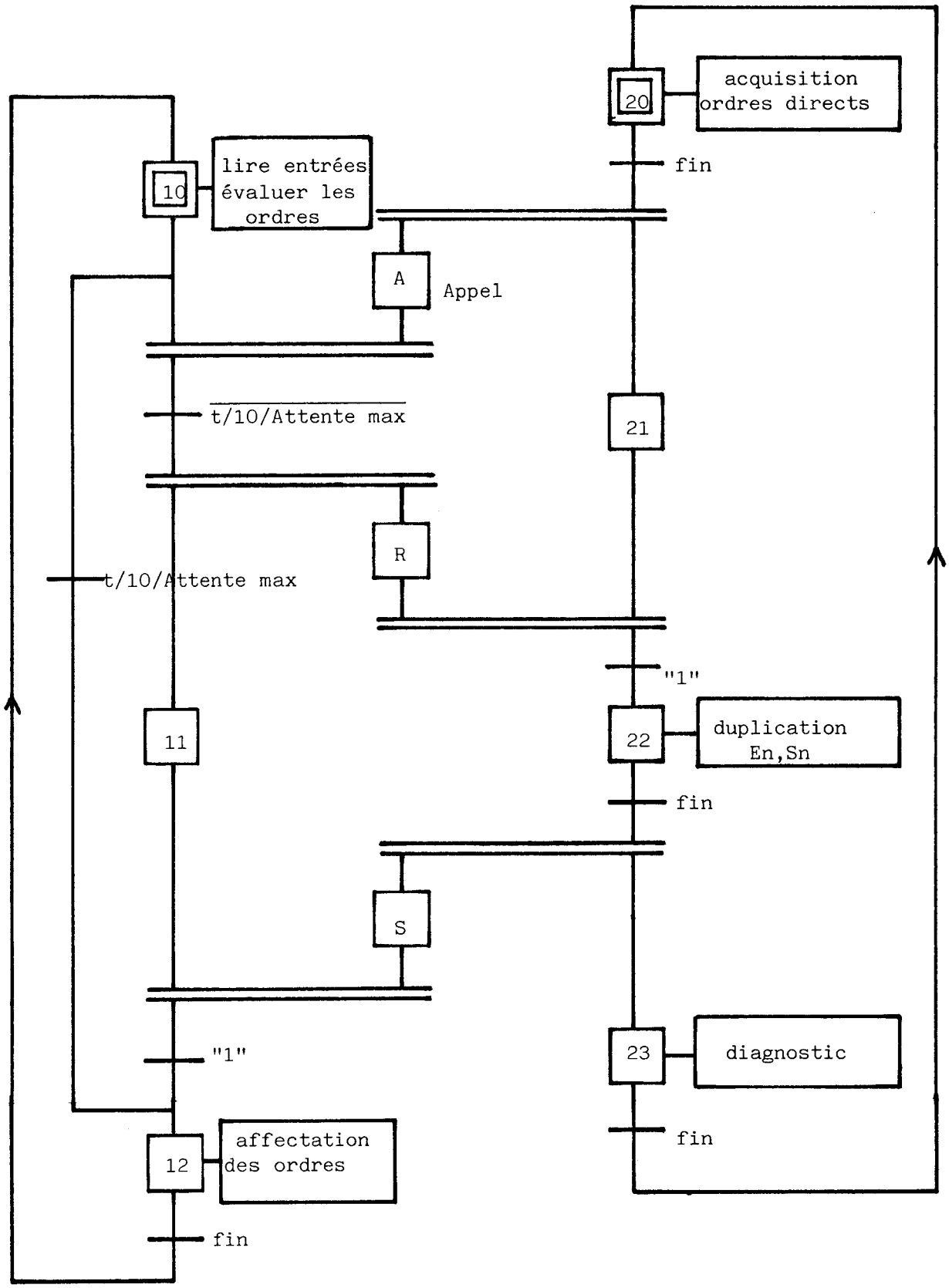
La figure 9.8 représente la synchronisation entre tâches que l'on peut déduire du diagramme précédent.


comptes rendus
environnement
interfaces capteurs
interfaces ordres directs
acquisition entrées
En
commande
En,Sn prêts
duplication
Sn
En',Sn'
affectation des sorties
défaut
diagnostic
interfaces préactionneurs
ordres

- figure 9.7 -


10
lire entrées
fin
11
évaluation sorties
fin
12
20
acquisition ordres directs
fin
21
"1"
22
duplication En,Sn
fin
13
23
diagnostic
Pas de défaut
fin
14
affectation des sorties
fin

- figure 9.8 -


20
acquisition ordres directs
fin
10
lire entrées évaluer les ordres
A
Appel
t/10/Attente max
21
R
t/10/Attente max
"1"
11
22
duplication En,Sn
fin
S
"1"
23
diagnostic
fin
12
affectation des ordres
fin

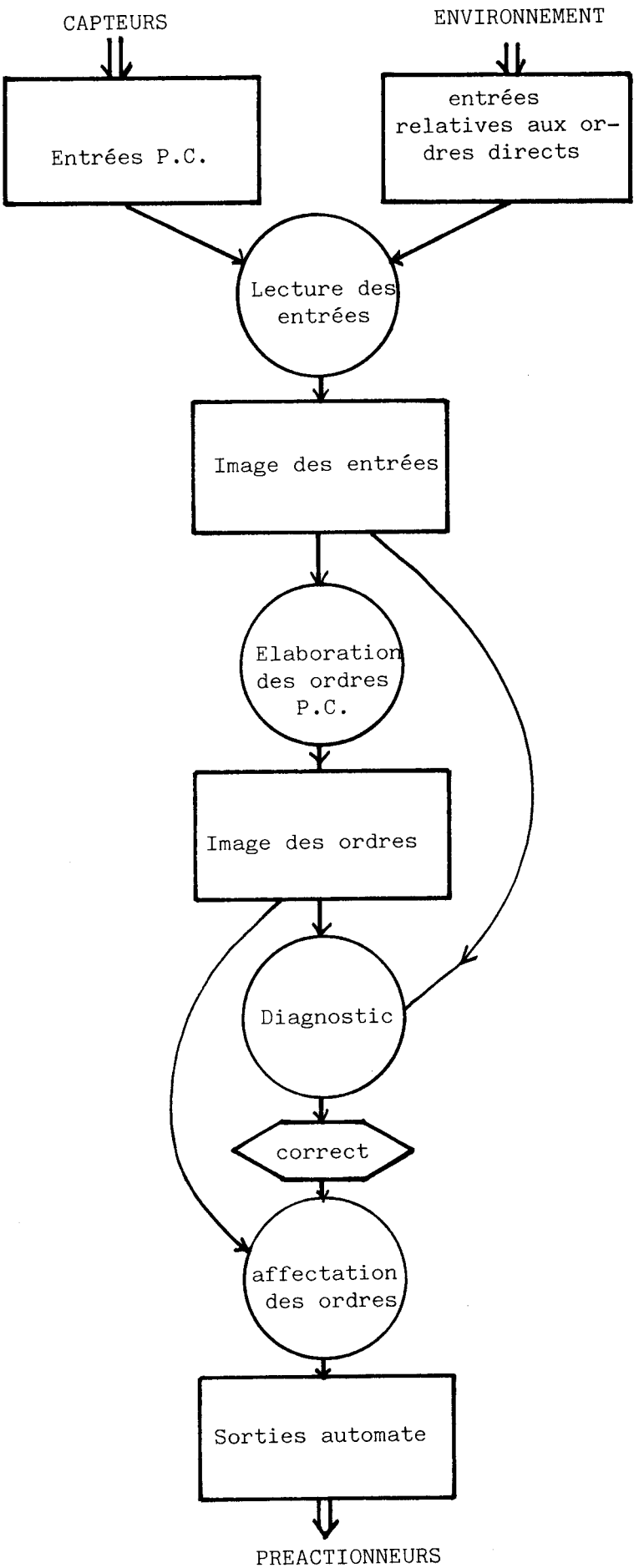
- figure 9.9 -

Les communications entre tâches d'un système réparti sont à l'origine de conflit. D'après |COU-83|, nous voyons que dans cet exemple simple, il est possible de donner l'initiative du rendez-vous, à l'un quelconque des automates. Nous choisissons de donner l'initiative de ce rendez-vous au système de test. Cela permet de placer, sur le système de commande, un chien de garde qui évite le blocage de la commande en cas de bouclage accidentel du système de test (figure 9.9).
Les étapes d'appel / réponse sont représentées par un booléen dans une zone mémoire à double accès.
Dans le cas de l'automate PB 100, ce double accès est en fait virtuel, car c'est l'automate lui-même qui écrit ou qui lit dans sa propre mémoire. L'information correspondante est échangée avec le micro calculateur par la ligne série. Ce moyen règle le problème des conflits d'accès, mais l'intervention répétée du système de test ralentit le traitement.

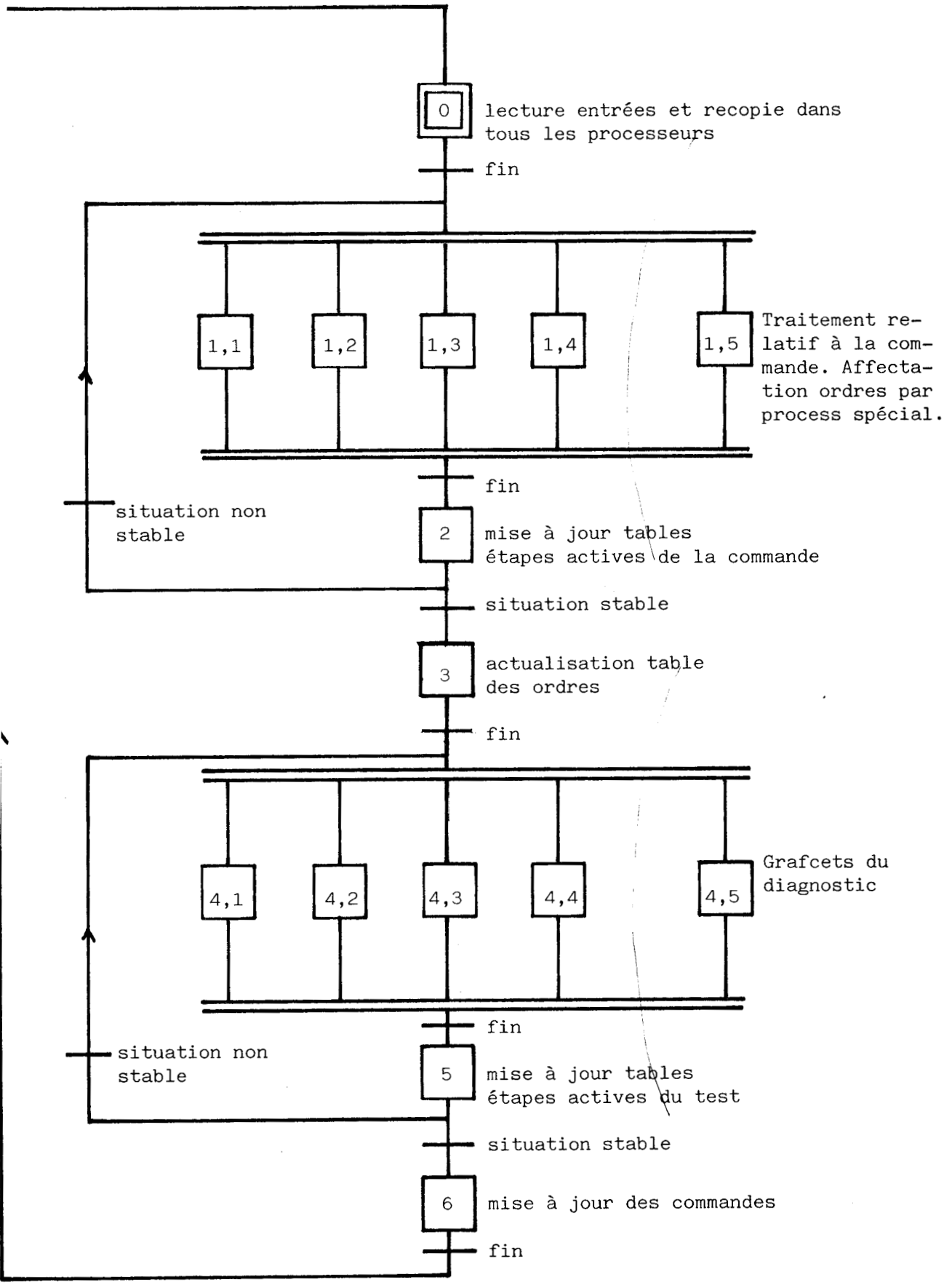
9.2.4 ORGANISATION DANS UN ENVIRONNEMENT MULTIPROCESSEURS

Le traitement relatif à la commande dans les systèmes importants correspond à la gestion de plusieurs grafcets ou de graphe d'états.
Il en est de même pour le test.
Il est possible de distribuer la gestion de ces différents graphes, au hasard, entre plusieurs processeurs, à condition qu'ils travaillent sur des données identiques et qu'ils connaissent la structure de l'ensemble des graphes.
La gestion des grafcets impose de travailler sur les données figées, actualisées seulement lorsque l'ensemble des graphes a été examiné.
Le traitement de l'ensemble P.C. / test peut être confié à un seul système multiprocesseurs, considéré comme monoprocesseur. Il suffit alors de créer des rendez-vous chaque fois que l'actualisation de données doit être faite, la gestion du flux de données étant alors celle d'un système monoprocesseur (figure 9.10).

Chaque processeur étant banalisé en dehors des processeurs d'entrée sortie, il suffit d'organiser des rendez-vous chaque fois qu'un traitement est achevé. Cette organisation des tâches est donnée figure 9.11.


CAPTEURS
ENVIRONNEMENT
Entrées P.C.
entrées relatives aux ordres directs
Lecture des entrées
Image des entrées
Elaboration des ordres P.C.
Image des ordres
Diagnostic
correct
affectation des ordres
Sorties automate
PREACTIONNEURS

- figure 9.10 -


0
lecture entrées et recopie dans tous les processeurs
fin
1,1
1,2
1,3
1,4
1,5
Traitement relatif à la commande. Affectation ordres par process spécial.
fin
situation non stable
2
mise à jour tables étapes actives de la commande
situation stable
3
actualisation table des ordres
fin
4,1
4,2
4,3
4,4
4,5
Grafcets du diagnostic
fin
situation non stable
5
mise à jour tables étapes actives du test
situation stable
6
mise à jour des commandes
fin

- figur 9.11 -

Une telle architecture peut être bâtie autour du bus V.M.E. par exemple. Chaque carte processeur dispose de ses ressources propres, c'est-à-dire une zone mémoire suffisante pour contenir l'ensemble des programmes relatifs à l'automatisme. Les capacités mémoires actuellement disponibles le permettent pour des problèmes de taille respectable.
De plus, chaque carte dispose d'une mémoire à double accès utilisée comme boîte à lettres.
Cette mémoire est destinée à recevoir:

- les messages de gestion du réseau,
- les données entrées/sorties/états des grafcets.

L'accès au bus V.M.E. est alors géré par passage d'un jeton par le bus série V.M.X..
Une ou plusieurs cartes sont dédiées aux tâches d'E/S et de liaison éventuelle avec un réseau local.

Si elles ne sont pas occupées par ces tâches spécialisées, elles participent à la gestion des graphes.
L'intérêt d'une telle organisation est évident:

- L'augmentation du nombre de processeurs doit permettre un gain de vitesse d'exécution. Il faut tenir compte ici de la perte de temps liée aux échanges entre processeurs.
- En cas de défaillance d'un processeur, la reconfiguration du système est lié à la politique de gestion du jeton. Lorsqu'un processeur défaillant n'accuse pas réception du jeton, le dernier processeur maître passe, en principe, le jeton au processeur suivant.
- Un tel système permet une redondance du traitement.
- Au pire des cas, le processeur d'E/S, disposant de l'ensemble des informations nécessaires à la gestion, peut assurer tout ou une partie de la commande en cas de défaillance grave. Le point dur est essentiellement constitué par les E/S physiques et le processeur correspondant.

Nous avons abordé l'étude du système d'exploitation d'un tel réseau dans l'optique restreinte de l'automate sûr de fonctionnement.
Toutefois, cette étude n'est pas, à ce jour, suffisamment avancée, pour faire l'objet d'une publication. Elle reste un objectif pour nos travaux à venir.

Dans ce chapitre, nous avons choisi d'utiliser les images des E / S utilisées pour la commande plutôt que les valeurs observables sur les lignes physiques correspondantes. Ce choix est justifié par la sûreté de fonctionnement globale de l'automatisme. Deux architectures multi processeurs sont proposées et dans chaque cas, le contrôle du flux de données et la synchronisation des tâches sont abordés.
Il reste toutefois à voir l'organisation logicielle du mécanisme de test et plus particulièrement la création du modèle.

CHAPITRE X

ELABORATION DU MODELE PONDERE

Dans ce chapitre, nous étudions la mise en oeuvre du test en ligne par analyse syntaxique et sémantique des C.R..
Deux aspects sont ici abordés:

- l'élaboration d'une structure de données correspondant à un modèle,
- son exploitation.

.1 ORGANISATION DU MODELE ET EXPLOITATION TEMPS REEL

Nous avons vu au chapitre VII que le modèle pondéré est un automate non pondéré, dit automate de référence, auquel sont associés:

- un prédicat pour chaque arc,
- un compteur à chaque sommet.

De plus, un automate est un graphe auquel est associé un noeud d'entrée et des noeuds de sortie. Ces remarques permettent de choisir une structuration de données relative au modèle retenu.

1.1 REPRESENTATION GENERALE D'UN AUTOMATE PONDERE: STRUCTURE DE DONNEES

A chaque sommet de l'automate est associé un pointeur qui permet de localiser l'ensemble des données relatives à ce sommet.

A chaque sommet de l'automate correspond:

- un indicateur de type permettant de différencier les noeuds final et non final,
- un ensemble de données organisé en fonction du type considéré.

Pour un noeud non final, chaque arc issu du sommet considéré est représenté par un enregistrement comprenant:

- l'identification de l'étiquette,
- deux valeurs numériques représentant les bornes de l'intervalle associé,
- un pointeur permettant l'accès direct au sommet extrémité de l'arc si la condition d'évolution selon cet arc est satisfaite dans l'intervalle considéré,
- un pointeur vers le sommet final.

L'étiquette est une fonction booléenne (en principe, un monome). L'identification de l'étiquette peut être:

- une description de ce monome,
- l'adresse renvoyant à cette description,
- l'adresse d'un sous programme d'évaluation.

La liste de ces enregistrements peut être une file séquentielle ou une liste linéaire chaînée. Cette dernière représentation est plus souple pendant la phase de création du modèle.

Dans ces conditions, nous disposons d'un chaînage en profondeur permettant l'évolution dans le graphe, et un chaînage en largeur donnant accès aux différents arcs ayant même origine quelqu'en soit le nombre.

Les différents champs de données pour un noeud final sont:

- l'indicateur de type,
- dans le cas du test, des "messages" permettant une aide à la localisation du défaut.

L'accès à ces informations est obtenu à partir d'un indicateur de type de défaut positionné au moment de la transition vers ce sommet final.

La modélisation des préactionneurs correspond à un ensemble de graphes d'état et à un combinatoire local. Une structure très voisine est donc utilisée; seuls l'étiquette et le noeud extrémité sont retenus pour chaque arc.

.2 AMENAGEMENTS APPORTES PAR RAPPORT AU MODELE ORIGINAL

- Chaque grandeur mesurée permet de définir une sous P.O.. Nous supposons la P.O. décomposée en ses sous P.O.. Dans ces conditions, chaque modèle a un seul noeud final.

- En dehors des arcs ayant le sommet initial qo comme origine, l'évaluation complète du monome formant l'étiquette d'un arc n'est pas indispensable. Pour chaque arc de la forme (qi, rij, qj), l'étiquette est réduite à la seule variable qui distingue rij de l'étiquette rii de l'arc (qi, rii, qi). La forme affirmée ou niée retenue est celle qu'avait cette variable dans l'expression rij.
  Cette transformation correspond à une compression de données qui conduit à une diminution de taille mémoire et à un gain en vitesse de traitement.

- L'arc (qi, rii, qi) n'a donc plus d'étiquette. Seul l'intervalle associé (constitué ici d'une valeur maximale) est conservé. L'aspect chien de garde de cet arc est alors très net.

- Pour tout élément r $\in$ R, qui ne peut être affecté de façon non équivoque à un sommet final ou non final du modèle localisé, il est créé un arc (qo, r, qo). Cette convention ne perturbe que très modérément la phase de localisation dans la majorité des cas. Toutefois, des P.O. à capteur incrémental ne sont plus localisables.

La stratégie du test est la suivante:

- les modifications de commande sont admises uniquement aux points d'entrée dans les P.S. (valeur de pi = 1);
- toutes exceptions à la proposition ci-dessus constitue une nouvelle date d'initialisation;
- le passage par le noeud final provoque l'émission d'un indicateur de défaut et constitue une nouvelle initialisation.

Les deux premiers points sont liés à la limitation de validité du modèle temporisé (chapitre VII, § 5). Ceci se traduit par la création d'un identificateur de commande en cours et d'un drapeau permettant de mettre en évidence le changement de sens d'évolution.
De même, chaque commande se voit attribuer un indicateur de classe de type "avance", "recul" ou "arrêt".

10.1.3. STRUCTURE DE DONNEES ET REPRESENTATION D'ETAT

La structure de données est adaptée aux aménagements introduits ci-dessus. Elle prend la forme suivante.

A chaque sommet est associée une liste d'enregistrements correspondant chacun à une commande.

Chaque enregistrement contient:

- le nom de la commande,
- les temps de maintien limités dans le P.S. correspondant,
- les points d'entrée d'une liste d'arcs.

Les deux dernières informations (temps de maintien et points d'entrée) sont doublées. En effet, les intervalles de temps qui s'y rapportent dépendent du fait qu'il y ait ou non changement du sens d'évolution au P.S. considéré.

Pour chaque arc, l'étiquette est réduite à la définition du nom et de la valeur attendue d'une variable. Le reste de la description est inchangé.

La figure 10.1a représente la structure de données retenue.

L'état de la P.O. est représenté par un ensemble d'indicateurs qui suivent l'évolution observée figure 10.1b.

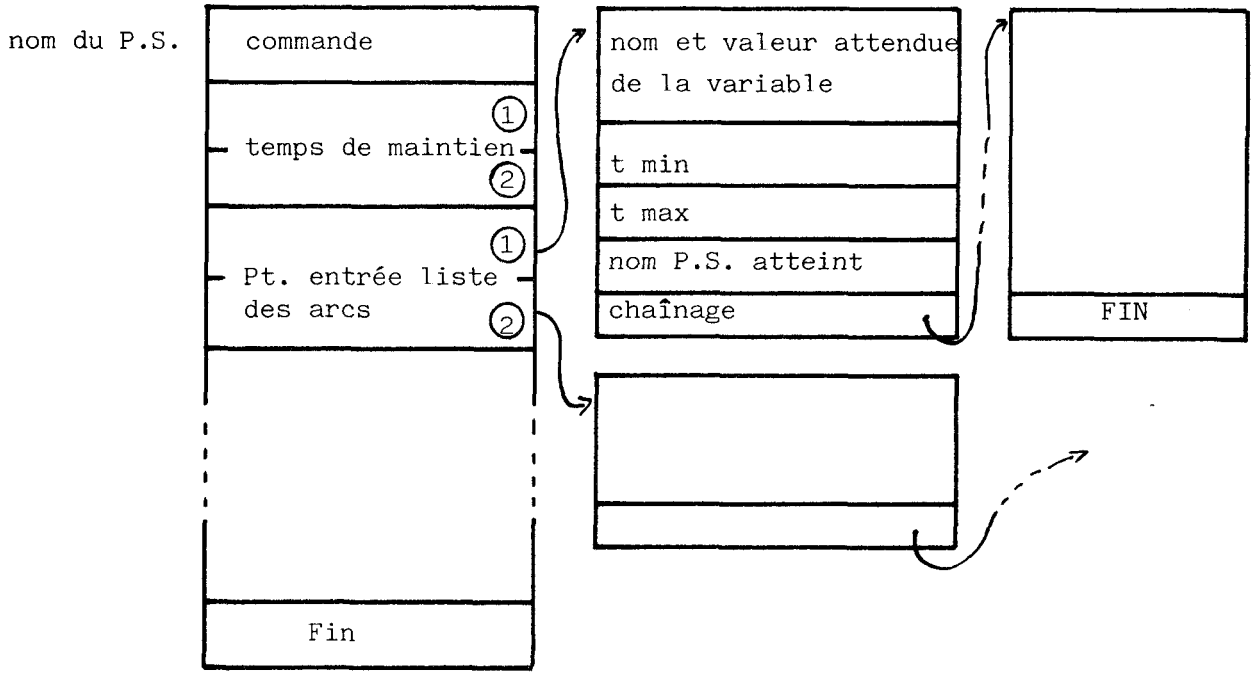
Nous trouvons:

- la localisation du point d'entrée de la chaîne d'arcs à tester (ou nom du P.S. si une nouvelle commande est acceptable);
- la valeur courante du temps limite de maintien dans cet état;
- le nom de la commande en cours;
- des indicateurs booléens.

Ces indicateurs permettent de caractériser l'état du modèle, à savoir:

- il est totalement localisé,
- il est partiellement localisé,
- il accepte une nouvelle commande,
- classe de la vitesse par laquelle le P.S. a été atteint.

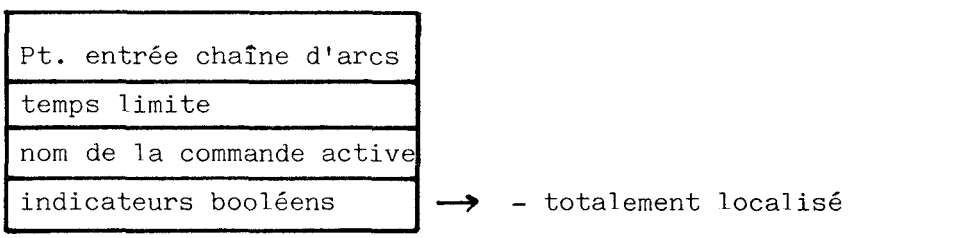
Compte tenu de l'étude faite au chapitre 7, et des conditions de localisation adoptées ci-dessus, nous dirons que le modèle non localisé devient partiellement localisé lorsqu'un C.R. peut être affecté de façon non équivoque à un sommet.



(1) Information à prendre en considération s'il n'y a pas de changement de sens d'évolution au P.S..

(2) Information retenue dans le cas contraire.

- a -



- b -

Structure de données et représentation d'état du modèle temporisé.

- figure 10.1 -

Comme il est impossible de préciser la position relative dans le P.S. correspondant, les temporisations ne sont pas utilisables. Le modèle sera complètement localisé seulement lorsque ce P.S. sera quitté.

Le modèle accepte une nouvelle commande dès qu'un nouveau P.S. est atteint. Il en sera ainsi jusqu'à ce qu'une commande, relative à une vitesse non nulle, soit appliquée.
A cet instant, il est possible de déterminer si la P.O. doit traverser le P.S. ou, au contraire, s'il y a un changement de sens. En tenant compte de cette information, les traitements suivants sont effectués en séquence:

- inscription du point d'entrée dans la chaîne d'arcs à tester,
- recopie du temps maximum de maintien dans l'état,
- écriture du nom de la commande en cours,
- positionnement de l'indicateur booléen caractérisant la classe de la commande appliquée.

A chaque période d'une horloge, les compteurs de temps de maintien pour toutes les P.O. en cours d'évolution sont décrémentés. Ces P.O. se caractérisent par le fait qu'elles n'acceptent pas une nouvelle commande. Tout passage par zéro d'un tel compteur indique une défaillance.

La scrutation du modèle proprement dite se résume en un balayage de la liste des arcs. Chaque fois que la valeur attendue d'une variable attachée à un arc est atteinte, il est vérifié que le temps écoulé est bien compris dans l'intervalle de tolérance. S'il en est ainsi, le nom du sommet atteint est porté en lieu et place du point d'entrée de la chaîne à tester. L'indicateur est positionné pour signaler qu'une nouvelle commande est possible. Sinon, un message de défaut est envoyé et le modèle est placé dans l'état non localisé.

Ces différents algorithmes permettent de réaliser le test en ligne par exploitation d'un automate pondéré. Le mode de représentation et de traitement est à rapprocher des méthodes de traitement des grafcets orientées données |THE-81|. Afin d'uniformiser le traitement relatif à la commande et au test, il est possible de représenter les modalités du test par des grafcets |HAC-84|, |DEF-84|, |ADI-84|.
Cette formule se présente comme une variante sous optimale de la méthode de la méthode proposée ici, par rapport au temps de traitement.

10.2 MODELISATION PAR AUTOAPPRENTISSAGE

La création du modèle doit être envisagée sous l'angle utilisateur. Cette création est faite en deux parties distinctes qui sont:

- la modélisation de la trajectoire,
- la modélisation des préactionneurs.

Ces deux descriptions sont interfacées par les commandes.

Le modèle est construit en deux temps:

a) L'automate de référence est constitué. Cette étape correspond à la création dynamique de la structure de données:
   - identification des étiquettes,
   - constitution des chaînages.

b) Les intervalles associés aux arcs y sont ensuite déterminés, soit par évaluation préalable au cours de l'étude, soit en cours d'exploitation.

Ces deux aspects peuvent être utilisés conjointement. Les avantages de la détermination dynamique des bornes en cours d'exploitation sont:

- la simplicité de mise en oeuvre,
- le suivi du vieillissement de l'automatisme par mise en évidence des dérives.

Pour ce qui est de la phase (a), différentes options sont envisageables, avec plus ou moins de facilité, selon le modèle choisi.
Deux approches sont possibles:

- le modèle est obtenu à partir d'un langage de description,
- il est déduit d'un historique constitué, soit en cours d'exploitation, soit par un cycle d'apprentissage.

Ces différents aspects sont abordés ci-après.

.2.1 MODELISATION DES TRAJECTOIRES DANS LE CADRE DU CYCLE DE TRAVAIL REPETITIF

Cette méthode a donné lieu à des réalisations industrielles proposées par Gould Modicon et Westinghouse |CHE-81| et |ELL-82|, |INA-82|.
La création du modèle est alors faite par autoapprentissage après enregistrement de la séquence observée, lorsque l'automatisme est soumis à un cycle de travail.

Au cours de la phase dite d'apprentissage, il est procédé à l'enregistrement des noms de variables qui changent de valeur, dans l'ordre où ces modifications apparaissent. Simultanément, les temps qui séparent deux évolutions consécutives de C.R. sont enregistrés.
Le modèle ainsi obtenu correspond à une liste séquentielle d'états. C'est un cas particulier du graphe sans cycle que nous avons proposé au chapitre VII,§ 3.

Si un tel système est d'une mise en oeuvre simple, donc séduisante, son intérêt est limité par les contraintes qu'il suppose. En effet, en cas de parallélisme imposé par la P.C., l'ordre des modifications de C.R. peut ne pas être unique.
La proposition faite, qui consiste à considérer comme simultanés les événements rapprochés, est peu satisfaisante. S'il est possible, en effet, de considérer dans ce cas, l'ordre comme indifférent, cette hypothèse conduit à une augmentation de l'intervalle de confiance. Or, comme le montre l'étude des performances, ce paramètre intervient directement dans la qualité du test.

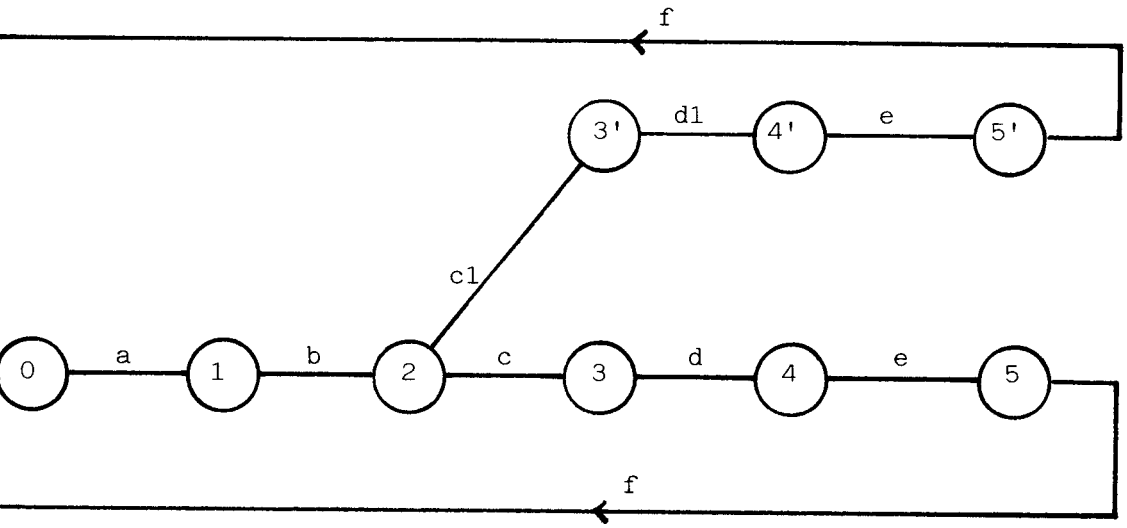
Pour notre modèle, nous avons admis que plusieurs séquences sont admissibles. Ceci permet de prendre en compte le parallélisme.

Apprentissage

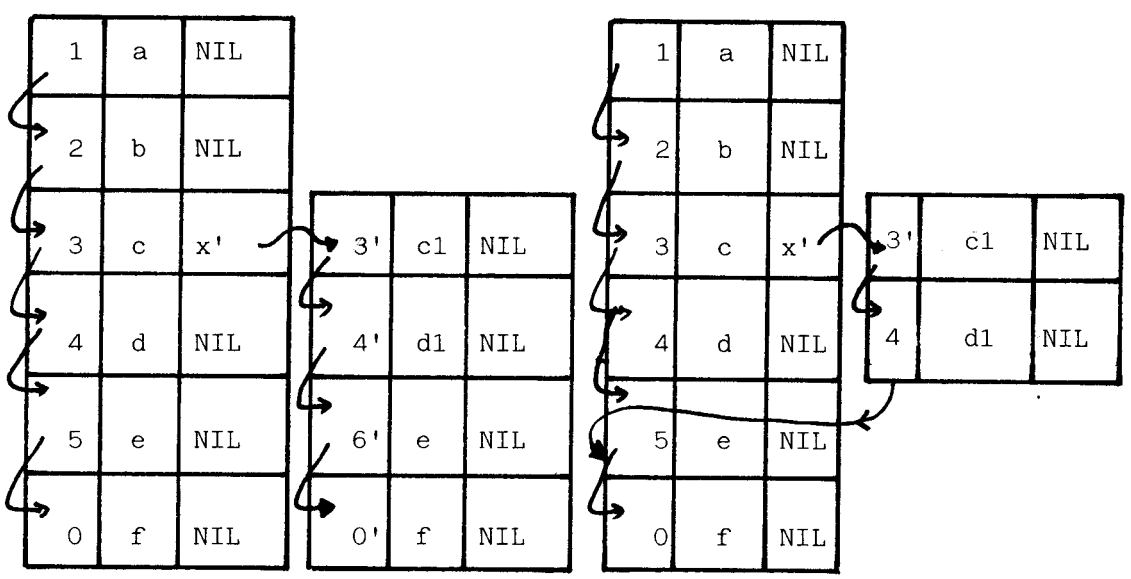
Supposons un échantillon formé de deux séquences générées par la P.O. saine, soumise à la même séquence de travail. Ces deux séquences sont représentées par une liste, qui contient les noms des variables, dans l'ordre où elles ont subi une modification de niveau. Elles sont complétées par les temps écoulés entre deux modifications consécutives. Ces listes correspondent à deux automates pondérés de même origine qo. Si l'on élimine les arcs qui reviennent à l'état initial qo, ce qui est dû au fait que le cycle de travail est répétitif, ces séquences ont pu être générées par un arbre (figure 10.2a)appelé ACM (automate canonique maximum).

La figure 10.2b donne une image de la structure de données obtenue à partir de deux séquences différenciées par un nombre fini de permutations.

Si les listes ont le même nombre d'éléments, et s'il existe des sous listes identiques, de même rang, il est possible de réduire la taille du modèle en fusionnant ces sous listes (figure 10.2c).



- a -



- b -

- c -

- figure 10.2 -

Ce genre de transformation, comme toute méthode d'inférence, risque de créer des chemins non acceptables. Ceci se vérifie simplement sur l'exemple suivant. Soit U,W,V1,V2,S1,S2 des séquences de C.R. telles que longueur V1= longueur V2 et longueur S1= longueur S2. Soient U V1 W S1 et U V2 W S2 deux séquences de même longueur.
La transformation envisagée amène à considérer comme acceptable toute séquence de la forme:

U (V1 + V2) W (S1 + S2) écrite en utilisant le symbolisme des expressions régulières |CHI-67|.
La séquence U V1 W S2 par exemple est donc acceptée alors qu'elle n'a pas été observée.

Nous ne poursuivons pas plus avant cette étude, car le modèle obtenu est, à notre avis, très mal adapté en cas de parallélisme.
Cette opinion est justifiée par les faits suivants:

- La taille du modèle a une croissance exponentielle avec le degré de parallélisme malgré les réductions envisagées ci-dessus.
- L'introduction de la pondération y est une erreur, puisque la modification de C.R. sur une grandeur mesurée ne peut pas être considérée comme point de régénération pour d'autres grandeurs indépendantes.

Ce dernier point semble avoir été souvent oublié par les auteurs qui préconisent ce modèle. En effet, il suffit à rejeter la méthode en cas de parallélisme, même si l'ordre des modifications de C.R. y est immuable.

Nous proposons donc pour remédier à cet inconvénient majeur,de considérer la P.O. comme un ensemble de sous P.O.. Une sous P.O. est, nous le rappelons, l'ensemble des matériels associés à une grandeur mesurée, donc à une trajectoire.

Deux problèmes se posent alors:

- l'interaction entre sous P.O.,
- l'introduction de cette décomposition dans la structure de données.

Le premier point sera développé ultérieurement.

Dans l'hypothèse où il n'y a aucune interaction, le deuxième point conduit à un partitionnement des capteurs en fonction des trajectoires. Il y a alors un modèle pour chaque trajectoire. La partition des entrées peut être spécifiée en associant à chaque sous P.O. la liste des variables à prendre en considération (ou une table de masquage si le nombre de variables total est peu important).
La constitution de cette liste (ou de la table des masquages) peut être faite par l'opérateur en phase de configuration (choix d'un langage). Elle peut également être élaborée automatiquement à partir d'une marche manuelle s'il est possible de parcourir toutes les trajectoires une à une de cette façon.

10.2.2 MODELISATION DES TRAJECTOIRES EN DEHORS DU CONTEXTE DU CYCLE DE TRAVAIL

Seule la recherche de l'automate de référence relatif à la trajectoire est abordée ici.

10.2.2.1 Domaine d'application

a) Hypothèse 1
Chaque trajectoire définit une sous P.O.. Il n'y a aucune interaction entre les sous P.O. ainsi constituées. En particulier, cette décomposition constitue une partition de l'ensemble des capteurs.

b) Hypothèse 2
Si le passage à vitesse positive d'un point singulier à un autre, entraîne la mise à "1" d'une variable booléenne, le passage par ce même point au retour force à "0" cette même variable (à la distance d'hystérésis près). Ceci exclut les capteurs de sens de passage.

La démarche de la constitution du modèle est alors la suivante:

- Déclarer pour chaque grandeur mesurée la liste des variables $x \in X$ significative. Compte tenu de l'hypothèse 1, cette décomposition forme une partition de X en Xj qui constitue également une partition de l'alphabet des comptes rendus R.
- Enregistrer une suite de référence globale S au cours d'une période de fonctionnement de l'automatisme . Cette suite, relative aux événements normaux, contient les noms des variables qui changent d'état, représentés sous forme affirmée si la valeur atteinte est 1, niée sinon.

A cette liste ordonnée est associée la suite D des dates relatives à chacun de ces événements.

Si X est l'alphabet d'entrée de la machine, S s'écrit:

$$S = x^*(1), x^*(2), \ldots x^*(i), \ldots$$

où $x^*(i)$ représente $x(i)$ ou $\overline{x(i)}$ selon la valeur atteinte par $x(i)$

$\forall i;\ x(i) \in X$.

Il est alors possible de créer une suite Sj relative à chaque grandeur, ainsi que la suite Dj des durées qui séparent deux événements pour cette jème grandeur.

- Créer la structure de données à partir de la suite ci-dessus.

Remarque

Nous envisageons surtout ici l'aspect géométrie de la trajectoire. En effet, la taille de l'échantillon nécessaire pour trouver cette organisation des P.S. est nettement réduite,par rapport à celle qui doit être exploitée,pour déterminer statistiquement les intervalles de confiance.

Par souci d'allégement d'écriture, nous abandonnons l'indice j relatif à la grandeur mesurée. Il n'y a aucune ambiguïté à ce sujet puisque la P.O. modélisée est en fait un ensemble de sous P.O. indépendantes.

Correspondance entre la séquence de C.R. et S

La correspondance entre la séquence de C.R. observée $r(1, n) \in R^+$ et la suite S est immédiate. Le nombre d'éléments de S est égal au nombre de changements d'éléments dans r(1,n). Il correspond au nombre d'arcs (qi, r, qj) tel que qj $\neq$ qi du chemin parcouru pendant la période d'observation.

Il est possible de décomposer S en sous suites s1, ... sn telles que:

$$S = s1, s2, s3, \ldots, sn$$

avec $s1, s3, \ldots, s2\,k+1, \ldots \in Sp$

et $s2, s4, \ldots, s2\,k, \ldots \in Sn$

où Sp représente l'ensemble des suites d'événements que l'on peut former à partir de séquences observées, correspondant à des déplacements de sens positif.

Sn regroupe celles qui se rapportent à des déplacements dans l'autre sens.

En cas de trajectoire ouverte, le nombre d'éléments de ces suites est borné supérieurement. Soit n cette borne; elle correspond à la longueur mesurée en nombre d'arcs, du plus grand chemin sans cycle de trajectoire. En cas de trajectoire fermée, cette valeur correspond à la périodicité du chemin correspondant à une évolution de sens constant.

Suites antisymétriques

Soit deux suites finies s^{-1} et s de n termes .

Notons s(i) (resp. $s^{-1}(i)$) le ième terme de la suite s (resp. s^{-1}).

Nous dirons que $(s)^{-1}$ est antisymétrique de s si pour tout $i \in [1,n]$, nous avons:

$$s^{-1}(i) = \overline{s(n-i+1)}$$

Exemple

a, b, c $\in$ X

s = a, $\bar{b}$, c et

$(s)^{-1}$ = $\bar{c}$, b, $\bar{a}$ sont antisymétriques

Suites englobées

La suite s1 de n termes est englobée dans la suite s2 de m termes si:

- $m \geqslant n$ et
- $\forall\, i \in [1,n]$; $s1(i) = \overline{s2(m-n+i)}$

Exemple

s1 = a, $\bar{b}$, b, a est englobée dans

s2 = a, $\bar{c}$, a, $\bar{b}$, b, a

Nous appelons trajectoire simple un ensemble ordonné de points singuliers sans bifurcation.

Proposition

Si S est la suite relative à un cycle dans une trajectoire simple, alors S se décompose en deux sous suites S = s1, s2 telles que

$$s1 = (s2)^{-1}$$

Ceci montre en particulier que si s1(n) est le dernier terme de s1, alors:

$$s1(n) = \overline{s2(1)}.$$

Cette propriété découle de l'hypothèse 2 admise au début de ce paragraphe. Elle peut s'exprimer également de la façon suivante:
si s1 est la suite relative à une évolution d'un P.S.i à un P.S.j d'une trajectoire simple, l'évolution de P.S.j à P.S.i entraîne l'observation d'une suite s2 telle que $s2 = (s1)^{-1}$.

Palindrome

La réciproque n'est malheureusement pas acceptable.

Soit une suite $s1 \in Sp$ telle que:

- $\exists$ s'1, s"1 $\in$ Sp; s1 = s'1, s"1

avec $(s"1)^{-1}$ englobée dans s'1

Dans ces conditions, il existe $s2 \in Sn$ telle que

$$s2 = (s"1)^{-1}.$$

Si la suite s'1, s2 correspond effectivement à un chemin contenant un cycle, s1, par hypothèse, correspond à un chemin de sens constant.

La suite s1 est révélatrice d'un palindrome.
Ce palindrome est dit borné s'il existe dans Sp (resp Sn), une suite s de longueur 2p telle que:

$$\forall i \in \mathbb{N} \; ; \; 0 < i < p \qquad s(p - i+1) = \overline{s(p+i)}$$

$$s(1) \neq \overline{s\ 2p}$$

Cett suite de 2(p-1) termes correspond à un chemin sans cycle de même longueur sur la trajectoire.

10.2.2.2 Base de l'inférence, propriétés de l'échantillon

Il est clair qu'entre l'automate canonique maximal |MIC-84|, que l'on peut construire directement à partir de l'échantillon observé (utilisé pour le test dans le cadre du cycle répétitif), et l'automate universel (correspondant au test statique), il existe un automate particulier et un seul, correspondant au modèle recherché. Il est donc nécessaire de définir des règles pour guider l'inférence.

Règles de base

a) L'inférence est conduite en évaluant la longueur (mesurée en nombre d'arcs (qi, r, qj)), du plus court chemin entre le P.S. occupé

à l'instant considéré et le P.S. d'entrée correspondant au début de la phase d'enregistrement de l'échantillon.

b) Cette distance, nulle au départ, est incrémentée pour chaque élément de la liste si le déplacement est positif; il est décrémenté si le déplacement est en sens inverse.

c) Soit d min et d max les distances extrêmes, rencontrées depuis le début de l'inférence, et d la valeur courante.
Si par application de la règle (b) la valeur de d sort de l'intervalle |d min, d max|, il est créé un nouveau sommet pour le modèle.

Sens d'évolution

L'inférence est guidée par la connaissance supposée du sens d'évolution.

Phase d'apprentissage

Si l'échantillon est obtenu pour un sens de déplacement constant à la suite, par exemple, d'une commande manuelle, la condition est, par avance satisfaite. Cette hypothèse d'auto apprentissage est industriellement tout à fait acceptable, elle ne doit pas être rejetée.

Voyons dans quelle mesure il est possible de détecter des changements de sens, à partir de l'échantillon, sans connaissance à priori, du sens réel d'évolution.

Echantillon complet

L'échantillon I sera dit complet,

- s'il a été généré par la P.O. saine,
- si tous les arcs ont été empruntés au moins une fois.

Cette définition est à rapprocher de celle donnée dans |MIC-84| qui stipule, qu'un échantillon I est structurellement complet pour une grammaire G = (X, V, P, S) où

X est l'alphabet terminal,
V est l'alphabet auxiliaire,
P est l'ensemble des règles de production,
S est un élément de V appelé axiome,

si

a) $I \subset L(G)$
b) l'alphabet sur lequel est écrit I est égal à X,
c) toutes les règles de P ont été utilisées au moins une fois.

Les points a et b sont ici remplacés par l'hypothèse de la P.O. saine, puisque tout élément r de l'alphabet terminal de C.R. (noté R), non utilisé dans I, conduit implicitement vers l'état final qf qu'il n'est donc pas nécessaire d'atteindre.

Si la séquence est telle qu'un aller retour complet n'a pas été effectué, la séquence sI peut être complète en faisant sI, $(sI)^{-1}$ si tous les arcs aller ou retour ont été parcourus.

Echantillon propre

Nous dirons que l'échantillon est propre, s'il est complet et s'il a été observé pour une évolution telle que le nombre de changement de sens, à l'intérieur d'un palindrome, est au plus, égal à l'unité. Nous notons sI la suite tirée de cet échantillon.

Détermination des changements de sens sur une trajectoire sans bifurcation

a) La présence dans sI de deux termes consécutifs de la forme $s(i) = x^*$, $s(i+1) = \overline{x^*}$ doit être considérée comme un changement potentiel de sens d'évolution.

b) Soit deux sous suites s1 et s2 de sI consécutives, délimitées par trois changements de sens potentiels, telles que:

l'une des suites s1, $(s2)^{-1}$ englobe l'autre.

Si l'échantillon est propre et s'il n'y a pas de bifurcation, il est possible de considérer que le passage du dernier élément de s1, au premier élément de s2, accompagne un changement de sens d'évolution. Cet événement est pris en compte comme tel dans le mécanisme d'inférence.

Si par contre, aucune des sous suites s1, $(s2)^{-1}$ n'englobe l'autre, le changement de sens potentiel n'est pas retenu comme changement de sens effectif.

En dehors des palindromes, la propriété (a) ne peut être satisfaite que par un changement réel de sens. Dans ce cas, la présence d'un cycle permet de justifier que l'une des suites s1, $(s2)^{-1}$ englobe l'autre.

Dans un palindrome, la propriété (a) peut être satisfaite au passage par le sommet pivot qui se traduit ici par le passage de la suite s1 à la suite $s2 = (s1)^{-1}$. Avec s1, s2 $\in$ Sp.

Comme l'échantillon est propre, il y a au plus un changement de sens dans le palindrome.

Soit un chemin qui traverse un palindrome de longueur 2p. Les sous suites s'1, s'2 relevées dans sI sont telles que

$$\text{long } (s'1) + \text{long } (s'2) \geqslant 2p$$

Dans ces conditions, aucune des sous suites s'1, $(s'2)^{-1}$ n'englobe l'autre. L'hypothèse du changement de sens est donc rejetée.

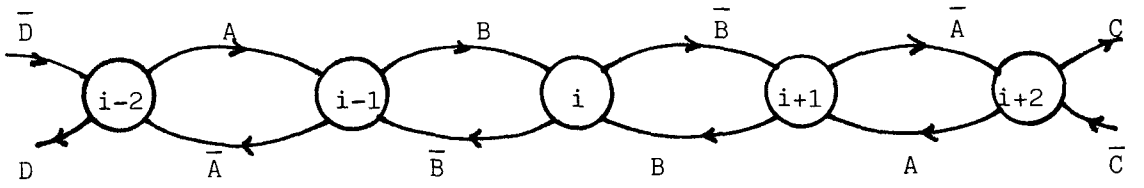
S'il y a un changement de sens, deux solutions sont possibles:

a) Cette inversion de sens est obtenue avant le passage par le sommet pivot. La présence du palindrome n'est pas ici révélée.

b) Si l'inversion de sens est effectuée après le passage par le pivot et avant la sortie du palindrome, il y a trois changements potentiels de sens. Il y a ici ambiguïté.
Notre hypothèse choisit de considérer ces trois événements comme réels.
Cette ambiguïté n'a pas d'influence sur le modèle.
En effet, l'échantillon I étant complet, il existe par ailleurs, au moins une traversée complète du palindrome qui en définit la structure.

Exemple

Soit à identifier le modèle de la figure 10.3, où qi représente le pivot du palindrome A B $\overline{B}$ $\overline{A}$.



- figure 10.3 -

Les différents cas sont les suivants:

- traversée: la suite est de la forme $\bar{D} \mid \bar{D}\ A\ B \mid \bar{B}\ \bar{A}\ C \mid \bar{C}$
  La séquence s1 = $\bar{D}\ A\ B$ et la séquence $(s2)^{-1} = \bar{C}\ A\ B$ ne s'englobent ni l'une, ni l'autre; l'hypothèse du changement de sens en qi est rejetée.

- Rebroussement avant qi; la suite relevée est $\bar{D} \mid \bar{D}\ A \mid \bar{A}\ D \mid D$.
  L'hypothèse du changement de sens est retenue. Elle correspond à la réalité. Le même résultat est obtenu avec $\bar{D}\ A\ B\ \bar{B}\ \bar{A}\ D$ pour laquelle le sommet pivot est atteint mais non dépassé.

- Rebroussement après passage par qi; une telle séquence est représentée par $\mid \bar{D}\ A\ B \mid \bar{B} \mid B \mid \bar{B}\ \bar{A}\ D \mid$

  Nous notons: s1 = $\bar{D}\ A\ B$
  s2 = $\bar{B}$
  s3 = B
  s4 = $\bar{B}\ \bar{A}\ D$

  Nous relevons: $(s2)^{-1}$ est englobée dans s1;
  $(s3)^{-1}$ est englobée dans s2;
  $(s4)^{-1}$ englobe s3.

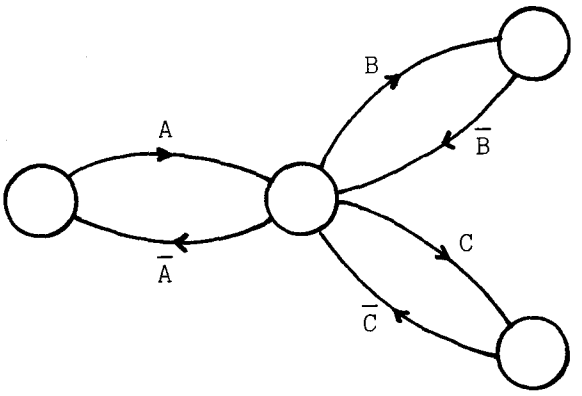
  Les trois hypothèses de changement de sens sont retenues. L'évolution réelle n'est peut être pas celle supposée, mais le modèle n'est pas erroné.

Si l'échantillon, bien que complet, n'est pas propre, mais que le nombre de changements de sens dans les palindromes est toujours impair, le modèle reste acceptable. Sinon, il y a détection possible du problème si la trajectoire ne contient pas de bifurcation.

Comportement en cas de bifurcation

Le modèle d'une telle trajectoire est donné figure 10.4 .

Si une partie de sI = $\bar{B}\ \bar{A}\ A\ C$, il est clair que cette sous suite présente les caractéristiques d'un palindrome. Si cette bifurcation n'a pas été révélée précédemment, le doute est levé par interrogation de l'opérateur.



- figure 10.4 -

).2.2.3 Algorithme de création du modèle

L'algorithme est donné ici pour une trajectoire sans bifurcation.

Le traitement est effectué par comparaison de deux suites:

- la suite notée sg (suite gauche) qui ne contient pas de doublons relatifs à un changement potentiel de sens,
- la suite de droite (notée sd).

Au départ de l'algorithme, nous faisons:

sd = sI; sg = vide

d = 0 (compteur de distance au point d'entrée)

d min = d max = 0 (valeurs extrêmes rencontrées de ce compteur)

Iv = + (indicateur de vitesse fixé arbitrairement à la valeur "sens positif")

P min et P max sont des pointeurs dans la structure de données.

a) Les premiers éléments de sd sont transférés dans sg jusqu'à ce que l'on rencontre deux fois de suite le même nom ou que sd soit vide.

b) lg est la longueur de sg. Il est effectué la comparaison pour i allant de 2 à lg entre sd(i)* et sg(lg - i+1)*.

S'il existe un rang i tel que $(sd(i)^* \neq \overline{sg\ (lg - i+1)^*})$

ET $\left[\overline{(sd(i-1)^* \neq sd(i)^*)}\ OU\ \overline{(sg(lg - i+1)^* \neq sg(lg-i)^*)}\right]$

alors, l'hypothèse du changement de sens n'est pas retenue, l'algorirhme est repris en (a).

c) sd est vidé dans la structure de données de la façon suivante.

Tant que sd n'est pas vide,
retirer le premier terme sg(1) de sg.
Si Iv = + alors d ⟵ d + 1

Si d > d max alors
- placer sg(1)* dans la structure de données à P max et actualiser le chainage avant.
- avancer P max, placer $\overline{sg(1)^*}$, créer le chaînage arrière.
- d max ⟵ d max + 1

Si Iv = - alors d ⟵ d - 1

Si d < d min alors
- placer $sg(1)^*$ dans la structure à P min, actualiser le chaînage arrière.
- avancer P min, placer $\overline{sg(1)^*}$, créer le chaînage avant.
- d min ⟵ d min - 1

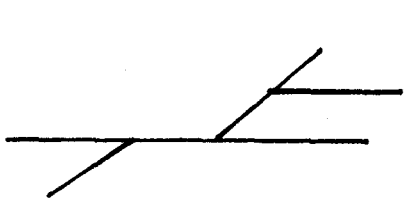
Si Iv = + faire Iv ⟵ (-), sinon Iv ⟵ (+).

Cet algorithme est repris en(a) tant que sg n'est pas vide.

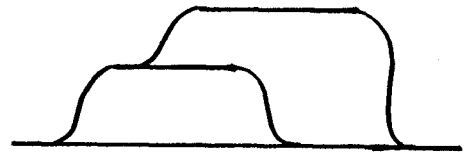
En cas de bifurcation, cet algorithme est modifié comme suit. L'hypothèse du changement de sens n'est pas retenue si l'opérateur déclare qu'il n'y a pas de bifurcation, sinon, une nouvelle branche est ouverte. En fait, cela revient à effectuer à chaque branche un compteur courant d avec sa valeur d max (d min est toujours nul) et son pointeur P max. Si l'on remarque qu'une trajectoire simple peut être considérée comme une trajectoire à deux branches, l'algorithme est généralisé à l'ensemble des trajectoires "étoilées".

La deuxième partie de(c) est alors modifiée comme suit:
si Iv = - alors d ⟵ d - 1
si d = 0 sortir de la branche.

En paramétrant chaque branche, nous avons un algorithme récursif de création du modèle.
Cet algorithme voit son domaine réduit aux trajectoires étoilées pour lesquelles chaque branche a un point unique d'entrée/sortie (figure 10.5).



trajectoire étoilée



trajectoire non étoilée

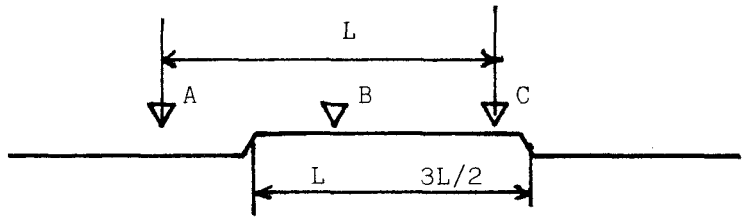
- figure 10.5 -

Les figures 10.6 et 10.7 illustrent le déroulement de l'algorithme sur une trajectoire simple sans palindrome pour la première, et avec palindrome de longueur 2 pour la seconde.

10.2.2.4 Initialisation

A partir de la valeur ro initiale du C.R. pris au début de la séquence d'observation, il est possible d'associer un C.R. à chaque sommet représenté dans la structure de données. Ceci permet de définir une table de correspondance Co: r $\in$ R $\longrightarrow$ Pq où Pq est le pointeur relatif au sommet q $\in$ Q dans la structure de données.

Pour tout élément r $\in$ R représenté plusieurs fois dans cette table de correspondance, il est associé le pointeur Pqo relatif au sommet initial qo.
Ce pointeur constitue le point d'entrée dans la structure après chaque initialisation.



- a - définition de la P.O.

Echantillon:

sI = A B C $\overline{C}$ $\overline{B}$ B C $\overline{A}$ $\overline{B}$ B A $\overline{A}$ A $\overline{C}$ $\overline{B}$ B C $\overline{A}$ $\overline{B}$ B

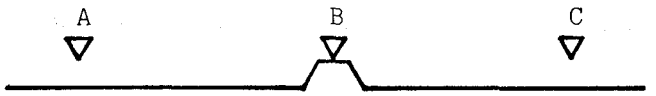
1ére passe V=+ dmax = 3 dmin=0	$(sg)^{-1}$= $\overline{C}$ $\overline{B}$ $\overline{A}$ d = 3 d = 1	changement de sens
2ème passe V=- dmax=3 dmin=0	$(sg)^{-1}$= B C d=1	chgt sens
3ème passe V=+ dmax=5 dmin=0	$(sg)^{-1}$=$\overline{B}$ $\overline{C}$ A B d=5	chgt sens
4ème passe V=- dmax=5 dmin=0	$(sg)^{-1}$=$\overline{A}$ B d=3	chgt sens
5ème passe V=+ dmax=5 dmin=0	$(sg)^{-1}$= A d=4	chgt sens
6ème passe V=- dmax=5 dmin=0	$(sg)^{-1}$= B C $\overline{A}$ d=1	chgt sens

7ème passe V=+ dmax=5 dmin=0	$(sg)^{-1} = B\ A\ \overline{C}\ \overline{B}$ d=5	chgt sens

	d	Vitesse > 0		Vitesse < 0	
passe 1	0	A	1		
	1	B	2	$\overline{A}$	0
	2	C	3	$\overline{B}$	1
passe 3	3	$\overline{A}$	4	$\overline{C}$	2
	4	$\overline{B}$	5	A	3
	5			B	4

- b - Structure de données

- figure 10.6 -



- a - définition de la P.O.

Echantillon:

sI= $\overline{B}$ C $\overline{C}$ B $\overline{B}$ A $\overline{A}$ B $\overline{B}$ A $\overline{A}$ B $\overline{B}$ C

1ére passe V=+ dmax=0 dmin=0	$(sg)^{-1}$= $\overline{C}$ B d=2	chgt sens
2ème passe V=- dmax=2 dmin=-2	$(sg)^{-1}$= B C $(sg)^{-1}$= $\overline{A}$ B $\overline{B}$ C d=-2	pas de chgt sens chgt sens
3ème passe V=+ dmax=2 dmin=-1	$(sg)^{-1}$= $\overline{B}$ A d=0	chgt sens
4ème passe V=-	$(sg)^{-1}$= $\overline{A}$ B d=-2	chgt sens
5ème passe V=+ dmax=2 dmin=-2	$(sg)^{-1}$= B $\overline{A}$ $(sg)^{-1}$= $\overline{C}$ B $\overline{B}$ A d=2	pas de chgt de sens chgt sens

	d	Vitesse < 0		Vitesse > 0	
1ére passe	-2	$\overline{C}$	-1		
	-1	B	0	$\overline{C}$	-2
2ème passe	0	$\overline{B}$	1	$\overline{B}$	-1
	1	C	2	B	0
	2			$\overline{C}$	1

- b - Struture de données

- figure 10.7 -

10.2.3 RECONSTITUTION DES COMMANDES

Les commandes sont des grandeurs non observables de l'automatisme, qu'il est donc nécessaire de reconstituer à partir de grandeurs observables. Deux catégories d'informations peuvent être exploitées à cette fin:

- l'état interne de la P.C. représenté par l'activation des étapes du grafcet par exemple,
- les ordres fournis par la P.C. complétés de ceux qui sont directement issus de l'environnement.

10.2.3.1 Utilisation de l'activation du grafcet

La définition du cahier des charges passe par des grafcets correspondant à la description fonctionnelle. A ce stade de l'étude, les actions définies sont en concordance avec la notion de commande qui nous préoccupe. La prise en compte de l'état de tout ou partie des étapes de ces grafcets, complétés éventuellement par le combinatoire local peut constituer une solution.
Néanmoins, les grafcets réellement implantés dans la P.C. sont généralement différents des grafcets de description fonctionnels.
En effet, une partie du traitement est généralement reportée hors de l'automate vers les préactionneurs, voire les actionneurs.
Par exemple, l'utilisation des sorties R.S, aujourd'hui admise dans la représentation grafcet, fait que l'état instantané du graphe de commande n'est plus suffisant pour connaître les actions actives, si la mémorisation est confiée aux préactionneurs.
Une solution envisageable est l'implantation dans le système de test, d'un grafcet fonctionnel dont les actions seraient les commandes appliquées aux actionneurs et non pas les ordres émis vers les préactionneurs par la P.C.. Dans ce cas, tout changement de programme de l'automatisme doit s'accompagner d'une redéfinition de ce graphe.

En résumé,

- si le grafcet de commande contient peu d'étapes ou s'il est possible de définir un nombre restreint de situations agissant sur chaque commande, et

- s'il existe une correspondance univoque entre les commandes et l'état du grafcet (limité éventuellement aux situations ci-dessus),

alors l'information sur l'état de ce graphe peut être utilisée comme indicateur de commande dans le modèle. Dans ce cas, l'auto apprentissage des commandes est immédiat, puisque l'on dispose d'une image observable de ces grandeurs.

?.3.2 Reconstitution des commandes à partir des ordres

Les ordres pris en considération sont à la fois les sorties de la P.C. et les informations reçues de l'environnement qui agissent sur l'évolution de l'automatisme sans passer par la P.C..
Ici aussi, il est possible de définir, pour chaque grandeur mesurée, le sous ensemble des ordres qui ont une influence directe sur son évolution.
S'il existe une correspondance univoque entre l'ensemble des ordres ainsi définis, et l'ensemble des commandes influençant la grandeur modélisée, la commande est introduite dans le modèle sous forme d'une expression combinatoire définie sur ces ordres. Dans l'hypothèse où la correspondance ordre, commande est biunivoque, l'auto apprentissage des commandes est immédiat.

En fait, ces hypothèses impliquent l'utilisation exclusive de préactionneurs et actionneurs monostables.
Si l'ensemble des préactionneurs fait appel à des dispositifs bistables et à plus forte raison à des assemblages de tels éléments, il devient très difficile de reconstituer les commandes par simple observation des ordres.
Une idée séduisante à priori, consiste à enregistrer, pendant la période d'apprentissage, une suite d'ordres O pendant l'observation de l'échantillon I qui a servi à élaborer les trajectoires.
Ayant repéré dans I les changements de sens de déplacement (en dehors éventuellement des palindromes), il est possible de classer les sous séquences d'ordres extraites de O, en fonction de leur effet sur l'évolution.

ypothèse

Toute séquence d'ordres, enregistrés entre deux changements de sens consécutifs sur la trajectoire contient:

- une mise en mouvement,
- la préparation du changement de sens,
- une mise à l'arrêt.

Une classification en deux temps de ces séquences peut être opérée. La première phase consiste à regrouper les séquences qui amènent à un déplacement positif d'une part, et celles qui provoquent une évolution dans l'autre sens d'autre part.

La deuxième phase vise à regrouper les séquences d'une même classe en fonction de la durée nécessaire pour passer d'un P.S. à un autre.

Nous n'avons pas approfondi cette étude pour le moment. Nous nous sommes orientés ici aussi vers un langage de description des préactionneurs.

10.3 DESCRIPTION LEXICOGRAPHIQUE DU MODELE

10.3.1 DESCRIPTION DES TRAJECTOIRES

Ce qui est envisagé ici, c'est la création automatique de la structure de données, à partir d'une description des trajectoires effectuées par un opérateur, dans la phase de conception de l'automatisme.
Cette procédure a fait l'objet de la publication d'un rapport dans le cadre d'un contrat (ADI-84) développé au laboratoire.
La solution retenue est celle de la description par un langage lexicographique. Ce choix est guidé par le souci d'éviter la limitation de taille, liée aux descriptions purement graphiques. Il est aussi justifié par le grand nombre d'informations à entrer en machine, qui rend vite fastidieuse la séquence de questions/réponses des systèmes purement interactifs. Un extrait du rapport cité ci-dessus joint en annexe, présente le langage proposé.
La création de la structure de données à partir du fichier de description est présentée dans le cadre d'une thèse de troisième cycle |HAC ..|.
Nous nous bornons donc ici à en tracer les grandes lignes.

La description d'une trajectoire se fait en deux temps.

Dans une première zone de déclaration,il est associé à chaque P.S., un mnémonique et un monome, définis sur l'ensemble des entrées relatives aux capteurs affectés par la grandeur mesurée concernée.

Dans la deuxième zone, la géométrie de la trajectoire est définie par la suite ordonnée des mnémoniques des P.S..

En cas de bifurcation de trajectoire, chaque sous trajectoire est décrite séparément. Des conditions de passage, d'une sous trajectoire à une autre, peuvent être indiquées.

Enfin, il est possible de déclarer une liste de capteurs (d'entrées) dont les valeurs peuvent être modifiées à partir d'autres grandeurs mesurées.
Ces deux derniers points permettent certains couplages entre les sous P.O. qui sont:

- la modification d'une trajectoire par une autre P.O.,
- le partage de capteurs.

.3.2 LANGAGE DE DESCRIPTION DES PREACTIONNEURS

L'ensemble des préactionneurs forme une machine séquentielle, obtenue par association en cascade de machines combinatoires ou séquentielles simples. L'idée retenue consiste à exploiter une bibliothèque de préactionneurs définis en fonction de leur comportement logique (monostable, bistable, ...). La règle d'association est basée sur une arborescence où chaque noeud représente un préactionneur et chaque arc un"circuit" de celui-ci. La possibilité de décrire l'ensemble des préactionneurs directement par un grafcet, est évidemment laissée à l'utilisateur.
De même, il est possible d'enrichir la bibliothèque par de nouveaux éléments. Le comportement de ces préactionneurs est décrit par un grafcet.
Comme pour les trajectoires, le document joint en annexe, présente et illustre par des exemples, la description des préactionneurs.

.3.3 INTERET D'UN LANGAGE DE DESCRIPTION DE LA P.O. DANS L'ETUDE DES AUTOMATISMES

Notre étude axée sur la sécurité et le test en ligne des automatismes, s'inscrit dans le cadre plus général de la conception assistée de ces systèmes industriels.

La définition d'un langage de description de la P.O., associé à son traducteur, fournit un outil puissant pour la mise au point et l'analyse prévisionnelle des performances des automatismes.
En effet, le modèle que nous avons créé peut devenir un outil de simulation par simple modification de gestion de la structure de données.
Cette gestion est alors la suivante:

- L'initialisation du modèle est choisi en définissant les valeurs des capteurs correspondants. S'il n'est pas possible de localiser l'ensemble des sous P.O., le système en informe l'opérateur.
- Soumis à une séquence d'ordres, le modèle des préactionneurs élabore la séquence de commandes. Pour chaque commande, il est recherché dans la structure de données, en fonction de l'état du modèle, l'arc normalement attendu et le temps moyen correspondant.
  Il est alors possible d'évoluer dans la structure de données en fonction de ce temps. Ceci est fait simultanément pour toutes les sous P.O..
- A chaque instant, il est créé une image des C.R.. Cette image est obtenue à partir des étiquettes relatives aux arcs tels que (qi, r, qi).

La seule difficulté réside ici dans la prédétermination des temps correspondant à chaque arc du modèle. Un couplage avec les systèmes de la C.A.O. mécaniques peut, dans certains cas, être un moyen d'évaluation commode.

Le modèle ainsi obtenu est un modèle de comportement hors contraintes de la séquence de travail (par opposition au grafcet dual du graphe de commande). Ce modèle est alors tout à fait pratique pour valider le grafcet de commande. Il est alors possible de faire tourner l'ensemble P.C./P.O. en simulation.

La vitesse peut être adaptée par changement du facteur d'échelle. La mise au point passe par la pause de points d'arrêt sur des situations du grafcet, des combinaisons d'ordres particulières ou même sur des états de la P.O..

A partir de ce modèle, il est alors possible d'extraire l'ensemble des situations actives du grafcet ainsi que leur durée moyenne dans le cycle de travail. A partir des calculs des fonctions dérivées proposées au chapitre III, il est alors possible d'évaluer de façon réaliste la susceptibilité de la commande aux pannes non consistantes sur chaque entrée.

De même, la simulation permet d'établir, pour chaque commande, la probabilité stationnaire d'être en un point singulier.

A partir de la structure de l'automate défaillant et de la connaissance de ces probabilités, il est possible de calculer de façon réaliste les taux de couverture Pcd, Pce et Pd. Cette procédure constitue un moyen d'évaluation prévisionnelle de l'efficacité du test. L'introduction de la défaillance dans le modèle peut être réalisée automatiquement. Les règles établies ramènent cette transformation à une simple manipulation de pointeurs.

De plus, la méthode peut être étendue aux pannes multiples.

Il est donc possible de faire une évaluation prévisionnelle des comportements en présence de pannes, à partir de la simulation de la P.O. saine. Ces résultats sont obtenus en un temps relativement court puisqu'il n'est pas nécessaire d'introduire les pannes en simulation.

Toutefois, ces mesures de nature probabiliste sont parfois insuffisantes. Pour les cas particulièrement critiques, il est possible également de faire tourner le modèle défaillant.
Il est très interessant de noter que la même structure de données puisse ainsi servir d'aide à la mise au point,d'évaluation de performance, de base au test en ligne.
L'apport important de ce modèle dans la phase d'étude, fait que le langage de description mis en oeuvre, revêt à nos yeux, une grande importance.

CONCLUSION DE LA TROISIEME PARTIE

La nécessité de quantifier les performances du mécanisme de test nous a amené à définir des taux de couverture et une probabilité de détection relatifs à une défaillance donnée.
Le taux de couverture de l'erreur Pce est particulièrement interessant pour l'exploitant puisqu'il permet d'évaluer le risque que l'erreur, résultant de la défaillance envisagée, ne soit pas révélée.
Nous avons également mis en évidence une stratégie permettant de localiser l'origine de la défaillance. Cette localisation conserve un caractère probabiliste, dans la mesure où différentes anomalies entraînent le même effet.
Les chiffres obtenus dans l'ensemble traité confirment l'intérêt du test dynamique pondéré.

Pour l'intégration matérielle du test, plus qu'un choix d'architecture, nous justifions de l'intérêt qu'il y a à éviter l'observation directe des ordres et comptes rendus échangés entre P.O. et P.C.. Cette proposition est faite à la lumière des taux de pannes des différents éléments mis en jeu, afin d'obtenir le meilleur compromis fiabilité / sécurité. Deux architectures sont plus particulièrement envisagées. La première utilise un processeur spécialisé pour la commande et un processeur affecté au test. Le mécanisme de détection est alors simplement ajouté à l'automatisme classique. Une architecture multiprocesseur, dans laquelle les processeurs se partagent indifferemment les tâches de commande et celles relatives au test, est également proposée. Ce deuxième dispositif d'une mise en oeuvre sophistiquée, se justifie lorsque la vitesse de traitement devient critique. Une telle architecture est pénalisante sur le plan de la fiabilité inhérente mais, la reconfiguration simple, en cas de défaillance d'une unité, permet de maintenir une fiabilité satisfaisante.

Dans le dernier chapitre, nous étudions la création de l'automate pondéré. L'hypothèse de la décomposition de la P.O. en sous P.O. relatives à chaque grandeur mesurée est ici adoptée.
La première méthode proposée, basée sur un auto apprentissage, nécessite un échantillon ayant des caractéristiques bien particulières.
La restitution de la commande, non observable, pose des problèmes pour lesquels les solutions sont simplement évoquées. Des études dans cette direction restent à faire.
Le deuxième moyen envisagé est l'utilisation d'un langage de description de la P.O.. L'automate pondéré est alors généré à partir d'un fichier créé au bureau d'étude. Cette dernière approche a notre préférence, car elle permet de créer un modèle de comportement de la P.O. sans que l'existence réelle de celle-ci ne soit effective.
Le modèle ainsi obtenu permet notamment une aide à la validation du logiciel de commande.

C O N C L U S I O N

Notre étude s'inscrit dans le cadre de la sûreté de fonctionnement des automatismes industriels considérés comme des systèmes réparables. L'accent a été mis sur la mise en place d'un mécanisme de détection des défaillances des éléments hors automate, basé sur une analyse syntaxique des comptes rendus. Les contraintes technologiques font que nos propositions s'appliquent essentiellement aux automatismes logiques à évolution séquentielle.
Le mécanisme de test proposé est une alternative à la duplication des éléments hors automate utilisés pour améliorer la sûreté des automatismes. Les taux de couverture calculés sur un exemple montrent les performances interessantes du test dynamique pondéré.
Nous avons introduit les bases d'un algorithme permettant de localiser la défaillance après détection de celle-ci. Cette proposition, après approfondissement, doit déboucher sur une assistance à la maintenance curative.
De même, le suivi permanent de l'évolution de la partie opérative permet de mettre en évidence des phénomènes de vieillissement qui se traduisent par une modification des vitesses d'évolution. Enfin, un historique des défaillances peut être constitué automatiquement.
Ces deux points relèvent de l'aide à la maintenance préventive.
Notre étude s'inscrit alors dans le cadre de la maintenance assistée par ordinateur qui conduit à une amélioration de la disponibilité des systèmes.

Nous avons vu que l'automate pondéré peut devenir un véritable modèle de la partie opérative et des organes hors automate qui lui sont associés.

L'utilisation du langage de description proposé permet de disposer de ce modèle dès la phase de conception de l'automatisme. Il devient alors possible de simuler l'ensemble partie commande - partie opérative, l'opérateur ayant seulement à créer les événements externes (arrivée de pièces par exemple) et de conduite.
La mise au point peut alors se faire par une technique de points d'arrêts sur certaines situations de la commande ou états de la partie opérative par exemple. Ce modèle constitue donc un outil d'aide à la validation des spécifications fonctionnelles et de leur transcription (validation des grafcets). De plus, il est possible d'introduire dans ce modèle des défaillances (en utilisant l'automate défaillant) ou des perturbations des temps d'évolution pour en mesurer les effets.
Enfin, l'utilisation de cette simulation permet de déterminer les différents estimateurs que nous avons proposés (susceptibilité aux pannes fugitives, taux de couverture de l'erreur) en tenant compte des évolutions imposées par la commande. Cette quantification des performances est utile pour le concepteur au moment du choix d'une architecture sûre de fonctionnement.

Il est certain que la sûreté d'un automatisme peut être compromise par des erreurs de conduite. Nous avons vu que dans certaines communications homme machine, le dispositif de test permet de révéler un manque de vigilance de l'opérateur. L'utilisation du modèle pendant la phase d'exploitation donne une image précise de l'état du système. Ceci doit pouvoir être exploité dans le cadre de l'aide à la décision, le conducteur effectuant alors un choix parmi différentes actions qui lui sont proposées par la machine.
Enfin, la simulation de l'automatisme complet peut être utilisée pour la formation des agents de conduite. Ici encore, la possibilité d'introduire dans le modèle des défaillances permet de placer l'opérateur dans des situations exceptionnelles nécessitant une réaction rapide et éclairée.
La modélisation de la partie opérative constitue donc une contribution importante à un vaste projet de conception et d'exploitation assistée par ordinateur des automatismes sûrs de fonctionnement.

BIBLIOGRAPHIE

ADE.80 GEMMA
Publication ADEPA

ADI.84 J.M. TOULOTTE, J. DEFRENNE, R.HACHEMANI. Contrat ADI n° 83/351 "Etude et réalisation d'un automate à sécurité intégrée".

AFC.77 AFCET. Groupe de travail Systèmes logiques. "Pour une représentation normalisée du cahier des charges d'un automatisme logique". Revue Automatique et Informatique Industrielle n°61-62. Novembre / Décembre 1977.

AFC.78 AFCET. Rapport de la commission systèmes logiques de l'AFCET. "Normalisation du cahier des charges d'un automatisme logique". Revue Automatisme tome XXIII n°3-4 Mars Avril 1978.

AFC.80 AFCET. Groupe de travail "sécurité et disponibilité des systèmes informatiques". "Sûreté de fonctionnement des systèmes informatiques".

AFC.82 AFCET. Document de synthèse du groupe de travail "systèmes logiques". "Les interprétations algébriques et algorithmiques du grafcet". Revue Automatisme Novembre 1982.

AFC.83 AFCET. Groupe de travail systèmes logiques. "Grafcet, interprétation algébrique et algorithmique et temporisations". Revue Automatisme Septembre 1983.

ALA.83 P. ALANCHE. "Localisation des défauts de l'installation. Défaut de non conformité de la structure d'un composant. DAST PA/CT/n° 274/83 du 12-12-83

AND.75 C. ANDRE. "Sur une méthode de conception assistée par ordinateur des systèmes logiques à évolutions simultanées". Thèse docteur 3ème cycle, Nice, juin 1975.

AND.76 C.ANDRE, F. BOERI, J. MARIN. "Synthèse et réalisation des systèmes logiques à évolutions simultanées". Revue française d'Automatique, Informatique et Recherche Opérationnelle (RAIRO) Vol 10 n°4, avril 1976.

AUT.82 "Problème de sûreté: le contrôbloc Revue Le Nouvel Automatisme, mai 1982.

AVI.75 AVIZIENIS. Fault Tolerance and Faults Intolerance Complementary. Approach to Reliable Computing, International Conference on reliability. Los Angeles Avril 1975.

BAC.80 J.P. BACONNET, B. GIRARD, S. NATKIN. "Introduction à la sûreté de fonctionnement des systèmes informatiques". Monographie AFCET 1980.

BEO.77 C. BEOUNES. "Automate sûr et modulaire adapté aux régulations avioniques". Thèse Docteur Ingénieur. Toulouse 1977

BER.74 J.C. BERTRAND. "Sur la détection automatique des pannes dans les systèmes logiques". Thèse d'état Montpellier 1974.

BLA.80 M. BLANCHARD. "Comprendre, maîtriser et appliquer le Grafcet". Editions CEPADUES. 1980

BOS.78 J.C.BOSSY, P.BRARD, P. FAUGERE, C. MERLAUD. "Le Grafcet, sa pratique et ses applications". Educalivre

BOU.83 E. BOUDOM, P. GUILLOIS. "Méthode d'aide à la conception d'automatismes par ordinateur. Congrès AFCET Automatique 83, Besançon, 15-17 novembre 1983.

BOU.80 P. BOURNAI. "Système C.A.O. d'analyse et de simulation de systèmes parallèles connectés à des transmittances". Groupe de travail AFCET "PARDI" Paris, mars 1980.

BOU.78 J. BOUSSIN. "Synthesis and analysis of logic automation systems". IFAC Congress 1978.

BRA.83 G.W. BRAMS. "Réseaux de Pétri, thèorie et pratique". Edition Masson 1983.

CAR.83 B. CARRIERE, C. CAZALOT, J.M. DUMAS, P.M. GROSJEAN, P. LEROY, F. PRUNET. "Un système de C.A.O. pour la commande de processus reposant sur un standard". IFIP 1983.

CHA.68 CHAPOUILLE, R. DE PAZZIS. "Fiabilité des systèmes". Masson 1968.

CHE.81 David M. CHERBA. "Programming PCs to detect faults in machines or processes". Control Engineering, février 1981.

CHI.67 J. CHINAL. "Techniques booléennes et calculateurs arithmétiques" Dunod 1967.

CLA.84 P. CLAUDIN. "Méthode d'évaluation de la sécurité de l'automate programmable". Colloque Automatique appliquée SEE, Nice 1984.

COR.74 M. CORAZZA. "Contribution à l'étude de la fiabilité des systèmes". Thèse de Doctorat 1974, Toulouse.

COR.75 M. CORAZZA. "Techniques mathématiques de la fiabilité prévisionnelle" Edition Sup Aero 1975.

COR.83 D. CORBEEL, J.C. GENTINA, C. VERCAUTER. "Généralisation des réseaux de Pétri".
IASTED Symposium on Applied Informatics, Lille 1983.

COS.76 A. COSTES, J.C. LAPRIE, LESTRADE, A. CARBONNEL. "Prévision de la sûreté de fonctionnement des systèmes par les processus stockastiques. Application à une structure redondante dynamique".
Annales du 3ème congrès de fiabilité, Perros-Guirec, 1976.

COS.79 A. COSTES, J.E. DOUCET, C. LANGRAULT, J.C. LAPRIE. "SURF: système d'évaluation de la sûreté de fonctionnement, 1ère partie méthode". Note technique LAAS 79 T 37, août 1979.

COS.80 A. COSTES, J.E. DOUCET, C. LANDRAULT, J.C. LAPRIE. "SURF: un programme d'aide à la conception de systèmes sûrs de fonctionnement".
2ème colloque international Fiabilité et Maintenabilité, Perros-Guirec, septembre 1980.

COS.80 A. COSTES, C. LANDRAULT, J.C. LAPRIE. "Sûreté de fonctionnement des systèmes informatiques".
Monographie AFCET, 1980.

COS.81 A. COSTES, J.E. DOUCET, C. LANDRAULT, J.C. LAPRIE. "SURF: a program for dependability evaluation of complex systems".
11th Symposium on Fault Tolerant Computing. (FTCS-11) Portland Maine, juin 1981.

COU.71 M. COURVOISIER, M. DIAZ. "Programme d'analyse booléenne des systèmes séquentiels".
revue Automatisme, décembre 1971.

COU.72 M. COURVOISIER. "Elaboration de tests de détection de pannes dans les systèmes séquentiels".
Automatisme, octobre 1972.

COU.72 M. COURVOISIER. " Localisation des pannes simples d'une machine séquentielle à partir des séquences de synchronisation".
Automatisme, octobre 1972.

COU.80 M. COURVOISIER, R. VALETTE. "Systèmes de commande en temps réel".
Edition SCM 1980, distribution Librairie Lavoisier.

COU.83 M. COURVOISIER, R. VALETTE, J.M. BIGOU, P. ESTEBAN. "Réseaux d'automates et ateliers flexibles";
Congrès automatique AFCET Productique et robotique intelligente Besançon, novembre 1983.

DAC.76 E. DACLIN, M. BLANCHARD. "Synthèse des systèmes logiques";
Ed. CEPADUES, collection Sup'Aero, 1976.

DAL.83 Y. DALLERY, H. DENEUX, R. DAVID. "Recherche d'une même base de description en vue de la simulation et de la commande d'un atelier flexible, utilisation du grafcet".
Congrès automatique AFCET, Besançon novembre 1983.

DAV.74 R. DAVID. "Testabilité des systèmes logiques par des séquences d'entrée aléatoires". Journée prévention des pannes dans les systèmes logiques, IRIA; avril 1974.

DEF.79 J. DEFRENNE. "Implantation de réseaux de Pétri sur automate microprocesseur à haute sûreté de fonctionnement". Thèse "ème cycle automatique, Lille 1, 1979.

DEF.81 J. DEFRENNE, J.M. TOULOTTE. "Diagnostic en ligne des capteurs et actionneurs". Journée AFCET, mars 1981.

DEF.83 J. DEFRENNE, J.M. TOULOTTE. " Sur l'accroissement de la sécurité des systèmes à commande logique". 4ème journée scientifique et technique de la production automatisée juin 1983.

DEF.84 J. DEFRENNE. " Amélioration de la sécurité et de la maintenabilité des machines séquentielles". 4ème colloque Fiabilité Maintenabilité, Perros-Guirec, mai 1984.

DEF.84 J. DEFRENNE. "Détection de défauts dans un système séquentiel par exploitation de séquences enregistrées". Colloque SEE Automatique Appliquée, Nice mai 1984.

DEI.83 DEI-SVALDI, M. COLLIER, J.P. VAUTRIN. "Machine commandée par automate programmable: amélioration de la sécurité". le Nouvel Automatisme mai 1983.

DIA.74 DIAZ. "Systèmes totalement auto-testables". Journée Prévention des pannes dans les systèmes logiques. IRIA avril 1974.

DIA.74 M. DIAZ. "Conception de systèmes totalement autotestables et à pannes non dangereuses". Thèse Doctorat, Toulouse, 1974.

DOU.81 J.E. DOUCET. "SURF: système d'évaluation de la sûreté de fonctionnement, 2ème partie: notice d'utilisation". Note technique, LAAS 81 T 52, décembre 1981.

DUB.80 B. DUBUISSON, P.LAVISON. "Surveillance of an Nuclear reactor by use of a Pattern Recognition methodologie". 13e on system man and cybernetic vol.SMC-10, n°10, october 1980.

DUR.74 C. DURANTE, J.C.BERTRAND, J.P. MARTIN. "Sur le problème de la maintenance automatique des systèmes informatiques". Journées d'études sur les calculateurs embarqués. LAAS Toulouse, juin 1974.

DUR.74 C. DURANTE, J.C. BERTRAND. "Le point sur le test automatique des circuits logiques par la méthode de l'inverse". Journée d'étude prévention des pannes dans les systèmes logiques, IRIA avril 1974.

ELL.82 Robert K. ELLIS. "PC diagnostics adapt to limited process changes". Control Engineering, septembre 1982.

ESQ.78 P. ESQUISSAUD, G.GUESNIER. "Systèmes logiques de commande dans les centrales électriques". Journées AFCET/SEE février 1978.

FU.74 K.S. FU. "Syntactic methods in pattern recognition". Mathematics in science and engineering, vol 112 Academic Press, 1974.

FU.75 K.S.FU, T.L. BOOTH. "Grammatical Inference: Introduction and Survey". IEEE Trans.on SMC, 1975.

GIA.74 N. GIAMBIASI. "Contribution au test en ligne des circuits séquentiels". Thèse de 3ème cycle, Montpellier 1974.

HAC.84 R. HACHEMANI. "Simulation et diagnostic de la partie opérative". Rapport de DEA, USTL Lille 1984.

INA.82 M. F. Mc INALLY."Machine diagnostics improved with Programmable Controllers". Control Engineering, juillet 1982.

ITI.82 P. ITISCOHN. "Sécurité et automates programmables". Electronique Industrielle n°35 6/1982.

KAR.78 M.F. KARAVAY, ES. SOGOMONYAN. "Compared reliability analysis of redundant systems". FTCS-8, Toulouse juin 1978.

KAU.75 KAUFMAN, GROUCHKO, CRUON. "Modèles mathématiques pour l'étude de la fiabilité des systèmes". Masson.

KER.76 KERAN GUEYEN. "Etude sur la disponibilité des sytèmes informatiques". IRISA Publication interne n°46, juillet 1976.

KUB.78 C. KUBIAK, L. ETESSE, A. PRINGENT, J.L. RAINARD. "La sûreté de fonctionnement dans les systèmes complexes". Note technique NT/RCI/PLC Juin 1978.

LAN.77 C. LANDRAULT. "Prévision de la sûreté de fonctionnement des systèmes numériques réparables". Thèse Docteur ès Sciences INP Toulouse, 1977.

LAP.75 J.C. LAPRIE. "Prévision de la sûreté de fonctionnement et architecture de structures numériques temps réel réparables". Thèse de Docteur ès Sciences, Univ. Paul Sabatier Toulouse, 1975.

LIE.76 LIEVENS. "Sécurité des systèmes". Edition CEPADUES, 1976.

MAR.75 J. MARIN. "Sur le test en ligne des machines séquentielles réalisées à partir des réseaux de Petri". Thèse de Docteur 3ème cycle, Nice, Décembre 1975.

MAZ.78 G. MAZARE. "Structure multimicroprocesseurs. Problémes de parallélisme. Définition d'un système particulier". Thèse de Docteur ès Sciences, INP de Grenoble, juin 1978.

MER.74 J.J. MERCIER. "Contribution au test en ligne des circuits combinatoires". Thèse de Docteur de 3ème cycle, Montpellier 1974.

MIC.79 C. MICHEL "Ensemble d'outils pour la conception assistée par ordinateur de systèmes numériques à haute performance". 7ème colloque traitement du signal et ses applications, Nice 79.

MIC.84 L. MICLET. "Méthodes structurelles pour la reconnaissance des formes". Eyrolles.

MOA.79 M. MOALLA, G. SAUCIER, J. SIFAKIS, M. ZACHARIADES. "A design tool for the multilevel description and simulation of interconnected modules". 3rd Symposium on Computer Architecture, Tampa 1979.

MOA.80 M. MOALLA, J. SIFAKIS, M. SILVA. "A la recherche d'une méthodologie de conception sûre des automatismes logiques basés sur l'utilisation des Réseaux de Petri". Sûreté de fonctionnement des sytèmes informatiques, monographies AFCET 1980.

MOA.81 M. MOALLA, R. DAVID. "Extension du GRAFCET pour la représentation de systèmes temps réel complexes". RAIRO Automatique vol 15 n°2, 1981, ou Journée AFCET Validation et spécificatiion du GRAFCET mars 1981.

NAI.83 A.NAIFI. "Contribution à l'étude des systèmes logiques, fonctions combinatoires et théorie de l'information. Sécurité et simulation de la partie opérative". Thèse de Docteur de 3ème cycle, UST Lille 1983.

OSA.70 OSAKI. "System reliability analysis by Markov reneval processes". Journal of Opérations Research Society of Japan, vol 12, n°4, may 1970.

OSS.80 B.E. OSSFELD, I. JONSON. "Recovy and diagnostics in the central control of the axe swiching systems". IEEE Trans. on computer vol. C-29, n°6, june 1980.

PAR.78 J.P. PARSY. "Méthode de décomposition des réseaux de Pétri interprétés en vue de leur réalisation". DEA Automatique, UST Lille, juin 1978.

PAR.80 J.P. PARSY. "Dispositif interactif d'aide à la conception des programmes de commande de processus logiques industriels". Thèse de Docteur de 3ème cycle, UST Lille 1980.

PRA.79 B. PRADIN. "Unoutil graphique interactif pour la vérification des systèmes à évolutions parallèles décrits par réseaux de Pétri". Thèse de Docteur Ingénieur, Univ. Paul Sabatier, Toulouse 1979.

PRU.74 F. PRUNET, C. DURANTE, C. CHICOIX. "Test hiérarchisé des ensembles complexes". Journée Prévention des pannes dans les systèmes logiques, IRIA avril 1984.

PTA.84 Projet PTA, document de travail du groupe. "Représentation interne du GRAFCET". ADEPA Août 1984.

RAU.84 A.RAULT, D. JAUME, M. VERGE. "Sûreté de fonctionnement des processus industriels". Congrés SEE Automatique appliqué, Nice 1984.

SAV.78 J. SAVIR. "Testing for mulitiple intermittent failures in combinational circuit by a maximizing the probability of fault detection. FTCS-8, Toulouse juin 1978.

SAV.80 J. SAVIR. "Detection of single intermittent faults in sequentials circuits". IEEE Trans. on computer vol. C-29 n°7 july 1980.

SCH.69 M. SCHWOB, G.PEYRACHE. "Traité de fiabilité". Masson.

SIE SIEMENS. Notice d'utilisation des automates programmables.

SIF.78 J. SIFAKIS. "Realisation of fault tolerant system by coding Petri nets. FTCS-8, TOulouse juin 1978.

SIF.77 J. SIFAKIS. "Homomorphisms of Petri nets: application to the realisation of fault tolerant systems". RR90 octobre 1977.

SUA.78 M. SILVA SUAREZ. "Contribution à la synthèse programmée des automatismes logiques". Thèse de Docteur Ingénieur, INPG Grenoble 1978.

SURF.81 Projet Pilote SURF. Bilan et perspectives 81. Agence de l'informatique.

THE.78 S. THELLIEZ. "Pratique séquentielle et réseaux de Pétri". Eyrolles, coll. EEA 1978.

THE.81 S. THELLIEZ, J.M. TOULOTTE. "Grafcet et logique industrielle programmée". Eyrolles 1981.

TOH.71 Y. TOHMA, Y. OHYAMA, R. SAKAY. "Realisation ef fail safe sequential machines by using a K out of n code". IEEE Trans. on computer vol. C-20 n°11, novembre 1971.

TOU.78 J.M. TOULOTTE. "Réseaux de Pétri et automates programmables". revue Automatisme, tome XXIII n°6-7 juillet/ août 1978.

TOU.83 J.M. TOULOTTE, J. DEFRENNE. "Sur la sécurité des sytèmes à commande logique". IASTED Symposium Applied Informatics, Lille mars 1983.

VAL.75 R. VALETTE, J.C. GEFFROY, M.COURVOISIER. "Modèlisation des systèmes de commande numérique et leur analyse". Congrès AFCET, Toulouse novembre 1975.

VAL.76 R. VALETTE. "Sur la description, l'analyse et la validation des systèmes de commande parallèles". Thèse de Docteur ès Sciences, Univ. Paul Sabatier Toulouse novembre 1976.

VAL.78 VALK. "Self modifing nets, a natural extension of Petri nets". ICALP 1978. Notes in computer Sc n°62, Springer Berlin 1978.

WAG.74 R.A. WAGNER, M.J. FISHER. "The string to string correction problem". JACM Vol. 21 n°1 1974.

ZAH.80 J. ZAHND. "Machines sequentielles". Traité d'électricité volume XI. Edition Georgi.

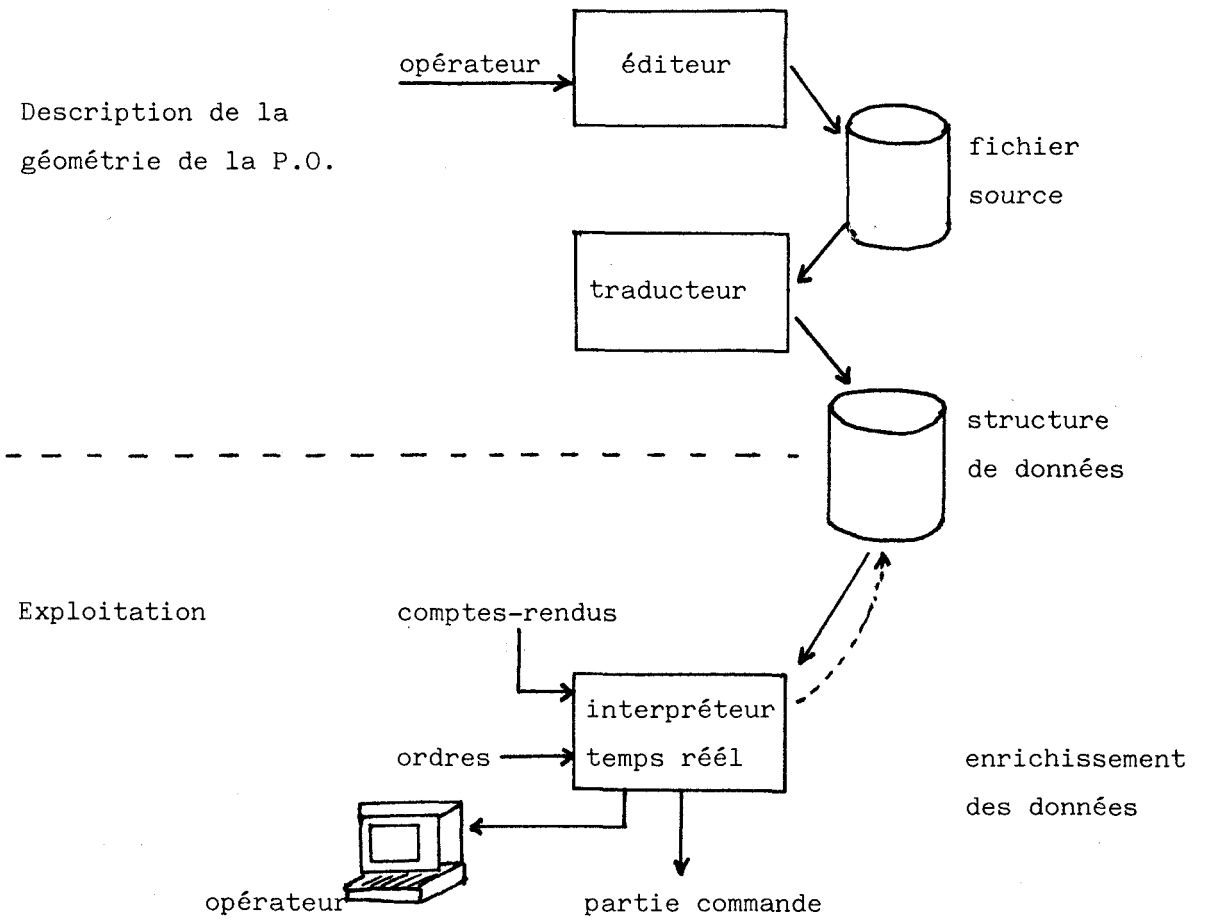
NF.82 "Diagramme fonctionnel "GRAFCET" pour la description des systèmes logiques de commande".
Norme NF C03-190 juin 1982.

A N N E X E

Cette annexe est extraite du contrat de recherche n°83/351 passé entre l'ADI et le CREATI. Elle présente un langage de description du cahier des charges mis au point conjointement par les différents intervenants sur ce contrat.

B - DESCRIPTION DE LA PARTIE OPERATIVE

L'utilisateur dispose d'un ensemble de logiciels permettant de transformer un microcalculateur en dispositif de surveillance. Ces logiciels sont implantés sur MICRAL 9050 sous CPM 86. Ils ont été développés en FORTH ce qui leur procure une relative transportabilité. La procédure de mise en oeuvre est la suivante (Figure 2-1)



- FIGURE 2-1 -

Le rôle de l'opérateur dans la phase initiale est de décrire la P.O. c'est à dire :

- spécifier le cablâge des préactionneurs :
- déclarer les trajectoires.

Pour cette opération il dispose d'un langage de description permettant de créer un fichier source constitué d'une suite de déclarations. Le fichier est créé à partir de l'éditeur de la machine. Un traducteur génère alors un ensemble de données qui seront exploitées en temps réel par un interpréteur. Nous avons adopté à ce sujet la démarche suivante.

Il a été choisi de représenter tout grafcet par une structure interne de données. L'interpréteur de cette structure gère donc en temps réel le grafcet correspondant.

La même structure de données a été utilisée pour la modélisation de la P.O. C'est donc le même interpréteur qui assure la surveillance. Cette gestion est optimalisée en temps d'exécution en limitant le volume de traitement en fonction de l'activation du grafcet géré à l'instant considéré.

La détermination des intervalles de temps associés aux arcs en fonction des commandes, est effectuée, en cours d'exploitation, par le logiciel spécifique de surveillance. Nous avons donc un enrichissement permanent du modèle. De plus un historique est constitué. Un interface permet à l'opérateur de ressortir des éléments de l'historique pour un suivi de la P.O.

Dans ce chapitre nous étudions le langage de description.

I- Règles générales

Le retour chariot n'est pas le séparateur d'instructions. De même un ou plusieurs espaces conduisent au même résultat. La mise en page proposée dans les exemples est donc donnée à titre indicatif. Tout texte entre parenthèse est considéré comme un commentaire.
La description de la P.O. est comprise entre le mot **DEBUT** et le mot **FIN**.
La rencontre du mot fin déclence la traduction et l'élaboration de la structure de donnée.

I-1 Identificateurs

Toutes les variables sont représentées par un identificateur. Il contient au maximum 31 caractères alphanumériques : le premier caractère est toujours une lettre.

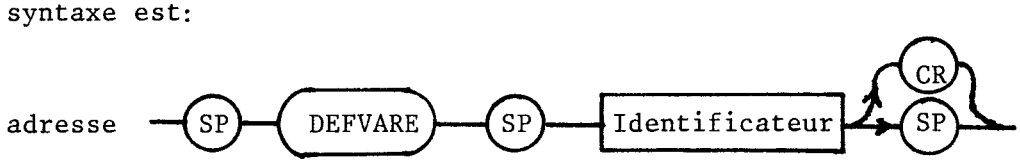
Le même identificateur ne peut évidemment pas être affecté à plusieurs variables (les identificateurs sont globaux).
De même les mots réservés ne doivent pas être utilisés. Pour être utilisé, un identificateur doit être déclaré. Ceci impose donc des contraintes dans l'ordre des instructions.
Toutefois les déclarations ne sont pas obligatoirement regroupées au début du fichier.

Déclaration d'un variable d'entrée

Chaque variable a une adresse logique représentée par un nombre décimal. La valeur de ce nombre dépend du couplage avec la partie commande (interface adresse logique, adresse physique).

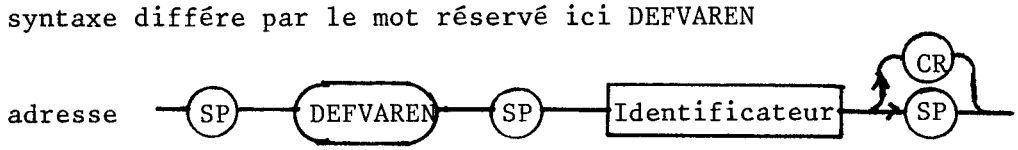
Variable binaire

la syntaxe est:



Variable décimale

La syntaxe diffère par le mot réservé ici DEFVAREN



Il est également possible de définir des variables internes par DEFVARI et DEFVARIN et des variables de sorties par DEFVARS et DEFVARSN mais ceci n'est pas utile pour la modélisation.

I-2 Expressions booléennes

Les signes d'opérations sont :

+ pour le OU (le ET est implicite)

| pour la négation (le signe précède l'identificateur de la variable niée)

Chaque signe d'opération est séparé par un espace, dans la version actuelle il n'y a pas de parenthèses explicites.

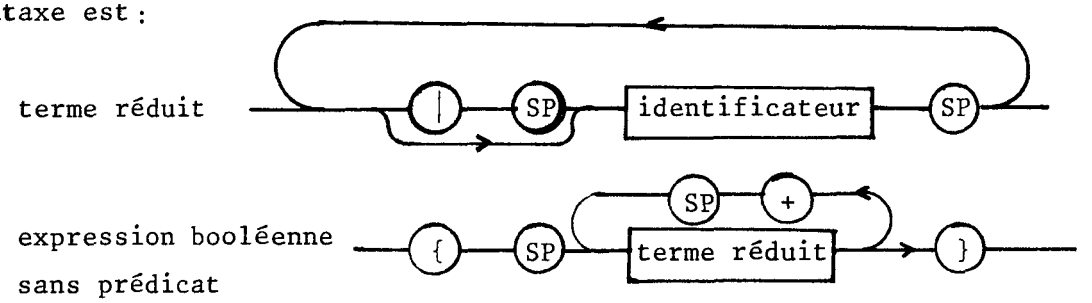
Une expression booléenne est encadrée des séparateurs { et }.

EXEMPLE :

F = a + $\overline{b}$c + $\overline{d}$e

```
 1  DEFVARE a
 4  DEFVARE b
 7  DEFVARE c
 2  DEFVARE d
15  DEFVARE e
{ a + | b c + | d e }
```

La syntaxe est :



Il est possible d'introduire des prédicats portant sur des variables numériques. Un prédicat permet de comparer une variable numérique à une constante ou deux variables numériques entre elles. Les signes de comparaisons sont :

= égal

< inférieur

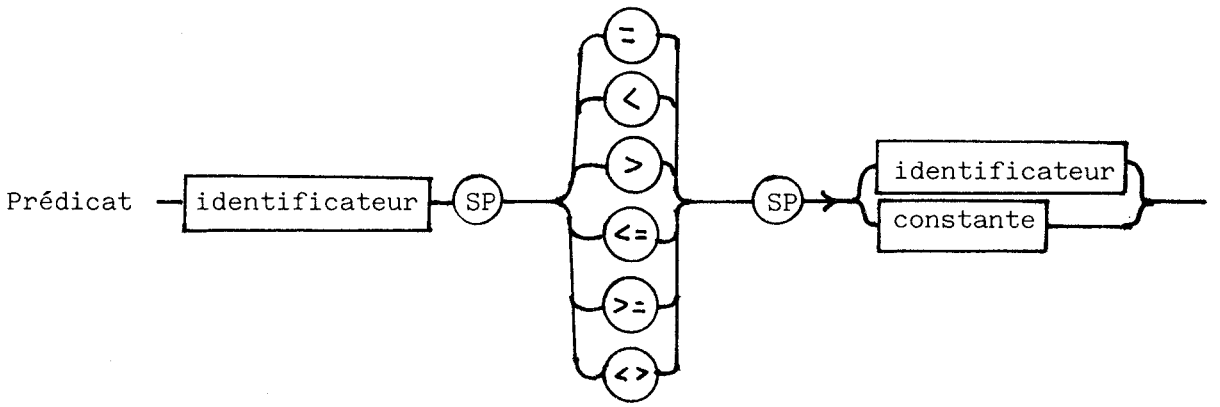
> supérieur

<= inférieur ou égal

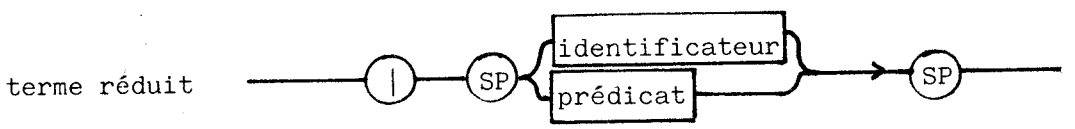
>= supérieur ou égal

<> différent

Les variables numériques et les constantes sont des entiers compris entre + 0 à + 65536 (16 bits).



La syntaxe d'un terme réduit devient alors terme réduit.



EXEMPLE : Les variables a b c étant supposées déclarées précédemment,

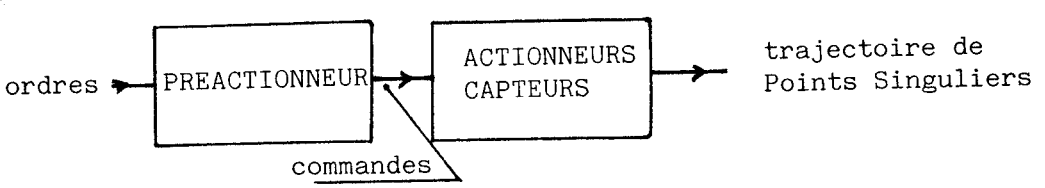
La fonction $F = a\bar{b}c + (CPT = 20)\bar{c}$ s'écrit :

```
10 DEFVAREN CPT
{ a | b c + CPT = 20 | c }
```

I-3 Décomposition de la P.O.

La P.O. est décomposée en grandeurs mesurées. L'ensemble des matériels qui agissent sur une telle grandeur forme l'actionneur.

L'ensemble des matériels qui élaborent les commandes appliquées à l'actionneur à partir des ordres, forment les préactionneurs.



Les capteurs placés sur une trajectoire permettent de décomposer cette trajectoire en points singuliers. Pour chaque grandeur mesurée il y a deux ensembles à décrire qui sont :

- Les préactionneurs ;
- l'actionneur.

Le description des préactionneurs conduit à la déclaration des commandes (appelées vitesses).
La description de l'actionneur permet de définir la géométrie de la trajectoire. Elle relie le sens d'évolution aux vitesses déclarées dans la partie préactionneurs.
La description des préactionneurs doit donc toujours précéder celle de l'actionneur correspondant. Cette contrainte peut être satisfaite en déclarant l'ensemble des préactionneurs puis l'ensemble des actionneurs. Elle peut également être satisfaite de bien d'autres façons et notamment en déclarant successivement, pour chaque grandeur mesurée, le bloc préactionneurs suivi de la partie actionneur. Cette façon de procéder est généralement plus efficace.

II - Déclaration des préactionneurs

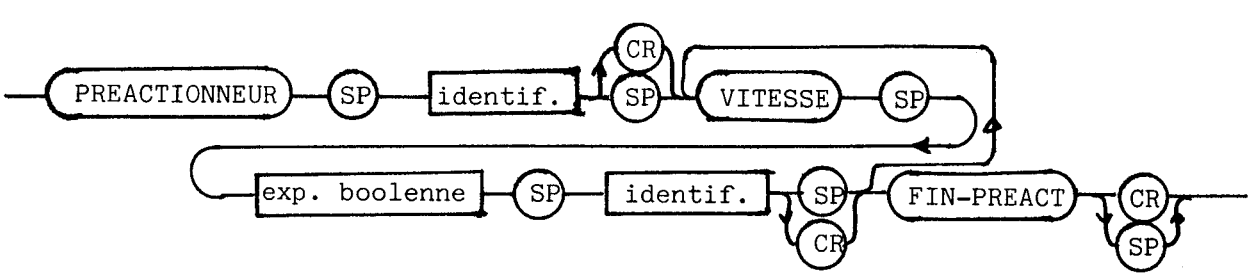
Chaque sous ensemble de préactionneurs associé à un actionneur est décrit entre les mots **PREACTIONNEURS** et **FIN-PREACT**. Deux cas sont envisagés.

II- Préactionneurs purement combinatoire

Dans certains cas particulièrement simples, la connaissance des ordres suffit à déterminer les vitesses. Les préactionneurs établissent dans ce cas une simple correspondance entre les ordres et les vitesses.
La déclaration prend la forme suivante :

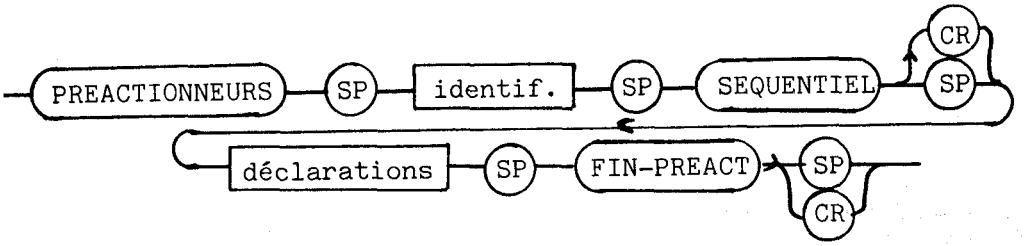
```
PREACTIONNEURS    identificateur
   VITESSE        expression booléenne        identificateur
     :
     :
     :
   VITESSE        expression booléenne        identificateur
FIN-PREACT
```

La mise en page proposée n'est pas restrictive, la syntaxe est en fait la suivante :



II-2 Préactionneur séquentiel

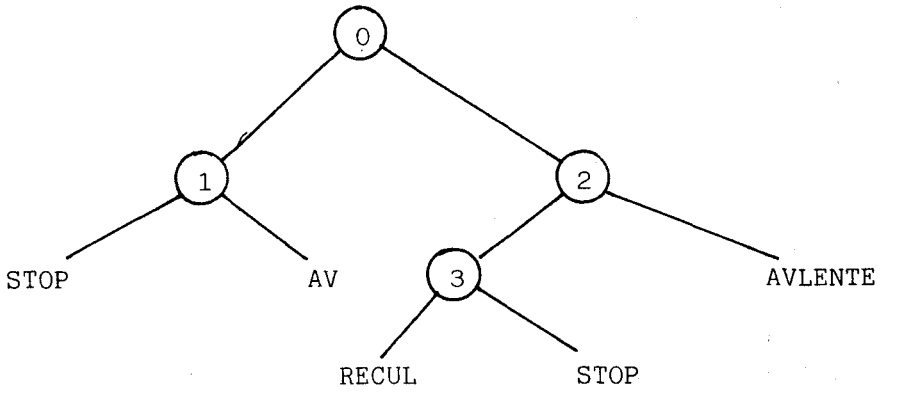
La syntaxe globale est :



Nous avons admis que le circuit de puissance commuté par les préactionneurs peut se modéliser par un arbre. Chaque préactionneur est représenté par un arbre élémentaire ayant une racine, n noeuds terminaux représentant les n positions de l'élément, pas de noeuds intermédiaires.

Pour l'ensemble des préactionneurs relatif à un même actionneur :

- les noeuds sont numérotés 0, 1, ..., 9, 10, 11,... en commençant par la racine (figure2-2 ;
- Les commandes (vitesses) sont affectées aux extrémités.

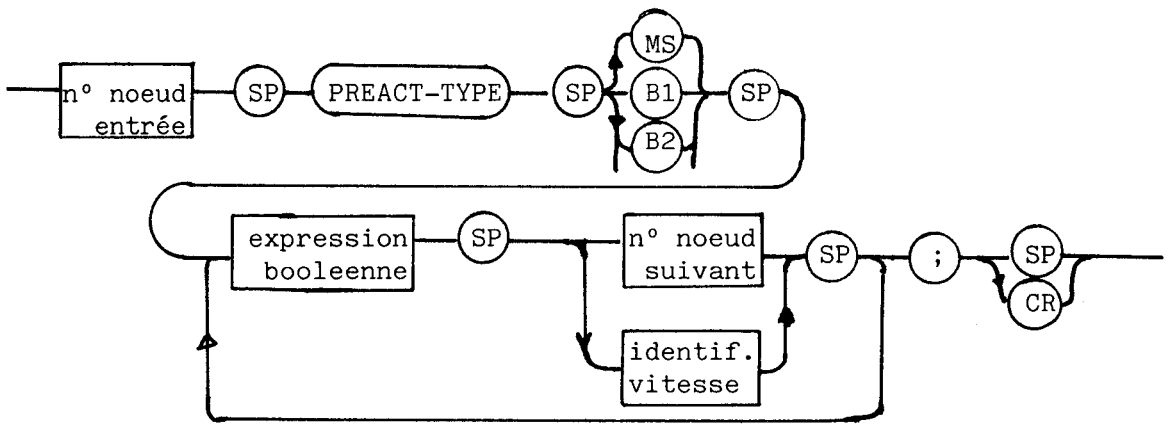


-FIGURE 2-2 -

Chaque préactionneur est alors déclaré. Les déclarations portent sur :

- le type ;
- les fonctions de pilotage ;
- l'affectation des positions.

La syntaxe est la suivante :



Cas d'un préactionneur monostable, la dernière expression booléenne est supprimée. C'est implicitement l'absence de toutes les autres.

Les types (MS, B1, B2,...) définissent l'évolution de la position du préactionneur en fonction des ordres appliqués. Ils font référence à une bibliothèque de préactionneurs qui peut être complétée par l'utilisateur.

Les expressions booléennes représentent les fonctions de pilotage. Le numéro du noeud de sortie, ou la vitesse qui suit cette expression, correspond à l'affectation de la position obtenue pour le pilotage considéré lorqu'il est le seul appliqué.
L'ordre de déclaration des pilotages peut ne pas être indifférent, il dépend de la description utilisée pour la constitution de la bibliothèque.

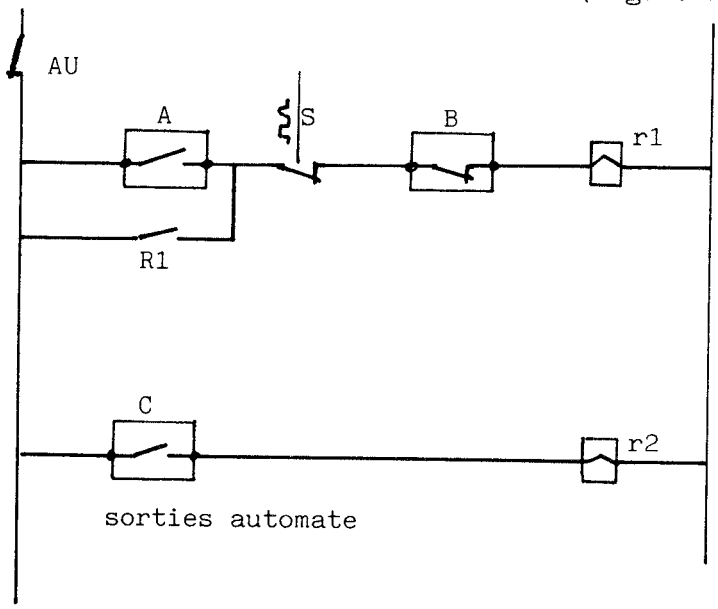
EXEMPLE : Le type B1 (type de base) est le préactionneur bistable à priorité fixe. Par convention le premier pilotage déclaré est celui qui est prioritaire.
Exemple :

01 PREACT-TYPE B1 { A } STOP { M } ROTATION ;

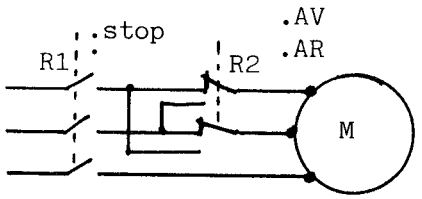
Indique que l'arrêt est obtenu de façon prioritaire par la variable A et que la rotation est provoquée par M.

Par contre pour le type B2 qui correspond à la priorité au 1er pilotage appliqué (courant en pneumatique), l'ordre de déclaration n'a évidemment pas d'importance.

EXEMPLE : Soit le schéma suivant : (figure 2-3)



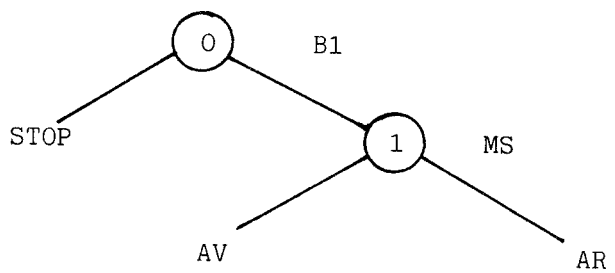
CIRCUIT MOYENNE PUISSANCE



CIRCUIT PUISSANCE

- FIGURE 2-3 -

L'arbre correspondant est le suivant : (figure 2-4)



- FIGURE 2-4 -

Compte-tenu du montage à arrêt prioritaire et du choix des contacts nous avons :

Le 1er préactionneur est de type B1, l'arrêt (vitesse STOP) est obtenu par B.S.AU, la marche par A ;

Le 2e préactionneur est monostable (type MS), la position de travail est obtenue pour C. $\overline{AU}$ qui conduit à la vitesse AR. L'absence de pilotage donne la vitesse AV .

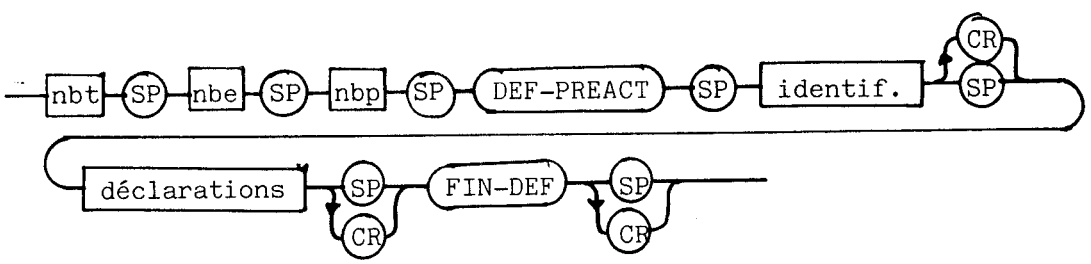
Les entrées AU, A, B, C, S étant supposées déclarées par une instruction DEFVARE, la déclaration de ces préactionneurs prend la forme :

```
PREACTIONNEURS broche SEQUENTIEL
            0 PREACT-TYPE B1 { B S AU } STOP { A } 1;
            1 PREACT-TYPE MS { C | AU } AR { } AV;

FIN-PREACT
```

II-3 Introduction d'un préactionneur en bibliothèque

Tout préactionneur représenté par un graphe d'état, peut être entré en bibliothèque. Nous conservons le formalisme du Grafcet.
La syntaxe est la suivante :



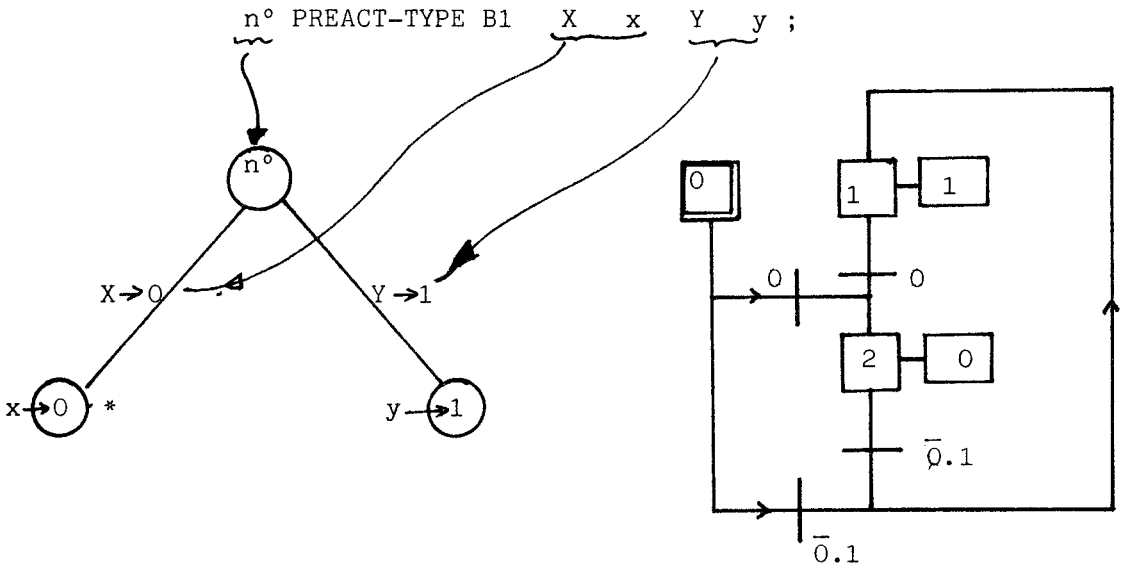
nbt représente le nombre de transitions ;
nbe représente le nombre d'étapes ;
nbp représente le nombre de positions du préactionneur.
L'identificateur est le nom utilisé dans les déclarations de type à l'image de MS, B1, B2.

Un préactionneur a deux représentations :

- l'arbre qui en spécifie l'utilisation ;
- le grafcet qui définit son évolution.

Ces deux représentations ne sont pas indépendantes. A la création du modèle elles doivent être interfacées.

A titre d'exemple nous traitons le cas du préactionneur B1 (figure 2-5).



- FIGURE 2-5 -

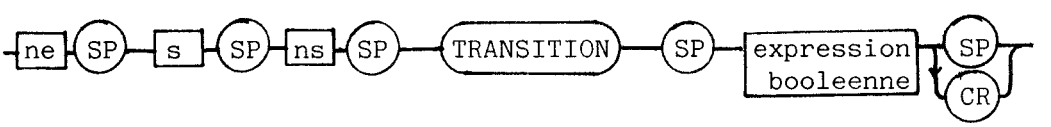
* pilotage prioritaire

X, Y représentent les fonctions de pilotage.

x, y sont des numéros de noeuds (ou des identificateurs de vitesse).

Les fonctions et sorties utilisées dans la phase descriptive sont des grandeurs muettes. Les équivalences entre les fonction X et 1, Y et 2 d'une part et entre les sorties x et 1, y et 2 d'autre part, sont établies automatiquement par le programme de traduction.

La déclaration d'une transition du graphe est de la forme :



où ne est le numéro de l'étape d'entrée de la transition ;

S est le numéro de la sortie associée à l'étape d'entrée ne ;

ns est le numéro de l'étape qui suit la transition.

L'expression booléenne est bâtie à partir des numéros de fonctions ou numéros de branche de l'arbre.

L'étape initialement active porte toujours le numéro 0.

En l'absence de sortie associée à l'étape d'entrée, le numéro (s) est remplacé par FP.

EXEMPLE :

Nous proposons ci-après les descriptions des préactionneurs de type B1, B2, MS. Nous proposons également celle d'un préactionneur 3 positions à priorité au 1er pilotage appliqué. Un distributeur 5/3 double pilotage, à centre fermé est de ce type.

```
298 LIST
SCR # 298
  0 ( BIBLIOTHEQUE DE PREACTIONNEURS  ) DECIMAL
  1 ( Type B1 : bistable a pilotage prioritaire fixe )
  2 ( la premiere sortie est prioritaire             )
  3 4 3 2 DEF-PREACT B1
  4     0 FP 1 TRANSITION { 0 }
  5     0 FP 2 TRANSITION { | 0 1 }
  6     1 0  2 TRANSITION { | 0 1 }
  7     2 1  1 TRANSITION { 0 }
  8  FIN-DEF
  9 ( Type B2 : bistable double pilotage , priorite )
 10 ( au premier applique                           )
 11 4 3 2 DEF-PREACT B2
 12     0 FP 1 TRANSITION { 0 | 1 }  0 FP 2 TRANSITION { | 0 1 }
 13     1 0  2 TRANSITION { | 0 1 }  2 1  1 TRANSITION { 0 | 1 }
 14 FIN-DEF
 15 -->
OK
299 LIST
SCR # 299
  0 ( Bibliotheque de preactionneurs : suite )
  1 ( type MS : preactionneur monostable     )
  2 2 2 2 DEF-PREACT MS
  3    0 0 1 TRANSITION { 0 }
  4    1 1 0 TRANSITION { | 0 }
  5  FIN-DEF
  6 ;S
  7
  8
  9
 10
 11


316
    LIST
SCR # 316
  0 ( Preactionneur 3 positions , priorite au 1er applique )
  1  4 3 3 DEF-PREACT T2
  2        0 1 1 TRANSITION { 0 }
  3        0 1 2 TRANSITION { 2 }
  4        1 0 0 TRANSITION { | 0 }
  5        2 2 0 TRANSITION { | 2 }
  6   FIN-DEF
  7
  8
  9
 10
 11
 12
 13
 14
 15
```

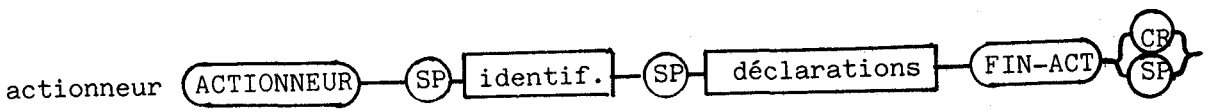
L'utilisation du préactionneur de type T2 conduit à :

```
2 PREACT-TYPE  T2 { A }  AV  { B }  RECUL  { }  BLOCAGE ;
```

On remarque que la fonction de pilotage 1 n'est pas utilisée (une position stable). Ceci justifie son absence dans la déclaration d'utilisation.
Si l'ensemble des préactionneurs associés à un actionneur peut être représenté par un graphe d'état, il est toujours possible de l'entrer globalement en machine sous un type particulier. A l'utilisation il y aura alors une seule ligne PREACT-TYPE (nom particulier).

III Déclarations des trajectoires

L'ensemble des matériels qui agissent sur une trajectoire est appelé actionneur.
Ceci justifie les termes utilisés dans la déclaration.
La syntaxe globale est :



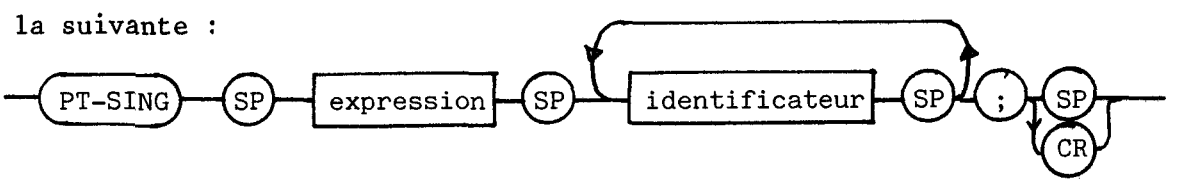
La trajectoire est décomposée en points singuliers. Chaque point singulier est caractérisé par un compte-rendu.
La déclaration de trajectoire comprend deux parties :

- la spécification des points singuliers ;
- l'association de ces points en trajectoire.

La déclaration des points singuliers d'une trajectoire précède toujours la description de la trajectoire.

III-1 Déclaration des points singuliers

Chaque point singulier est identifié par un nom. La syntaxe utilisée est la suivante :



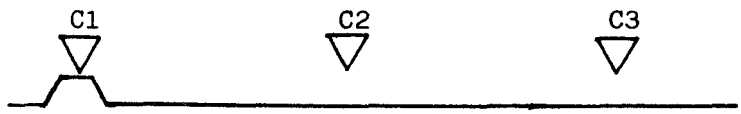
L'identificateur est le nom d'un point singulier.
L'expression booléenne est un ET entre les variables relatives à chaque capteur prises sous forme affirmée ou niée.
Seules les variables correspondant à la trajectoire décrite sont évidemment utilisées.

IMPORTANT

Afin d'éviter un temps de traduction très long, nous avons admis les contraintes suivantes :

- tous les points singuliers caractérisés par une même expression booléenne doivent être définis dans la même déclaration ;
- les variables doivent toujours avoir le même rang (ou le même ordre) dans toutes les expressions régulières.

Soit une canne ponctuelle qui se déplace devant trois capteurs (fig 2-6)



- FIGURE 2-6 -

La déclaration des points singuliers est la suivante :

```
315 LIST
SCR # 315
  0 ( Exemple de trajectoire )
  1  DEBUT
  2      13 DEFVARE C1  14 DEFVARE C2 15 DEFVARE C3
  3      ( declaration des preactionneurs )
  4      ACTIONNEUR came
  5         PT-SING { | C1 | C2 | C3 } PS2 PS4 ;
  6         PT-SING { C1 | C2 | C3   } PS1 ;
  7         PT-SING { | C1 C2 | C3   } PS3 ;
  8         PT-SING { | C1 | C2 C3   } PS5 ;
  9      ( Declaration des trajets )
 10
 11
 12
 13
 14
 15
OK
```

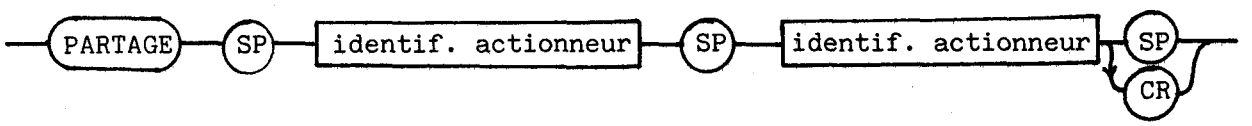
Capteurs communs à plusieurs actionneurs

Nous avons envisagé la possibilité d'avoir des capteurs communs à plusieurs actionneurs. L'observation d'un tel phénomène amène à considérer que les actionneurs ont des éléments de trajectoire commune.

Deux problèmes se posent alors. Le premier au niveau de la déclaration, le second au niveau de l'exploitation.

Par convention, le codage des points singuliers utilisé dans la déclaration est celui que l'on obtient si l'autre actionneur est hors du tronçon commun.

Au niveau de l'exploitation, ce phénomène modifie la stratégie du test notamment lors de la phase initiale de localisation. Pour guider le travail du traducteur, l'utilisateur doit alors utiliser la déclaration de partage dont la syntaxe est la suivante :



Il est à noter qu'en cas d'oubli de cette déclaration le système diagnostiquera de fausses erreurs.

Si deux actionneurs se partagent strictement la même trajectoire (sans tronçons spécifiques) le dispositif ne peut jamais localiser les actionneurs.

La stratégie de surveillance considère que les deux actionneurs ne sont jamais sur le même point singulier.

III-2 Organisation de la trajectoire

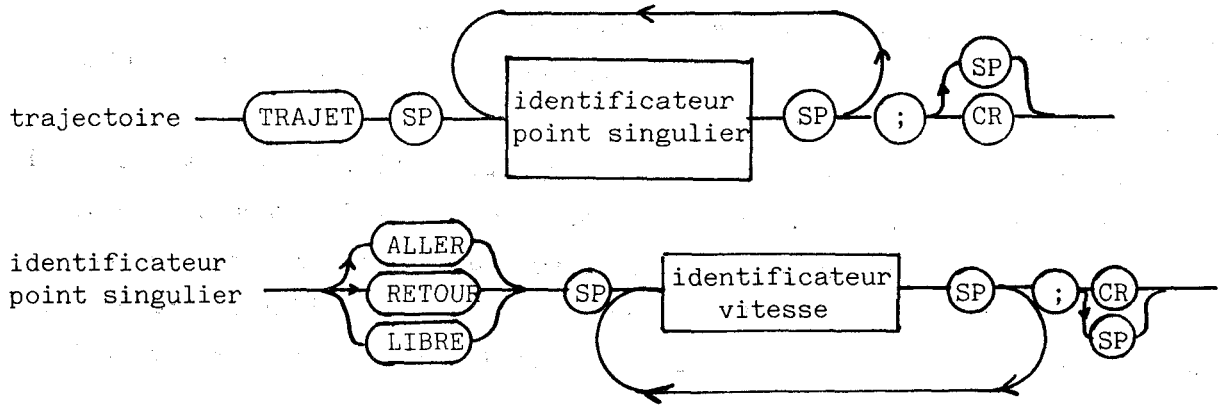
III-2-1 TRAJECTOIRE SIMPLE

La trajectoire est une liste ordonnée de points singuliers. La déclaration est dans ce cas immédiate. Pour limiter la taille de la structure de donnée, il est demandé à l'utilisateur de spécifier le sens de déplacement sur la trajectoire.

Trois cas sont explicitement envisagés.

- l'aller correspond à un déplacement sur la trajectoire dans le sens de la déclaration (sens positif arbitraire) ;
- le retour correspond à un déplacement en sens inverse ;
- la libération est utilisée lorsque la commande ne permet pas de connaître la vitesse (utilisée en cas d'arrêt d'urgence par exemple).

Implicitement toute commande qui n'est pas dans une de ces trois classes correspond à un arrêt.
La syntaxe est alors la suivante :



Exemple:

```
TRAJET PS1 PS2 PS3 PS4 PS5;
        ALLER AV RAPIDE AVLENTE ;
        RETOUR AR ;
```

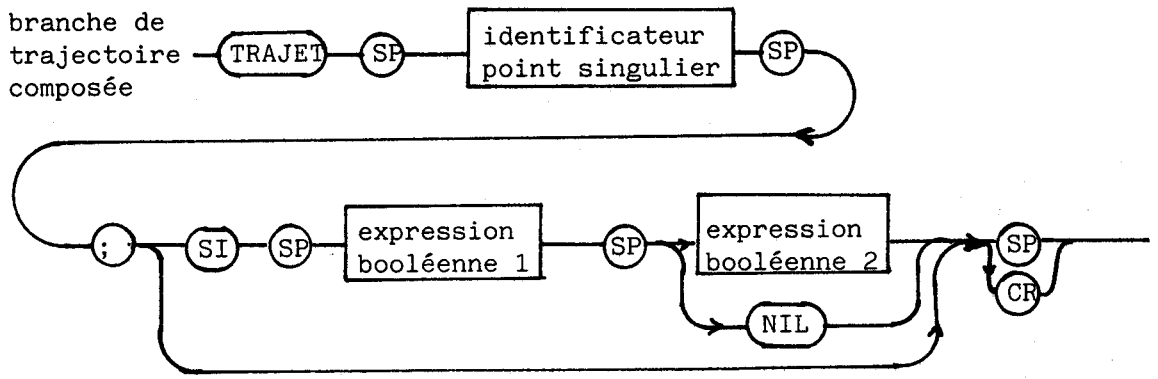
III-2-2 TRAJECTOIRE COMPOSEE

Une trajectoire peut présenter des bifurcations. Elle est alors décomposée en sous trajectoires qui sont des trajectoires simples. Le traducteur réassemble les tronçons en fusionnant les points singuliers de même nom à l'extrémité des tronçons.
Une trajectoire fermée est alors définie comme une trajectoire simple dont les points singuliers origine et extrémité ont même nom.

Une divergence dans la trajectoire entraine une perte d'information s'il est impossible de connaître la branche empruntée réellement. Dans ce cas la tolérance sur les temps est fixée par $|\text{Min min } P_i, \text{ Max max } P_i|$. Il y a donc un temps de latence plus grand.
Par contre si la sélection de branche se fait par un actionneur (grandeur mesurée) il est possible d'introduire un choix de branche par une expression

booléenne sur les capteurs.

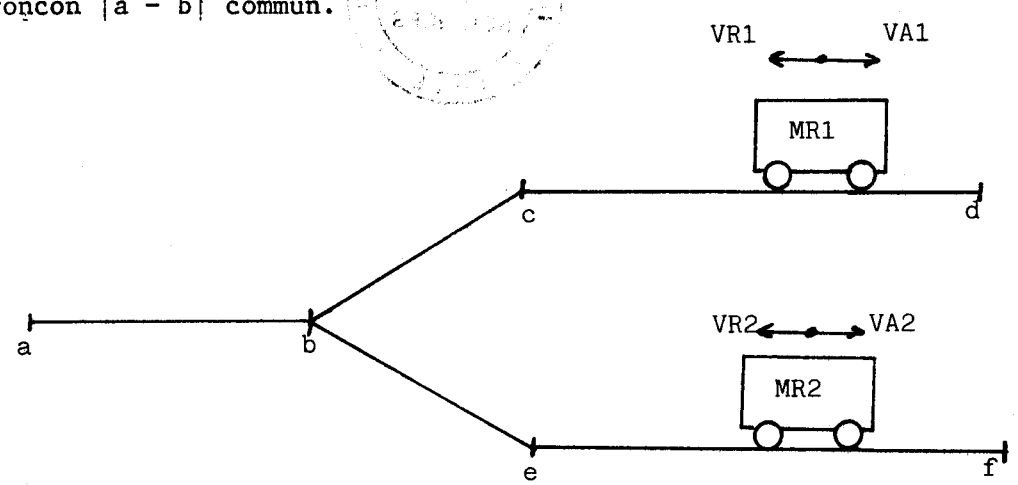
La syntaxe générale d'un tronçon de trajectoire est complétée comme suit:



Les exemples ci-après explicitent le langage.

EXEMPLE 1 (figure 2-7)

Soit deux chariots dont le déplacement est commandé par deux moteurs MR1 et MR2 suivant les trajectoires représentées à la figure ci-dessous avec un tronçon |a - b| commun.

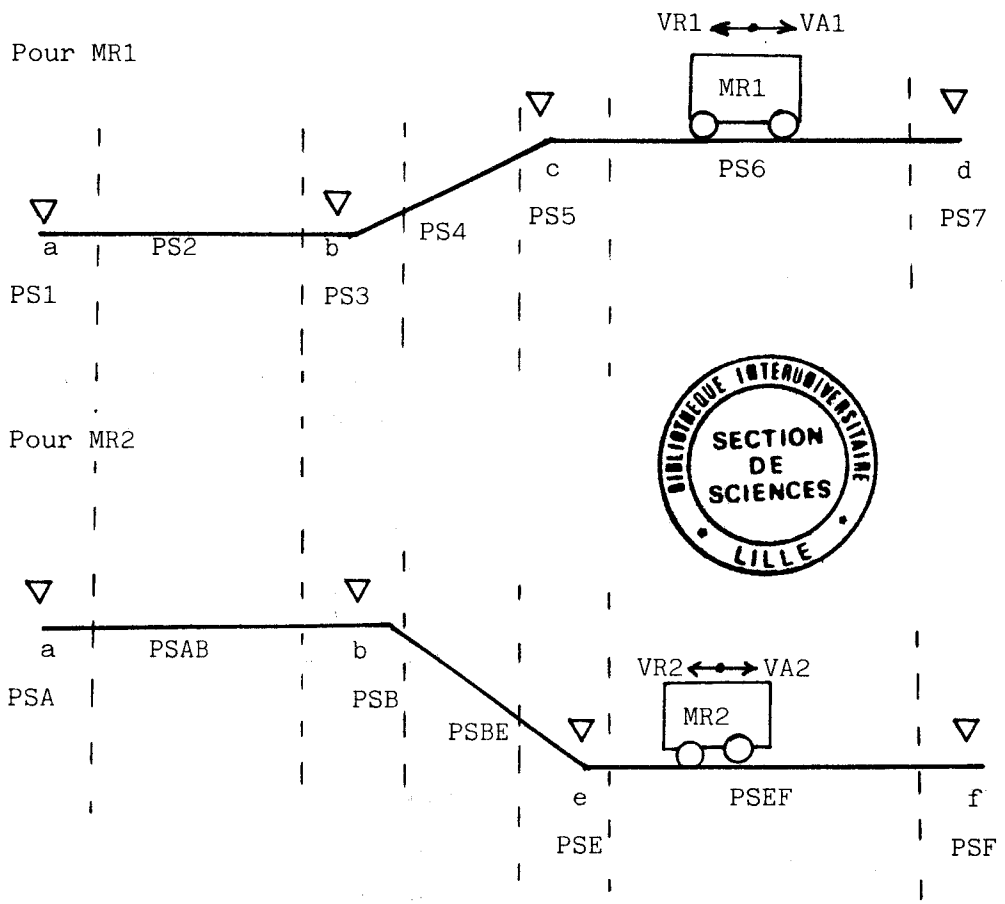


- FIGURE 2-7 -

Chaque moteur possède deux vitesses VA et VR obtenues par deux lignes d'ordres va, vr telles que :

vr \ va	0	1
0	--	VA
1	VR	--

La trajectoire de chaque actionneur est numérotée de la façon suivante (figure 2-8):



- FIGURE 2-8 -

RESUME

La sûreté de fonctionnement est une discipline qui vise à prévoir quantitativement le comportement d'un système face au risque de défaillance de ses composants. L'étude présentée est orientée vers la sûreté des systèmes utilisés en production automatisée.

La première partie discute des bases de la sûreté de ces systèmes. Elle met en évidence l'insuffisance des moyens habituellement mis en ouevre pour améliorer la sûreté de ces automatismes.

La deuxième partie présente les bases d'un mécanisme de test de la partie opérative par analyse syntaxique des comptes rendus qu'elle émet. L'enrichissement progressif du modèle ainsi créé par introduction d'éléments sémantiques, conduit àdifférents niveaux de test (statique, dynamique, dynamique avec contexte commande, dynamique pondéré).

La troisième partie traite de la mise en oeuvre. L'intégration matérielle et logicielle du mécanisme de test à l'automatisme ainsi que la création du modèle y sont discutés. La quantification des taux de couverture permet de comparer les performances des différents niveaux de tests. Il est également défini une stratégie permettant de localiser le composant défaillant à partir du type d'erreur mise en évidence par le mécanisme de test.

MOTS CLES: Fiabilité, Sécurité, Surté de fonctionnement, Autotest, Test en ligne, Maintenance assistée.