La sûreté de fonctionnement des automatismes

JOËL BERTHELOT[1]

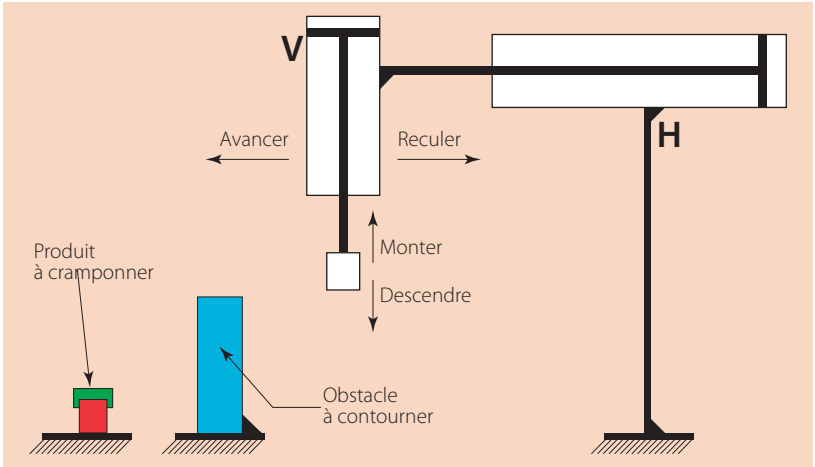
La sûreté de fonctionnement d'un système n'est pas un vain mot. Elle se définit en termes d'objectifs de qualité du système, exprimant la confiance dans la pérennité de son fonctionnement.
Il faut savoir que : 50 % des accidents sont dus à une défaillance du fonctionnement, 20 % à une défaillance d'un composant, 50 % affectent l'opérateur, et 80 % auraient pu être éliminés à la conception.
Par conséquent, la recherche de solutions sûres doit se faire dès la conception en tenant compte de tous les modes de fonctionnement ainsi que de tous les modes de défaillance des matériels. Et, pour cela, la mise en œuvre d'une méthode et l'emploi d'outils adaptés sont nécessaires.

mots-clés
actionneurs
automatismes
capteur
outil et méthode
partie commande
sûreté de fonctionnement

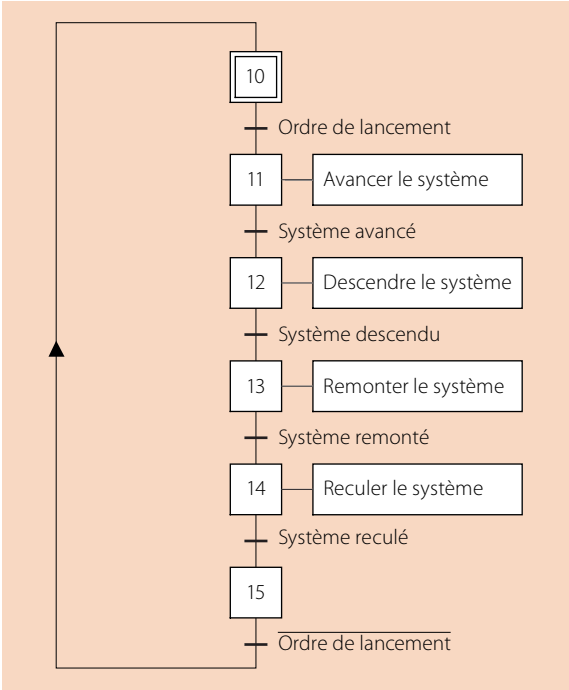
La sûreté de fonctionnement des SAP (Systèmes Automatisés de Production) recouvre deux concepts : **la sécurité** et **la disponibilité**.

L'aspect sécurité, fondamental, obéit à des réglementations très strictes. Il a déjà fait l'objet d'une littérature abondante. En revanche, la disponibilité, critère parfaitement mesurable *a posteriori*, est plus délicate à prévoir et à assurer. Sans être complètement oubliée des concepteurs, elle n'est donc que rarement une priorité. Or, si la sécurité est l'affaire de tous, la disponibilité ne doit pas ne concerner que les constructeurs de composants et les techniciens de maintenance ! En particulier, l'automaticien, lors de l'élaboration de la partie commande du SAP, peut apporter sa contribution à l'amélioration de la disponibilité.

Ce concept de disponibilité intègre deux volets : **la fiabilité**, c'est-à-dire l'aptitude à ne pas présenter de défaillance pendant une durée prévue, et **la maintenabilité**, qui caractérise l'aptitude à la remise en service après défaillance. Les deux sujets seront successivement abordés dans ces pages, et il sera proposé quelques solutions logicielles visant à leur amélioration.



1 **Un exemple simple d'une partie opérative représentative d'un système**



2 **Le grafcet de la tâche de manipulation**

Grafcet et fiabilité : un exemple de support

On suppose que le poste de travail d'un SAP correspond au schéma **1**. Il anime un système de cramponnage qui doit contourner un obstacle mécanique. Équipé de deux vérins, l'un à axe horizontal et l'autre à axe vertical, il est ici représenté dans sa position de repos. Son fonctionnement, généralement très simple, se résume dans notre exemple à un cycle en L qui peut être décrit par le grafcet de tâche **2**.

L'information *Ordre de lancement* est supposée délivrée par une étape d'un grafcet de coordination des tâches. L'étape 15 est alors une étape de fin de tâche assurant la synchronisation avec ce grafcet de coordination.

Jusque-là, rien que de très classique ; aucune difficulté non plus pour coder ce grafcet puis programmer son déroulement. Le cycle, très simple, doit alors s'exécuter correctement, sans surprise.

Il est cependant intéressant, pour ne pas dire indispensable, de s'interroger sur le comportement de l'ensemble à l'apparition d'un incident. En effet, si l'on ne peut éliminer tous les risques de défaillance, il est en

revanche de notre devoir de les prendre en considération et d'en limiter les conséquences par une action préventive – et chacun sait que la meilleure prévention s'obtient dès la conception.

Parmi les dysfonctionnements possibles, analysons les conséquences de celui lié au détecteur (60 % des défaillances des systèmes automatisés sont imputables à la fonction détection) délivrant l'information *Système remonté*, dans le cas où cette information reste vraie en permanence (information logique « figée » à l'état 1).

Examinons le déroulement du cycle :

❶ Le cycle s'effectue normalement jusqu'à l'activation de l'étape 12.

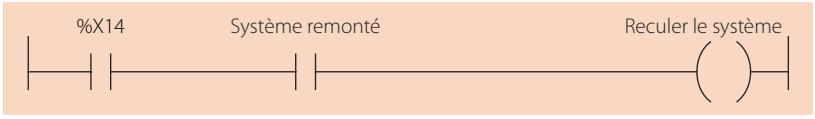
❷ La descente du système a lieu et l'information *Système descendu* apparaît.

❸ Le franchissement de la transition suivante entraîne alors l'activation de l'étape 13.

❹ La réceptivité associée à la transition entre 13 et 14 étant **déjà vraie**, cette transition est franchie, l'étape 14 est activée et l'étape 13 désactivée **sans qu'ait pu être réalisée l'action** *Remonter le système*.

La conséquence est évidemment **une collision** entre la tige du vérin vertical et l'obstacle sur la trajectoire !

Remarque : Même une condition restrictive sur la sortie API associée au mouvement de recul n'aurait pas empêché la collision **3**.



3 La programmation de l'équation de sortie

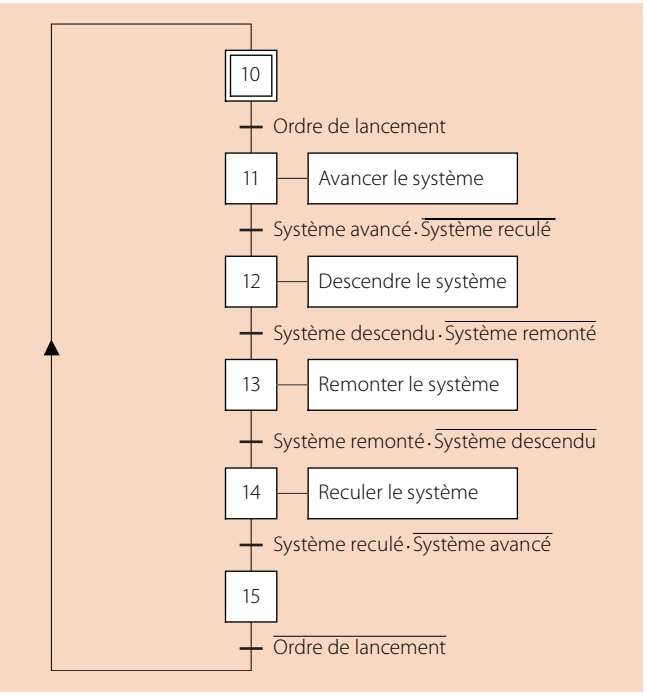
La prévention intrinsèque par autodétection de discordance

On peut imaginer plusieurs solutions, plus ou moins sophistiquées, pour anticiper un incident de ce type, parmi lesquelles celle-ci : **à chaque mouvement d'actionneur, on décide de contrôler non seulement l'état actif du détecteur de fin de course associé à la fin de ce mouvement, mais également l'état inactif du détecteur de début de course.**

Le grafcet devient alors celui donné en **4**.

L'intérêt est multiple :

- Cycle stoppé dès l'étape où le dysfonctionnement apparaît ;
- Détection immédiate de discordance (informations corrélées incohérentes) ;



4 Le grafcet de tâche avec la détection de défaut par discordance

- Prévention du risque de collision ou de non-qualité imputable à une information erronée ;
- Simplicité de mise en œuvre ;
- Coût matériel nul ;
- Possibilité d'alerter rapidement le personnel d'intervention ;
- Plus grande facilité pour **diagnostiquer** les causes d'une interruption de production.

C'est sur ces deux derniers points que la suite de notre étude va porter.

Le constat d'interruption et l'alerte opérateur

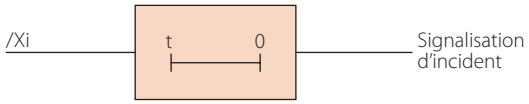
L'interruption immédiate du cycle, dès l'apparition d'une discordance, est une mesure **préventive** de nature à éviter une casse matérielle consécutive à une information erronée délivrée par un détecteur de fin de course. Elle améliore donc la fiabilité et, par voie de conséquence, la disponibilité du SAP. En l'absence de risque matériel, elle reste un facteur de qualité. Toutefois, l'arrêt du processus reste pénalisant en termes de productivité. Il est donc important de réduire au minimum la durée de cette interruption.

Si le temps de cycle est court (de quelques secondes à quelques dizaines de secondes), on peut envisager une surveillance globale du cycle par **contrôle du respect d'un « temps enveloppe »** (lire en encadré « Le principe du "temps enveloppe" » page suivante).

[1] Professeur agrégé de génie mécanique en STS MAI au lycée Georges-Dumézil de Vernon (27).

Le principe du « temps enveloppe »

À chaque début de cycle, on active une temporisation d'une durée un peu supérieure à la durée maximale d'un cycle. La durée retenue pour le temps enveloppe doit englober la durée du cycle avec une marge suffisante pour ne pas signaler d'incident de façon intempestive. Dès la fin du cycle, la temporisation est désactivée.



Où Xi est l'étape initiale du grafcet de cycle
En l'absence d'incident, la temporisation est normalement arrêtée avant son terme, à la fin du cycle. En revanche, si la fin de temporisation est atteinte, cela signifie que le cycle ne s'est pas terminé dans le temps imparti ; on en déduit une interruption anormale du processus.

Lorsque le temps de cycle est plus long, le temps de réaction que suppose cette méthode peut être jugé inacceptable. Toutefois, le même principe peut alors être appliqué en considérant non pas le cycle complet, mais chacune des tâches individuelles qui composent le cycle. La durée de chaque temporisation devra alors être adaptée à la durée d'exécution de la tâche correspondante.

On peut encore envisager une surveillance beaucoup plus fine où la durée de chaque action est contrôlée individuellement ; une telle mesure, parfaitement justifiée dans certains cas, est cependant lourde à mettre en œuvre et longue en mise au point. (Notons que la surveillance individuelle d'une action critique par la méthode du temps enveloppe peut également être utilisée à d'autres fins : en liaison avec un ordinateur, on peut ainsi envisager une analyse statistique des durées ou l'observation d'une dérive dans le temps permettant d'anticiper les effets d'une usure matérielle.)

La maintenabilité par l'aide au diagnostic

Si l'action préventive développée initialement n'était guidée que par le souci de détecter au plus tôt une défaillance de détection, l'analyse du dysfonctionnement que nous allons proposer peut également s'appliquer à d'autres sources d'incident.

Un dysfonctionnement du système peut avoir des origines très diverses (détecteur endommagé ou déréglé, fil coupé, distributeur grippé, moteur ou vérin hors service, etc.). La localisation du composant défectueux et la cause précise de l'incident paraissent alors difficiles à établir autrement qu'à partir d'une analyse humaine. Il est toutefois possible de guider le technicien de maintenance en lui fournissant quelques éléments d'information que la partie commande sait recueillir et peut analyser. Après l'avoir alerté pour solliciter son intervention dans les meilleurs délais, on pourra donc l'aider à **localiser l'incident** et à en **déterminer la cause probable**.

Nota : On exclut *a priori* toute erreur de programmation du cycle décrit par le grafcet, les phases de test et de mise au point permettant de corriger ce type d'aléas.

La localisation de l'origine de l'interruption du cycle

Pour localiser la cause de l'arrêt, il est nécessaire d'identifier l'action qui a conduit à l'interruption du cycle. Il s'agit donc de repérer quelle fin de mouvement n'a pu être vérifiée. Une analyse exhaustive des sorties, bien que possible, est inutile ; en effet, seule une action pilotée peut être à la source du défaut.

Nota : Dans un premier temps, notre étude se limitera aux mouvements réalisés par vérins ou moteurs travaillant entre deux fins de course.

Pour chacune des sorties pilotées, il suffit alors de contrôler l'état des deux détecteurs associés à l'action correspondante :

- Le détecteur de début de course doit être à l'état logique 0 ;
- Le détecteur de fin de course doit être à l'état logique 1.

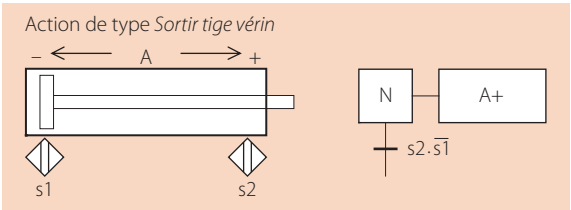
Toute autre combinaison logique empêchera l'évolution du grafcet.

Conclusion

Lorsqu'une action est commandée, si la combinaison logique du couple de détecteurs associé au mouvement n'est pas celle attendue, alors l'action correspondante est vraisemblablement à l'origine de notre incident.

L'analyse de situation et la détermination de la cause probable d'incident

Analysons, sur une action simple, les causes pouvant mener au blocage de l'évolution d'un grafcet construit selon le modèle donné en **4**. Le blocage du grafcet à l'étape N, qui commande l'action *A+*, permet d'affirmer que la réceptivité $s2 \cdot \overline{s1}$ n'a pas été obtenue **5**. Or, à



5 **La dualité partie opérative, partie commande**

partir des informations binaires s1 et s2, il existe quatre combinaisons possibles de leur association logique en ET : celle attendue et trois autres qui témoignent donc d'un incident. À chacune de ces trois dernières combinaisons correspond une cause probable d'incident :

- s2 = s1 = 1

Constat : Le détecteur s2 a bien commuté à 1 alors que s1 est resté figé à l'état 1.
Cause probable : Le détecteur de début de course s1 est resté anormalement actif. Il est vraisemblablement défectueux (discordance).

- s2 = 0 et s1 = 1

Constat : Aucun des deux détecteurs n'a commuté.

Cause probable : L'action *A+* n'a vraisemblablement pas été réalisée.

- s1 = s2 = 0

Constat : s1 a bien commuté à 0 alors que le détecteur de fin de course s2 (A sorti) est resté inactif.
Cause probable : Soit l'action *A+* a été interrompue soit s2 est déréglé, déconnecté ou défectueux.

Remarque : L'analyse de l'action *Rentrer tige vérin* conduit à des conclusions comparables en inversant toutefois les rôles respectifs de s1 et s2.

Conclusion

Le tableau 6 donne les différentes combinaisons logiques du couple (détecteur de fin de course, détecteur de début de course) et les diagnostics correspondants.

6 Les diagnostics correspondant aux états des détecteurs

État des entrées détecteurs	Diagnostic réalisé à partir de l'observation des entrées*
(0,0)	Défaut détecteur de fin de course ou mouvement interrompu
(0,1)	Le mouvement n'a pas été réalisé
(1,1)	Défaut détecteur de début de course
(1,0)	*C'est l'état attendu : pas de défaut signalé*

* On indique le défaut le plus probable afin de guider le technicien de maintenance, mais d'autres défaillances sont possibles.

La signalisation de l'incident et du diagnostic réalisé

Aujourd'hui, les terminaux de dialogue sont à la fois très performants, très simples à programmer et d'un coût toujours plus compétitif. Il est donc intéressant de les exploiter pour un échange d'informations de plus en plus riche. Ainsi, si l'analyse de diagnostic peut être facilement développée et prise en charge par l'API, un terminal de type texte, ou mieux encore graphique, permettra d'indiquer à l'opérateur ou au technicien de maintenance les conclusions de cette analyse.

Le message devra indiquer :

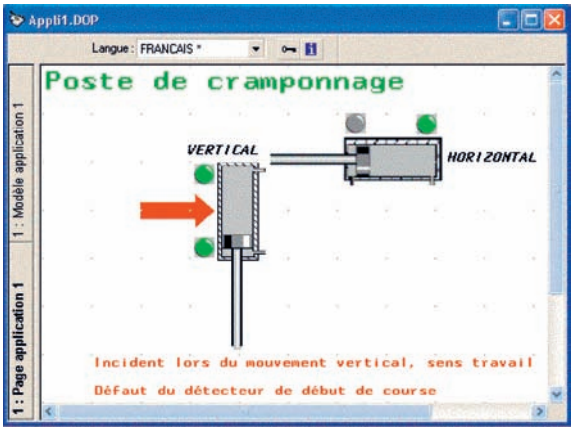
- L'actionneur concerné par le défaut ;
- Le mouvement attendu (travail ou repli) ;
- Le diagnostic réalisé.

Si l'on dispose d'un terminal graphique, on peut avantageusement compléter le message avec un schéma du système (ou du sous-système) représentant les différents composants dans leur état supposé. Il est également simple de désigner l'actionneur en défaut.

Un exemple

Étudions une page-écran 7 qu'un terminal graphique (ici de type XBT F de Schneider) peut afficher dans le cas d'un dysfonctionnement tel que celui évoqué sur notre exemple support :

- Chaque vérin est représenté dans son état supposé à l'aide d'une boîte à image.
- Le premier champ alphanumérique signale l'actionneur sur lequel le défaut est apparu.



7 La signalisation, sur pupitre graphique d'exploitation, des dysfonctionnements

- Le deuxième précise le sens du mouvement attendu.
- Le troisième indique la cause probable diagnostiquée.
- Chaque détecteur est également représenté par une boîte à image, **dans une couleur différente selon son état** tel qu'il est « vu » par l'API (si le détecteur est équipé d'une led, une incohérence entre l'image et l'état de la led témoignera d'une erreur dans la chaîne d'acquisition de l'information).
- Chaque flèche est également dans une boîte à image et n'apparaît que pour désigner l'actionneur en défaut sans ambiguïté.

Récapitulatif, synthèse et implémentation de la méthode

À l'apparition d'un incident, plusieurs phases d'analyse sont nécessaires pour stopper le cycle et renseigner l'opérateur sur la source et la cause du dysfonctionnement :

1. Conception des grafcets avec prévention intrinsèque d'incident
2. Surveillance du cycle et constat de l'interruption
3. Localisation de la situation de blocage, c'est-à-dire de l'action qui n'a pu être vérifiée
4. Détermination de la cause probable à partir de l'analyse de l'état des détecteurs associés à cette action
5. Signalisation, sur terminal d'affichage, du diagnostic réalisé (origine probable)

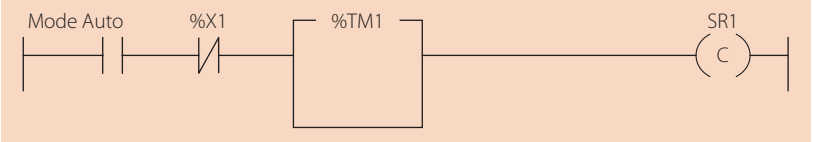
Le principe de traitement associé et des éléments d'implémentation

La conception des grafcets intégrant la prévention d'incident

Il suffit de contrôler, à chaque évolution, que les détecteurs – celui de fin mais également celui de début de course – du mouvement contrôlé ont effectivement changé d'état.

Le constat d'interruption

L'implémentation en langage *ladder* de la méthode du temps enveloppe est donnée en 8.



8 La programmation en ladder du temps enveloppe

La durée de la temporisation *%TM1* sera choisie légèrement supérieure à celle d'un cycle automatique complet.

Le module logiciel (sous-programme) *SR1* contiendra les procédures d'analyse de situation et de diagnostic de défaut ; ainsi, elles ne seront exécutées qu'en cas de dysfonctionnement (dépassement du temps enveloppe).

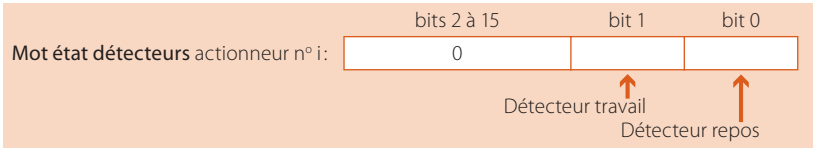
La localisation de l'action en défaut

Le principe de détermination de l'action en défaut est particulièrement simple : il consiste à repérer quelle est l'action demandée et non réalisée. Pratiquement, il suffirait donc, pour chacune des sorties activées, de contrôler l'état du détecteur de fin d'action associé ; s'il est resté à l'état logique 0, on peut en conclure qu'il s'agit de l'action recherchée.

Toutefois, comme nous l'avons vu, ce simple constat serait insuffisant pour diagnostiquer les causes de l'incident. Par conséquent, et afin de systématiser notre analyse, nous observerons quelques règles de codage, peu contraignantes mais efficaces pour généraliser le traitement.

Les principes de codage et d'adressage

Pour chaque action réalisable, il pourra être nécessaire de contrôler l'état des deux détecteurs associés, de début et de fin de course. Pour cette raison et pour faciliter l'exploitation ultérieure, on aura intérêt, dans un premier temps, à copier dans un mot dédié les deux bits d'état des détecteurs associés à un actionneur selon le principe donné en 9.



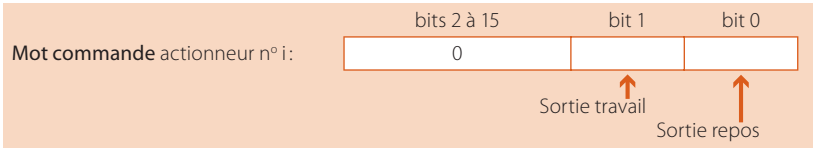
9 Le principe du mot image de l'état des détecteurs

Exemple d'implémentation en littéral structuré :

```
(* Acquisition et recopie des états des couples de détecteurs de fins de course *)
EtatVerinH :=%I1.0 :2 ;
EtatVerinV :=%I1.2 :2 ;
etc.
```

Remarque : Les adresses des entrées des détecteurs associées à un même actionneur sont ici choisies consécutives pour faciliter le codage.

Afin de simplifier la comparaison de l'état des détecteurs associés à un actionneur avec la commande qui lui est appliquée, comme pour les entrées, on aura intérêt à copier dans un mot dédié les deux bits de sorties associés à cet actionneur selon le principe donné en 10.



10 Le principe du mot image de la commande de l'actionneur

Exemple d'implémentation en littéral structuré :

```
(* Recopie dans des mots des couples de sorties associées aux vérins *)
CommandeVerinH :=%Q2.0 :2 ;
CommandeVerinV :=%Q2.2 :2 ;
etc.
```

Remarque : Comme précédemment, les adresses des entrées des détecteurs associées à un même actionneur sont ici choisies consécutives pour faciliter le codage.

Guidé par le souci de généraliser le traitement, on choisira une liste de mots d'adresses consécutives pour mémoriser l'état des couples de détecteurs et une seconde liste de mots consécutifs pour y copier la situation de commande de l'ensemble des actionneurs.

L'ordre retenu pour ces deux listes devra être identique afin d'établir une **correspondance entre mots de même rang** 11.

11 L'adressage séquentiel des mots d'états

Adressage des listes :	Liste des mots d'état	Liste des mots de commande
Actionneur **1**	Mot d'adresse **i** + 1	Mot d'adresse **j** + 1
Actionneur **n**	Mot d'adresse **i** + **n**	Mot d'adresse **j** + **n**

Application

Un premier examen de chacun des mots de commande permettra de mettre hors de cause tous les actionneurs qui ne sont pas sollicités (le mot de commande correspondant aura alors la valeur 0). En effet, l'actionneur à l'origine du défaut se trouve obligatoirement parmi les actionneurs sollicités.

Ensuite, une simple comparaison entre chaque mot de commande d'actionneur sollicité et le mot d'état correspondant permettra l'identification formelle de l'action sur laquelle le défaut est apparu. En effet, si la valeur du mot de commande d'un actionneur et celle de son mot d'état sont différentes, cela signifie qu'il y a incohérence entre l'état attendu et celui obtenu ; on peut en déduire qu'il s'agit de l'actionneur concerné par le dysfonctionnement.

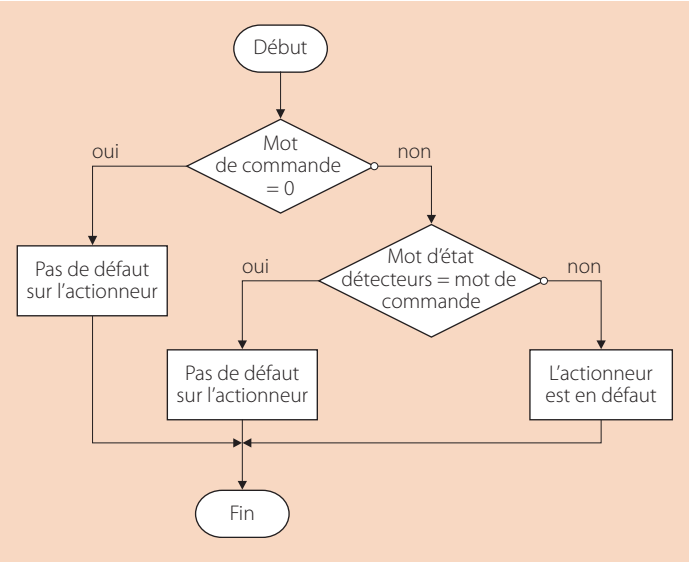
En résumé, si l'on fait référence à l'actionneur de la figure 5, on devra vérifier

$$(A+, A-) \neq (0, 0)$$

puis, le cas échéant :

$$(A+, A-) \neq (s2, s1)$$

Ainsi, une analyse algorithmique utilisant le codage précédent abouti à l'organigramme 12 pour un actionneur.



12 L'algorithme de détection de défaut d'un actionneur

L'identification de l'actionneur en défaut

Une variable mot *Numero_actionneur* peut être utilisée pour citer l'actionneur en défaut si elle est associé à un champ texte de type liste énumérée 13. Cette liste doit impérativement être ordonnée conformément au classement retenu pour l'adressage des mots de commande et d'état.

Nota : La même variable peut également permettre de faire apparaître, sur la page d'un écran graphique, la flèche qui désigne cet actionneur ; l'apparition des différentes images est alors liée à la valeur de ce mot.

L'organigramme 12 devient celui donné en 14.

Nota : L'examen de tous les actionneurs par adressage indexé dans une boucle itérative n'est envisageable que si les adresses des mots de commande ainsi que celles des mots d'état ont été choisies consécutives. Les deux listes de mots ainsi constituées doivent également être classées dans le même ordre (voir le tableau d'adressage 11). Ainsi, la variable mot *Numero_actionneur* recevra le rang de l'actionneur en défaut dans la liste des actionneurs ainsi classée.

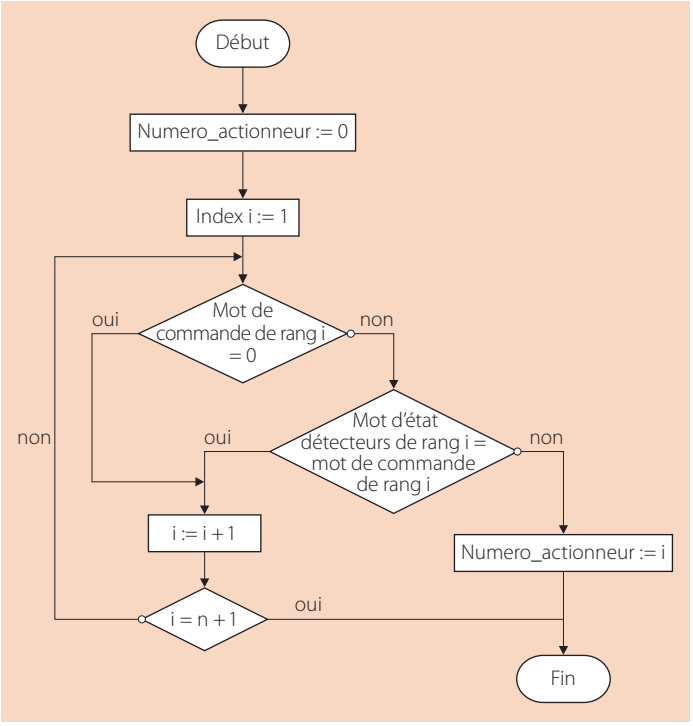
13 L'affectation des numéros de l'actionneur

Valeur	Texte correspondant
0	
1	Actionneur 1
2	Actionneur 2
n	Actionneur n

Le diagnostic du défaut

En cas de défaut, une variable mot *Defaut* recevra la valeur contenue dans **le mot d'état des détecteurs** de l'actionneur en défaut. Cette valeur sera 0, 1, 2 ou 3 :

- La valeur 0 (les deux fins de course sont à l'état logique 0) témoigne en principe d'un défaut sur le détecteur de fin de course.
- La valeur 1 ou la valeur 2 signifient que le mouvement (travail ou repos) ne s'est pas exécuté.
- La valeur 3 (les deux fins de course sont à l'état logique 1) témoigne en principe d'un défaut sur le détecteur de début de course.



14 L'algorithme général de détection de défaut des actionneurs d'un système

La signalisation du défaut

Ainsi, la variable mot *Defaut* sera utilisée pour indiquer la cause probable, là encore associée à un champ texte de type liste énumérée 15.

15 Le codage des messages de défaut

Valeur	Texte correspondant
0	Mouvement interrompu ou détecteur de fin de course HS (état 0)
1	Le mouvement de travail n'a pas été exécuté
2	Le mouvement de repli n'a pas été exécuté
3	Le détecteur de début de course est déréglé ou HS (état logique 1)
4 *	

* La valeur 4 a été choisie pour initialiser la variable *Defaut* ; elle correspond à l'absence de défaut constaté.

La désignation explicite de l'actionneur et l'indication de la cause probable du défaut ne permettent toutefois pas de connaître le sens du mouvement attendu et, par voie de conséquence, quel est le détecteur nommé « début de course » ni quel est celui appelé « fin de course » (dans les cas des valeurs 0 et 3).

Pour remédier à cela, il suffit de lire le mot de commande associé :

- Dans le cas où il vaut (1,0), il s'agira du mouvement de travail ;
- Dans le cas où il vaut (0,1), il s'agira du mouvement de repli.

Ainsi, en associant la variable mot *Mouvement* à un champ texte de type liste énumérée, on obtient le tableau 16.

16 Le couplage des mots d'état des détecteurs avec les messages de défaut

Couple état détecteurs	Valeur	Texte correspondant
(0,1)	1	Mouvement de travail
(1,0)	2	Mouvement de repli
	4	*Valeur d'initialisation*

Il faut toutefois noter que cette liste n'est valable que si tous les mots de type *Commande_actionneur* ont été assignés selon le principe défini au début de ce récapitulatif.

Synthèse

Exemple d'implémentation :

```
(* Localisation du défaut *)
Numero_actionneur :=0 ;          (* Numéro actionneur repéré *)
Mouvement :=4 ;                  (* Indication du sens du mouvement *)
Defaut :=4 ;                     (* Cause probable du défaut *)
For index :=1 to Dernier_actionneur do
        If Commande_actionneur[index]<>0 then
                If Etat_detecteurs[index] <> Commande_actionneur[index] then
                        NuméroActionneur :=index ;
                        Mouvement :=Commande_actionneur ;
                        Defaut := Etat_detecteurs ;
                End_if ;
        End_if ;
End_for ;
```

Par exemple, dans le cas d'un incident survenu sur l'actionneur λ, le détecteur de début de course étant resté actif, on pourra obtenir, en juxtaposant trois champs texte distincts, les deux messages suivants :

Incident survenu durant l'exécution du mouvement de travail de l'actionneur λ

Le détecteur de début de course est déréglé ou HS (état logique 1)

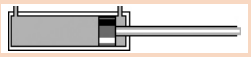
La représentation graphique de l'état de la PO

Dans le cas où l'on dispose d'un terminal graphique, on peut montrer tous les détecteurs tels qu'ils sont « vus » par l'automate, mais également tous les actionneurs dans leur état supposé. En l'absence de défaut, la lecture de l'état des détecteurs associés à un actionneur permet de choisir une image représentant l'état de cet actionneur sans ambiguïté.

L'exemple du cas d'un vérin tige rentrée au repos est donné en 17.

En présence d'un défaut, lorsqu'un seul des deux détecteurs est à l'état 1, nous avons supposé que l'actionneur n'avait pas exécuté son mouvement. L'image choisie est alors déduite, là encore, de l'état des détecteurs.

17 Le mot d'état et l'image associée

Couple état détecteurs	Valeur du mot d'état	Valeur du mot de commande	Champ image associé
(0,1)	1	1	
(1,0)	2	2	

Les choses se compliquent un peu lorsque l'état des deux détecteurs est incohérent.

Ainsi, au couple d'état (1,1) des détecteurs peuvent correspondre les deux images précédentes ; l'un des deux détecteurs est certainement hors service ou déréglé, mais il est difficile *a priori* de dire lequel. L'unique façon de trancher consiste à examiner la commande de cet actionneur. En effet, l'état supposé correspond vraisemblablement à l'état attendu, l'un des détecteurs ayant conservé son état 1 tandis que l'autre a pris l'état 1 ; on peut donc en déduire que le mouvement demandé a été réalisé. C'est donc le détecteur de début de course qui a conservé anormalement son état 1.

Au couple d'état (0,0) peuvent correspondre trois images distinctes : les deux images précédentes si l'on privilégie l'hypothèse selon laquelle le détecteur de fin de course est hors service ou une troisième qui représente l'actionneur à mi-course si l'on privilégie la thèse du mouvement interrompu.

La liste des images selon les cas de figure est donc celle du tableau 18.

En résumé, lorsqu'un et un seul des détecteurs est actif (mot d'état des détecteurs = 1 ou 2), alors l'image associée est déduite du mot d'état des détecteurs. Dans les autres cas (mot d'état des détecteurs = 0 ou 3), on se réfère au mot de commande pour choisir l'image la plus vraisemblable.

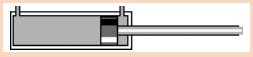
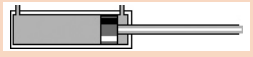
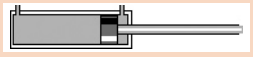
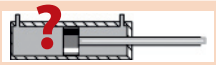
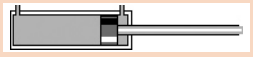
Il devient donc nécessaire de définir une variable mot spécifique pour l'affichage des différentes images dans une boîte à image. Nous appellerons ce mot *Etat_suppose* de l'actionneur puisque l'image choisie résulte d'une analyse.

La synthèse de notre analyse peut être représentée graphiquement par l'organigramme 19.

Pour représenter graphiquement, sur terminal d'affichage, l'état des actionneurs, on ajoutera donc le module de programme suivant :

```
(* Etat supposé des actionneurs *)
For index :=1 to Dernier_actionneur do
    If (Etat_detecteurs[index]=1 or Etat_detecteurs[index]=2) then
        Etat_suppose[index] := Etat_detecteurs[index] ;
        Elsif Commande_actionneur[index]=1 or Commande_actionneur[index]=2 then
                Etat_suppose[index] := Commande_actionneur[index] ;
        Else Etat_suppose[index] := Etat_suppose[index];
    End_if ;
End_for ;
```

18 Le combinatoire de détection des défauts

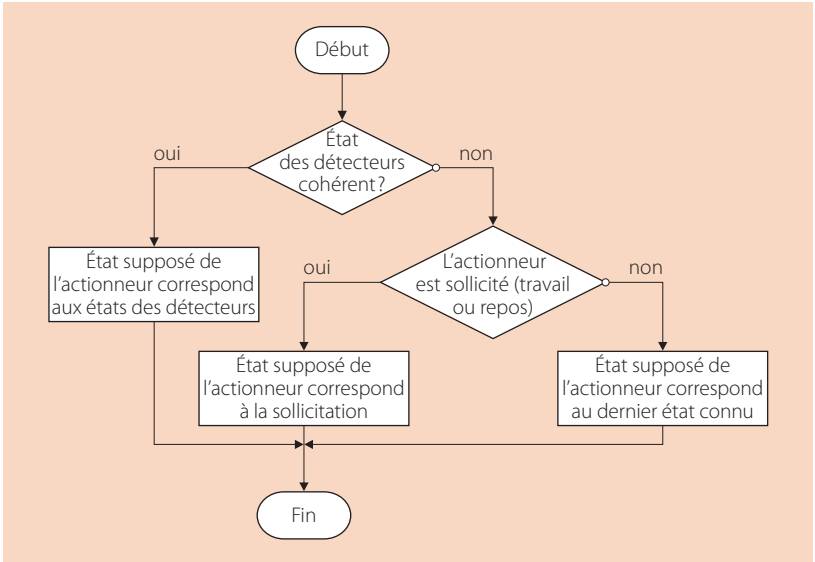
Couple état détecteurs	Valeur du mot d'état	Valeur du mot de commande	Champ image associé
(0,1)	1	1	
		2	
(1,0)	2	1	
		2	
(1,1)	3	1	
		2	
(0,0)	0	Quelconque (1 ou 2)	
		1	
		2	

Les restrictions d'application de la méthode

La méthode précédemment développée suppose que soit associé un détecteur à chaque (fin de) mouvement. Il est exclu de faire l'économie d'un détecteur de fin de course par vérin, aussi faible soit sa course et même s'il n'existe aucun obstacle sur la trajectoire de la tige. En effet, l'analyse de diagnostic repose essentiellement sur le contrôle d'incohérence entre les états des deux détecteurs ; la suppression de l'un d'eux n'autorise donc plus les déductions mettant formellement en cause l'état d'un détecteur plutôt que l'absence de mouvement réalisé.

Pour en savoir plus

BOUTEILLE (Daniel), *Les automatismes programmables,* Cépaduès éditions, 1987



19 La détermination de l'état supposé de l'actionneur

Les extensions de la méthode

Bien souvent, il sera nécessaire de tenir compte de particularités non envisagées ici. Il faut notamment citer tous les actionneurs à effet unique contrôlés chacun par un détecteur unique ; on observe alors un seul changement d'état de détecteur pour signaler l'action réalisée.

Par exemple, si le poste de l'exemple support est équipé d'une ventouse permettant la prise, le maintien puis la dépose d'une pièce, l'information *Pièce aspirée* pourra également être à l'origine d'un dysfonctionnement. Pour tenir compte de cette source supplémentaire d'incident, une comparaison entre la commande de l'aspiration et l'information *Pièce aspirée* permettra de diagnostiquer la « pièce lâchée » (il faut toutefois noter qu'il ne s'agit pas d'une généralisation de la méthode mais de son adaptation).

Un grand nombre d'exceptions peuvent ainsi être traitées selon le même principe, ce qui permet de couvrir la plupart des situations rencontrées sur les SAP de type séquentiel.

Prévention, intervention, qualité

L'exemple support est volontairement simple afin de présenter le plus clairement possible les bases de l'analyse proposée.

La méthode est facile à mettre en œuvre, rapide à développer et se prête à une programmation structurée sans interférer avec le reste du programme de commande. Indépendante du cycle et de sa description par grafcet, elle peut être développée très tôt, dès que les capteurs et actionneurs sont déterminés, sans être remise en question lors des différentes évolutions de l'analyse.

D'autre part, elle oblige à une démarche rigoureuse dès le codage des E/S.

Elle peut également être conduite *a posteriori,* et même sur une machine existante (au prix toutefois d'un codage un peu plus conséquent).

Outre les situations de défaut en production automatique, cette méthode permet également de détecter et de signaler l'origine d'un défaut d'initialisation machine (si toutefois les grafcets d'initialisation sont conçus selon le principe de prévention intrinsèque défini dans la méthode).

Elle a pu être pratiquée avec succès sur plusieurs projets industriels, et il faut également noter qu'avant même la mise en production elle s'est déjà révélée précieuse pendant les phases de mise au point. Elle permet en effet une réduction significative du temps de mise au point en signalant aux techniciens les causes d'arrêt. Elle facilite également les interventions pour réglage.

Enfin, indirectement, elle contribue à améliorer la qualité puisqu'on acquiert l'assurance que chaque mouvement a systématiquement été réalisé complètement. ■