Influence des performances d'une architecture informatique sur la fiabilité des systèmes échantillonnés

Fabrice Jumel, Nicolas Navet, Françoise Simonot-Lion

Influence des performances d'une architecture informatique sur la fiabilité des systèmes échantillonnés

Fabrice Jumel, Nicolas Navet, Françoise Simonot-Lion

4 mars 2003

LORIA - TRIO

ENSEM - 2, avenue de la Forêt de Haye

54516 Vandoeuvre les Nancy - France

jumel@loria.fr, nnavet@loria.fr, simonot@loria.fr

tel : 03 83 59 55 79 - fax : 03 83 59 56 62

Résumé

Dans cette étude, nous montrons, sur l'exemple de la régulation d'une cuve d'eau, comment lier la probabilité d'apparition d'une erreur de contrôle avec la probabilité de défaillance en considérant la dynamique du système régulé et les algorithmes de contrôle. Précisément, nous calculons sur différentes architectures opérationnelles deux métriques de fiabilité qui sont le temps moyen entre deux défaillances du système et la probabilité de défaillance en une heure de fonctionnement. Lorsque la fiabilité est une contrainte importante, comme c'est le cas dans certaines applications comme le X-by-Wire, de telles études sont nécessaires pour valider des choix de conception comme la mise en place de mécanismes de tolérance aux fautes ou le choix de l'Architecture Matérielle support.

mots clés : Fiabilité, Sûreté de fonctionnement, Systèmes distribués de contrôle, Architecture tolérante aux fautes

1 Introduction

Les applications de contrôle-commande font partie des applications dites temps réel où le respect de propriétés temporelles est primordiale. Leurs développements nécessitent une attention particulière en raison des conséquences dramatiques que peut avoir un dysfonctionnement sur le système physique contrôlé et son environnement. Les choix de conception et d'implémentation sont donc fait de manière à garantir un niveau de fiabilité suffisant du système de contrôle. Les études usuelles de sûreté de fonctionnement considérent l'existence de défaillances permanentes de certains services et cherchent à garantir la validité de chaque information utilisée par le système de contrôle. Nous proposons dans cet article de lier l'occurence de défaillances (permanentes ou non) au niveau du système de contrôle à la défaillance du système physique.

Ce problème est complexe car il est difficile de lier explicitement les défaillances du système de contrôle à l'apparition d'une défaillance sur le système physique, la plupart des systèmes régulés (système physique + système de contrôle) ayant une certaine capacité intrinsèque de corriger des défaillances non permanentes du système de contrôle, ce qui peut diminuer les risques de défaillance globale.

Dans cette étude, nous nous plaçons plus précisément dans le cadre des systèmes échantillonnés et analysons la notion de sûreté de fonctionnement des systèmes régulés, c'est-à-dire ceux qui présentent une boucle de retour de l'état du système physique sur le système de contrôle. Nous abordons ce problème en nous appuyant sur un système élémentaire. Cet exemple, représentatif d'une large classe de systèmes, présente des particularités qui permettent d'obtenir simplement des résultats. C'est-à-dire qu'il est possible de définir une relation formelle explicite qui lie les défaillances du système de contrôle (vu ici comme un système de traitement de l'information) et l'apparition d'une défaillance du système régulé.

Après avoir précisé dans la section 2 les modèles pertinents du système régulé, nous définissons, dans la section 3, la notion de défaillance du système régulé et identifions ses relations avec la défaillance d'une information. Nous introduisons, en particulier, la notion de séquences admissibles de contrôle ainsi que les moyens de les calculer. A partir de cette étude, nous montrons, en section 4 comment calculer sur cet exemple des métriques probabilistes de fiabilité. Enfin, nous concluons sur le travail présenté.

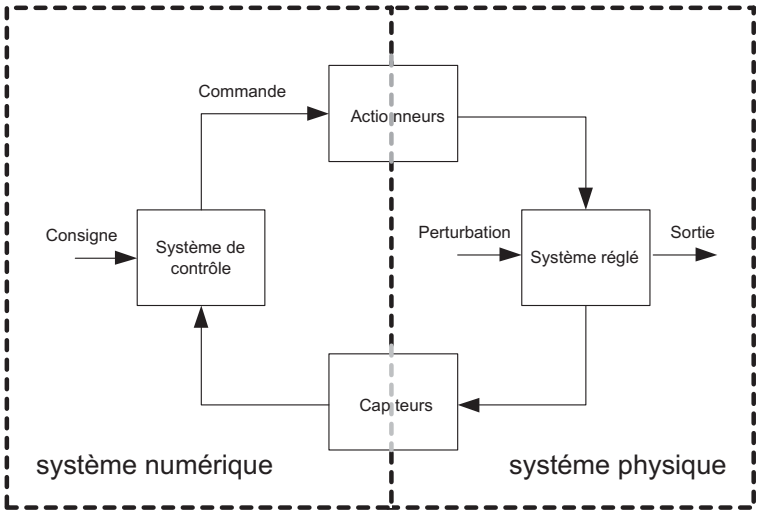

FIG. 1 – Boucle de régulation.

2 Contexte de l'étude et modèles du système

2.1 Modèle de la régulation

Une boucle de régulation se caractérise par un système de contrôle, un système physique (système réglé) à contrôler / réguler et un ensemble de capteurs et actionneurs. L'ensemble est désigné sous le terme "système régulé" (figure 1).

L'objectif de la régulation peut être d'une part, de faire suivre au système le changement de la consigne, c'est-à-dire de faire évoluer sa sortie rapidement vers la nouvelle valeur désirée avec une certaine tolérance (respect de bornes sur les dépassements) et / ou d'autre part, de permettre au système de rejeter les perturbations ; ce dernier point signifie qu'en présence d'une situation anormale et ponctuelle le système doit avoir la capacité de retrouver l'état désirée (donné par la consigne).

On s'intéresse plus particulièrement ici aux systèmes échantillonnés pour lesquels le système réglé est contrôlé par un dispositif de traitement de l'information qui met en oeuvre des activités périodiques d'observation du système et d'action sur celui-ci. Le système régulé se compose donc d'une partie physique dont l'évolution est de nature continue et d'une partie numérique, c'est-à-dire d'algorithmes et de données implantés sur une architecture matérielle (calculateurs connectés sur des réseaux). Le contrôleur lui-même est réalisé par un algorithme qui calcule la nouvelle commande en fonction des dernières valeurs mises à disposition par les capteurs. Chaque période correspond à un cycle au cours duquel une commande est élaborée à partir des différentes informations récupérées pendant le cycle ou auparavant.

2.2 Etats du système - notion de défaillance

Fonctionnement normal. L'évolution du système est caractérisée par des grandeurs, comme par exemple une pression, une vitesse, une quantité de liquide dans une cuve. Le but du système de contrôle est de "commander" ces grandeurs c'est-à-dire, à tout instant de garantir une valeur souhaitée (ou consigne) de ces grandeurs. En particulier, on peut ne commander qu'une grandeur, et définir la notion de fonctionnement normal du système comme étant le maintien de cette grandeur à une valeur constante.

Défaillance. Le système physique présente cependant des contraintes sur les valeurs des grandeurs. Ces contraintes s'expriment, par exemple, sous forme de bornes qu'une grandeur ne doit pas dépasser sous peine d'entraîner de graves problème de sécurité (destruction de tout ou partie du système physique, atteinte à l'intégrité des hommes et / ou de l'environnement).

Le schéma (figure 2) introduit, sur l'exemple d'une consigne constante, les différents états du système :

- l'état "en fonctionnement normal" est celui pour lequel le système oscille autour de la valeur de la consigne avec une précision donnée (notée ε sur la figure 2),
- dans l'état "en fonctionnement perturbé", le système de contrôle a la capacité de corriger l'erreur, dues aux perturbations non maîtrisées, et de ramener le système autour de la valeur de la consigne.

Lorsque, en fonctionnement perturbé, la valeur de la caractéristique atteint une borne non acceptable C_{max}(par exemple, le niveau de liquide dans une cuve dépasse la hauteur de la cuve), on parlera d'une “occurrence d'une défaillance" en ce point (t_D est la date de cette occurrence). Une analyse du système physique et de sa dynamique peut permettre de fixer une valeur limite dont on sait que, une fois cette valeur atteinte par la caractéristique, le système ne pourra éviter l'occurrence de la défaillance. Nous désignons la date où la caractéristique atteint ce point par "point de non retour" (t_l).

2.3 Modélisation du système de contrôle (système de traitement de l'information)

Fonctionnellement (figure 3), on peut modéliser le système de commande sous la forme d'une chaîne de traitement de l'information (flux de données) qui part des fonctions d'acquisition vers les fonctions d'actionnement. Cette chaîne représente le traitement logique que subit l'information dans l'élaboration et l'application de la commande par le système de contrôle. Sur ce schéma, a aussi été indiqué le fait que les différentes fonctions (en particulier celles réalisant les lois de commande) ainsi que leur propre contrôle (sous forme de règles d'activa-

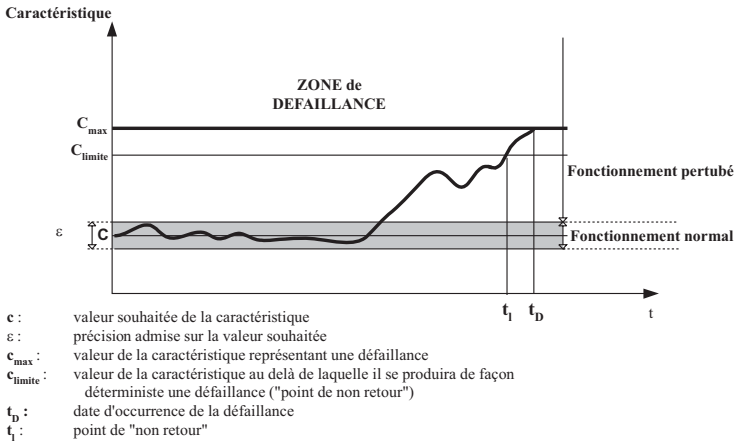

FIG. 2 – Etats d'un système.

tion) ont été spécifiés à partir d'un modèle du système à réguler. La conception finale du système de commande consiste à spécifier la distribution de l'architecture fonctionnelle sur une architecture matérielle support (distribution des fonctions et des flux de données et définition des mécanismes d'implantation des règles d'activation). C'est ce que, par la suite, nous désignons sous le terme d'architecture opérationnelle.

2.4 Causalité des défaillances

Dans cette étude, nous faisons deux hypothèses. La première considère qu'il n'y a pas de causes de défaillances du système physique autre que celles provenant du système de contrôle. La deuxième spécifie que les lois de commande sont correctes et pertinentes pour le contrôle du système physique en l'absence de défaillance du système de contrôle. Nous étudions, alors, les causes de défaillances du système physique dans la chaîne de traitement de l'information. La figure 4 donne une représentation schématique des liens entre faute à un niveau de service et défaillance au niveau du système global. Ce schéma ne fait pas apparaître les fautes latentes de conception (erreur sur un mot mémoire, erreur de codage, erreur de dimensionnement, ...). A chaque niveau "de service", nous faisons apparaître dans ce tableau, quelques mécanismes qui peuvent être mis en place pour éviter la propagation de la faute initiale; il s'agit, par exemple d'une prévention de fautes par utilisation d'un médium de communication tolérant de fortes perturbations électromagnétiques, d'une redondance active de services avec vote majoritaire, d'un sur-échantillonnage des données, ...

Nous proposons par la suite une méthode qui connaissant la loi d'apparition d'une faute dont la conséquence est une "défaillance" sur une information per-

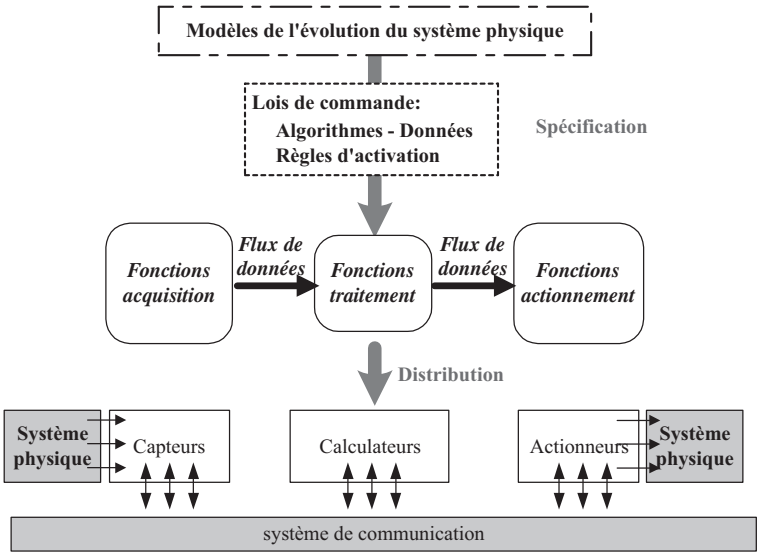

FIG. 3 – Modélisation fonctionnelle d'un système de commande.

met d'évaluer le risque d'avoir une défaillance au niveau du système global. Le terme "défaillance" sur une information exprime le fait que la valeur de l'information consommée à un instant par une activité n'est pas cohérente avec l'état instantané du système. Par exemple, dans le cas des systèmes échantillonnés, pour un échantillon donné, une telle défaillance peut provenir d'une défaillance d'un capteur (information altérée voir non produite), d'un processeur indisponible (information non produite "à temps") ou d'une faute de transmission de l'information (information altérée ou information transmise "trop tard"). La "défaillance" de l'information de commande peut alors éloigner le système de son état normal désiré et, compte tenu des lois de commandes implantées de le conduire à la défaillance du système global.

La méthode que nous développons dans la suite de cet article, montrent l'influence sur les risques de défaillances que peuvent avoir les choix les choix :

- de support matériel (avec ou sans redondance),
- de la distribution de la même architecture fonctionnelle sur une architecture support,
- des mécanismes supportant les flux d'information (par exemple avec l'introduction de mémoires partagées ou de files),
- de l'implantation des mécanismes implantant les règles de contrôle des activités (périodiques ou sur événement),

De plus, nous évaluons ce même risque en présence des solutions de tolérances aux fautes spécifiées au niveau de l'architecture fonctionnelle (comme le sur-échantillonnage).

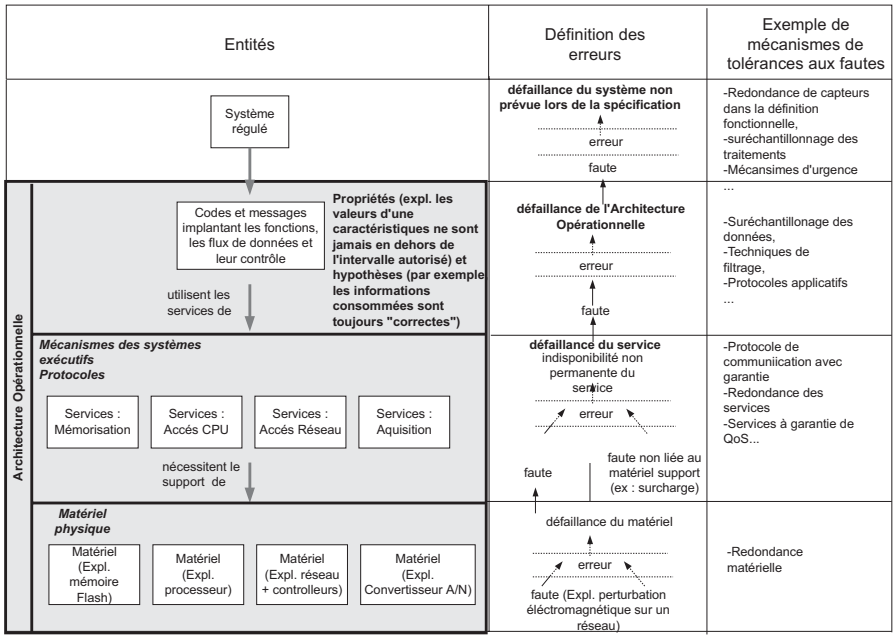

FIG. 4 – Causalité des défaillances.

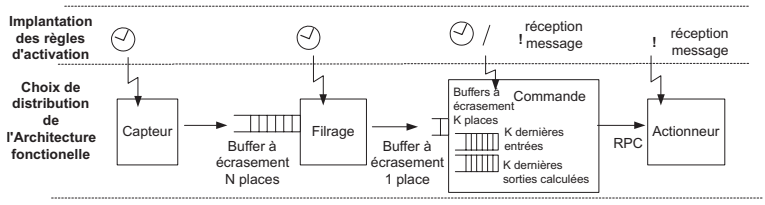

FIG. 5 – Exemple d'architecture opérationnelle détaillée.

3 Etude de cas : " système de contrôle de niveau de liquide dans une cuve "

3.1 Présentation du système

Ce système très simple a pour objectif d'introduire clairement les concepts utilisés. Il est constitué d'une cuve de hauteur H_{max}. L'ensemble du dispositif est représenté à la figure 6. L'objectif du système de commande est de maintenir le niveau du liquide dans la cuve à une valeur donnée H_0. La cuve sert à des opérations annexes qui sont ici considérées comme des perturbations, c'est-à-dire comme des événements non contrôlables (noté "perturbation" sur le schéma 6).

Le liquide est incompressible, le modèle de la cuve considéré est un intégrateur, c'est-à-dire que le niveau d'eau dans la cuve à chaque instant est proportionnel à la quantité de liquide entrant moins la quantité de liquide sortant.

La quantité de liquide est proportionnelle à la hauteur du liquide mesurée par un capteur de niveau.

L'action sur la cuve est faite par un système composé d'une vanne et d'une pompe. La loi de contrôle est de type " plus ou moins à seuil ". Le correcteur utilise l'écart entre la consigne et le niveau actuel pour décider de la valeur de la commande à appliquer. A chaque période d'échantillonnage, on a trois cas possibles :

- aucune action dans le cas où le niveau est égal à H_0
- commande de vidange : dans le cas où le niveau est supérieur à H_0 ; on considère que la pompe retire une quantité de liquide Q_v pendant la période d'échantillonnage δt.
- commande de remplissage : dans le cas où le niveau est inférieur à H_0 ; on considère que la pompe introduit une quantité de liquide Q_R pendant la période d'échantillonnage δt.

On suppose que $Q_v < Q_R$. On travaille par la suite avec les hauteurs de liquides H_v et H_R correspondantes à Q_v et Q_R.

La seule défaillance dû système physique que nous considérons est le débordement de la cuve, c'est-à-dire que le niveau de liquide dans la cuve H dépasse un niveau H_{max}.

3.2 Choix d'une architecture opérationnelle

L'élaboration de la commande peut être vue comme une chaîne de traitement de l'information constituée de 3 fonctions :

- une fonction acquisition de l'état du système grâce à un capteur,

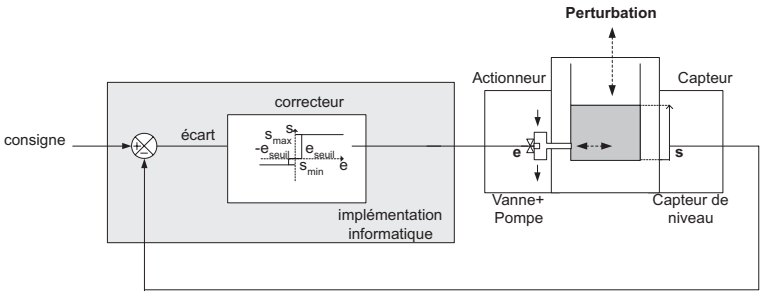

FIG. 6 – Système de contrôle de niveau dans une cuve.

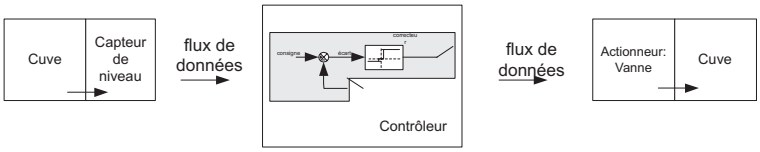

FIG. 7 – Chaîne de l'information.

- une fonction d'élaboration de la commande en fonction de la loi de commande choisie et de l'état du système à contrôler,
- une fonction actionnement qui modifie la dynamique du système par la commande d'un actionneur (vidange / remplissage / aucune action).

L'information représente les flux de données entre ses différentes fonctions. Le schéma 7 montre cette chaîne de l'information dans le cas de notre régulation de niveau.

3.3 Défaillance de l'information de commande vs défaillance du système.

3.3.1 Modèle de la défaillance sur l'information

On considère pour l'étude que les perturbations sont nulles ; en effet l'existence des perturbations peut être intégrée dans la définition de la limite avant défaillance présentée au paragraphe 2.2 et n'intervient pas directement dans l'étude. A tout instant, nous nous placerons dans le pire cas.

La seule défaillance considérée est le débordement de la cuve et, pour cette présentation, on ne s'intéresse qu'aux défaillances de l'information de commande qui augmentent le niveau de liquide. Il existe deux cas à considérer :

- la commande demande un remplissage alors qu'il faudrait vidanger ou ne rien faire,
- la commande demande de "ne rien faire" alors qu'il faudrait vidanger.

Nous considérons le pire cas c'est-à-dire le premier cas. La seule faute considérée est donc une commande de remplissage alors qu'il faudrait vidanger ou ne rien

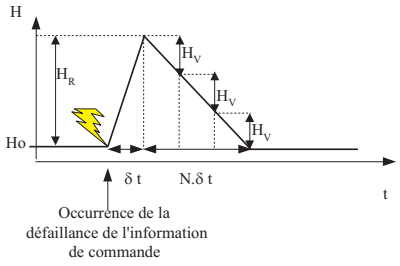

FIG. 8 – Réponse à une "défaillance de l'information de commande "

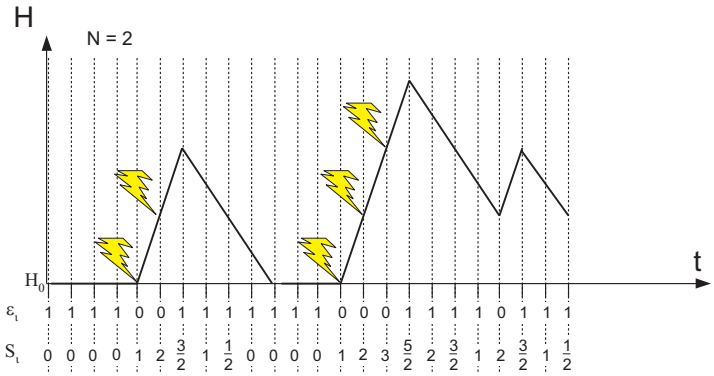

FIG. 9 – Réponse du système à une séquence d'ordres contenant des défaillances.

faire.

3.3.2 Réponse du système à une défaillance non permanente de l'information.

Le système de contrôle ayant pour rôle de maintenir la hauteur de liquide au niveau H_0 doit appliquer des ordres de vidange. Comme la capacité de vidange est plus faible que la capacité de remplissage, le système requiert N pas d'échantillonnage pour retrouver sa position d'équilibre H_0 (où $N = [H_R/H_V]$). La figure 8 représente la réponse du système à une défaillance de la commande.

3.3.3 Défaillance du système

Nombre de défaillances admissibles consécutivement. Dans le cas du système considéré (un intégrateur), on obtient facilement la réponse du système pour Nc défaillances consécutives, le temps avant retour à l'équilibre est $N.Nc.\delta t$ et la hauteur maximale atteinte après la dernière erreur est de $Nc.H_R + H_0$.

Le nombre de défaillances consécutives limites avant défaillance du système physique (c'est-à-dire débordement, dans ce cas) est $N_{\text{ecl}} = (H_{max} - H_0)/H_R$.

Evolution admissible du système en présence de "défaillances de l'information de commande". On cherche maintenant à caractériser toutes les séquences de commandes qui mènent à une défaillance du système physique. On définit pour cela une fonction qui lie le niveau atteint à la séquence de commandes .

Définition d'une séquence. Nous considérons toujours un seul type d'erreur : un ordre de remplissage non attendu. Soit ε_i l'état de la commande pour l'échantillon (ou le pas) n°i : s'il y a défaillance de l'information de commande $\varepsilon_i = 0$ sinon $\varepsilon_i = 1$

Une séquence est une suite $(\varepsilon_i, \varepsilon_{i+1}...\varepsilon_j, \varepsilon_{j+1})$ d'informations de commande relatives aux échantillons $i, i+1, ..., j, j+1$.

La suite (11011111111) correspond à une séquence de 2 échantillons sans erreurs, 1 défaillance de l'information de commande, et 9 échantillons sans erreurs.

Réponse du système à une séquence. Le niveau H_i après i échantillons est donné par :

$$H_i = S_i.H_R$$

où

$$Si = S_{i-1} - \frac{1}{N}\varepsilon_i\delta(S_{i-1}) + (1 - \varepsilon_i)$$

avec

$$\begin{aligned} \delta(x) &= 0 \,\mathrm{si}\, x = 0 \\ &= 1 \,\mathrm{si}\, x \neq 0 \end{aligned}$$

S_i représente une mesure de l'écart à l'équilibre H_0 du à la présence d'une ou plusieurs défaillance de l'information de commande dans une séquence. On peut considérer que S_i caractérise, à tout instant, " l'état d'erreur " du système (voir figure 9). Dans l'exemple, une défaillance de l'information de commande fait monter l'état d'erreur de 1 et il faut N pas d'échantillonnage sans défaillance pour revenir à l'équilibre. La défaillance du système globale correspondant à $H > H_{max}$ apparaît pour $Si > Smax$. avec $Smax = \frac{H_{max}}{H_r}$

4 Etude de fiabilité

La connaissance de l'architecture informatique, associée à un modèle d'apparition des fautes, peut être utilisée pour évaluer les risques de défaillances. Sur l'exemple de la cuve d'eau, nous allons estimer la fréquence de dépassement du

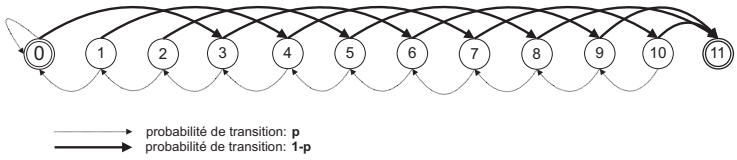

FIG. 10 – Diagramme de transition où 0 est l'état initial et 11 est l'état absorbant.

niveau maximal acceptable S_{max} dans la cuve. Considérons le cas dans lequel chaque consigne a la même probabilité $1-p$ d'être erronée (et donc p d'être correcte) avec les paramètres de la cuve $N = 3$ (nombre de consignes sans erreur pour revenir à l'équilibre) et $S_{max} \sim 11$ ($Smax = 3 + \frac{2}{3} \sim 11$ pas avec le niveau de discrétisation choisi). Une consigne est erronée lorsque le message qui la contient est corrompu et qu'il passe au travers des mécanismes protocolaires ou lorsque le message est détecté corrompu et qu'il ne parvient pas à l'actionneur. Dans ce dernier cas, nous considérons la situation la plus défavorable qui est que lorsque la valeur est manquante, une consigne erronée est toujours appliquée. Dans ce cas, nous considérons que sa valeur représente le "pire cas" et nous l'assimilons à une consigne erronée.

4.1 Modélisation

L'évolution du niveau de la cuve d'eau (S_i) peut être modélisée par une chaîne de Markov, son espace d'état étant discret et toute l'information importante pour l'évolution future de la cuve étant contenue dans la valeur de l'état courant. Nous choisissons cette chaîne en temps discret avec comme instant d'observation les dates d'application successives de la consigne. Les états de la chaîne sont les différents niveaux possibles de la cuve, dans notre exemple de 0, le point d'équilibre, à 11, le point de défaillance. Cet état particulier est modélisé par un état absorbant qu'il est impossible de quitter une fois atteint ce qui correspond au fonctionnement réel du système.

La règle générale de transition entre états est qu'en cas d'erreurs, le niveau de la cuve monte de trois unités alors qu'une consigne correcte réduit le niveau de la cuve d'une unité. Les transitions sont représentées sur la figure 10.

La matrice de transitions de la chaîne de Markov est :

$P =$

$P_{i,j}$	0	1	2	3	4	5	..	8	9	10	11
0	p	0	0	$1-p$	0	0	..	0	0	0	0
1	p	0	0	0	$1-p$	0	..	0	0	0	0
2	0	p	0	0	0	$1-p$	..	0	0	0	0
.	.	.	.	.	.	.	..	.	.	.	.
8	0	0	0	0	0	0	..	0	0	0	$1-p$
9	0	0	0	0	0	0	..	p	0	0	$1-p$
10	0	0	0	0	0	0	..	0	p	0	$1-p$
11	0	0	0	0	0	0	..	0	0	0	1

A partir de cette matrice, il est possible d'effectuer un certain nombre de calcul nous renseignant sur la fiabilité du système. En particulier, P^n nous donne la probabilité d'être dans l'état défaillance après n consignes en partant de l'état d'équilibre 0. Il est également possible de calculer le temps moyen pour atteindre l'état de défaillance ainsi que sa variance. On note N_i la variable aléatoire qui donne le premier temps d'atteinte de l'état de défaillance en partant de n'importe quel état i. L'ensemble $\mathcal{T}$ est l'ensemble des états transients (tous sauf l'état absorbant). A l'aide d'une analyse classique en "un pas", on obtient :

$$N_i = \begin{cases} \gamma_i + N_j, & \text{avec probabilité } \sum_{j\in\mathcal{T}} P_{i,j}, \\ \gamma_i, & \text{avec probabilité } P_{i,11} \end{cases}$$

avec $\gamma_i = 1$ si $i \neq 11$ et 0 sinon. On peut en dériver l'espérance de N_i :

$$\begin{aligned} E[N_i] &= P_{i,11}E[\gamma_i] + \sum_{j\in\mathcal{T}} P_{i,j}E[\gamma_i + N_j] \\ &= \gamma_i + \sum_{j\in\mathcal{T}} P_{i,j}E[N_j] \end{aligned}$$

Cet ensemble de 12 équations linéaires se résout aisément à l'aide du logiciel Maple. Comme le système est remis en marche au niveau 0 après une défaillance, $E[N_0]$ est le temps moyen d'atteinte de la défaillance en partant de l'état initial.

D'une façon similaire, il est possible de calculer la variance des temps d'atteinte de l'état de défaillance qui est par définition $V[N_i] = E[N_i^2] - E[N_i]^2$. On a :

$$N_i^2 = \begin{cases} (\gamma_i + N_j)^2, & \text{avec probabilité } \sum_{j\in\mathcal{T}} P_{i,j}, \\ \gamma_i^2, & \text{avec probabilité } P_{i,11} \end{cases}$$

L'espérance de N_i^2 est :

$$
\begin{aligned}
E[N_i^2] &= P_{i,11}E[\gamma_i^2] + \sum_{j\in\mathcal{T}} P_{i,j}E[(\gamma_i + N_j)^2] \\
&= \gamma_i + \sum_{j\in\mathcal{T}} P_{i,j}E[N_j^2] + 2\sum_{j\in\mathcal{T}} P_{i,j}E[N_j]\gamma_i
\end{aligned}
$$

Après résolution de cet ensemble d'équations, la variance des temps d'atteinte de l'état de défaillance est $V[N_0] = E[N_0^2] - E[N_0]^2$.

4.2 Fiabilité de diverses architectures

Nous allons maintenant mesurer l'influence de choix de conception sur la fiabilité du système (la fiabilité correspond à l'aptitude d'un système à accomplir sa fonction). Cinq architectures sont évaluées :

- l'architecture 1 est le système tel que décrit au paragraphe 3.1 avec $N = 3$, $S_{max} = 11$ et une probabilité d'erreur sur la consigne de 5% ($p = 0.95$). La fréquence des consignes est de une par minute,
- l'architecture 2 est identique au cas 1 mais la probabilité d'erreur est divisée par 5 ($p = 0.99$) ce qui correspond par exemple à l'utilisation d'un support de transmission blindé (cf. [1] pour des mesures sur la robustesse relative de différents supports),
- l'architecture 3 est identique au cas 1 mais la fréquence des consignes est doublée (toutes les 30 secondes). La quantité d'eau injectée lors d'un ordre de remplissage étant divisée par 2, cela revient à fixer $S_{max} = 22$.
- l'architecture 4 est identique au cas 1 mais une même consigne est transmise deux fois sous une hypothèse de fail-silence*[2]* : une donnée reçue est nécessairement bonne au niveau de sa valeur mais la donnée peut-être occasionnellement manquante. Une erreur se produit sur la consigne lorsque les deux messages qui la contiennent sont corrompus. Sous l'hypothèse d'erreurs indépendantes, la probabilité d'un tel événement est p^2, soit ici 0.0025 .
- l'architecture 5 est obtenue par modification de la capacité de récupération du système contrôlé, on augmente H_v (en changeant la loi de commande ou si nécessaire les actionneurs). On diminue ainsi le rapport N, on prend ici N=2 au lieu de N=3 pour l'architecture de référence 1. Il ne faut plus que 2 échantillons au système contrôlé pour corriger un ordre défaillant.

Nous avons choisi de comparer les différentes architectures vis-à-vis de deux métriques de fiabilité :

- le temps moyen d'atteinte de la défaillance (cf. 4.1),

Architecture	Moyenne du temps d'atteinte de la défaillance	Espérance du temps d'atteinte de la défaillance	Probabilité de défaillance en 1 heure	Safety Integrity Level
Cas 1 : Architecture de référence	20 jours	20 jours	0.19e-02	-
Cas 2 : La probabilité d'erreurs de transmissison diminue fortement	42 années	42 années	0.24e-05	SIL1
Cas 3 : La fréquence du contrôle est doublée	227 années	227 années	0.44e-06	SIL2
Cas 4: Architecture "fail silence"	11000 années	11000 années	0.89e-08	SIL4
Cas 5 : Modification de la capacité du systéme controlé à récupérer une erreur	11000 années	13 années	0.68e-05	SIL1

FIG. 11 – Tableau comparatif de la fiabilité obtenue pour les différentes architectures.

– la probabilité de défaillance en 1 heure, qui sert de référence pour la définition des niveaux d'intégrité SIL [5].

Les niveaux SIL on été définis pour décrire le niveau de sûreté attendu des systèmes de sécurité, ils peuvent cependant , à titre comparatif, servir de référence pour décrire la sûreté du système global.

La première architecture sert de référence et ne possède aucun mécanisme de tolérances aux fautes. Les architectures 2, 3 et 4 peuvent être vues comme une extension de l'architecture 1 avec la mise en oeuvre d'une technique de tolérance aux fautes (cf. figure 4).

Les différentes techniques apportent un gain certain au niveau de la fiabilité du système global. On remarque en particulier que les solutions qui prennent en compte la possibilité de défaillances au niveau de la conception de l'architecture fonctionnelle et/ou opérationnelle (sur-échantillonnage et garantie de "fail silence") donnent de meilleurs résultats que des solutions de bas niveaux vis-à-vis des fautes primaires (blindage du câble). Les gains obtenus en utilisant des architectures possédant des tolérances aux fautes sont très importants (passage d'un niveau SIL0 à SIL4 grâce à l'architecture "fail silence"). Des solutions simples et peu onéreuses comme le sur-échantillonnage sont également intéressantes (passage de SIL0 à SIL2).

Enfin cette étude montre que, dans le cadre des systèmes échantillonnés, de nombreux éléments interviennent dans la fiabilité du système global. Ainsi, les différentes techniques qui ont étés utilisées pour augmenter la fiabilité de

l'architecture 1 se placent à des niveaux de conception très différents :

- l'amélioration du matériel en fonction des niveaux de perturbations rencontrés (Architecture 2),
- le sur-échantillonnage (Architecture 3), qui est une solution de haut niveau et qui doit être définie dans l'Architecture Fonctionnelle,
- la mise en place d'une architecture tolérante aux fautes (Architecture 4), qui relève d'une caractéristique désirée de l'Architecture Opérationelle support,
- La définition d'une loi de commande qui prenne en compte les risques de défaillances de l'information en plus des critères habituels liés à la dynamique du système physique. (Architecture 5) .

5 Conclusion

L'étude de la fiabilité des architectures distribuées dans le cadre des systèmes échantillonnés est devenue critique, en particulier dans l'automobile en raison des problèmes posés par le X-by-Wire (contrôle des différentes fonctions comme le freinage sans liaison mécanique)[6, 7]. Cette étude montre, sur un exemple, comment lier la probabilité d'apparition d'une erreur de contrôle avec la probabilité de défaillance du système en connaissant la dynamique du système régulé et les algorithmes de contrôle. Précisément, nous avons calculé, sous certaines hypothèses de pire cas, les deux métriques de fiabilité que sont le temps moyen entre deux défaillances du système et la probabilité de défaillance en une heure de fonctionnement. De tels calculs de fiabilité sont nécessaires pour évaluer la qualité de différentes Architectures Opérationnelles en concurrence lorsque les contraintes de sûreté de fonctionnement sont importantes.

Une des causes de défaillance de l'information est une surcharge temporaire d'une des ressources du système. Dans ce cas, la fonction d'état d'erreur peut servir à mettre en place des mécanismes d'allocation des ressources en fonction de la criticité des différents traitements vis-à-vis du risque d'apparition d'une défaillance globale. Il est ainsi possible de définir un besoin sous la forme d'une fonction $m(k)$ qui définit le nombre de traitements corrects que toute séquence de k traitements consécutifs doit comporter. Cette contrainte est une extension du modèle (m, k)-firm [4] qui définit un besoin de m traitements corrects pour une valeur particulière de k. Une extension naturelle de cette étude est la définition de mécanismes exécutifs permettant le respect des contraintes $m(k)$.

Nous avons évalué des métriques de fiabilité sur un système simple qui est l'intégrateur. Il est possible de généraliser les techniques utilisées à des systèmes d'ordre supérieure au sens de l'automatique (le lecteur intéressé pourra consulter [8]). Le passage à un système d'ordre 1 ne pose pas de difficulté ; les transitions

dans la chaîne de Markov deviennent fonction de l'état courant. Pour des ordres supérieurs, il est nécessaire de prendre en compte les dérivées du système. Une solution possible est qu'un état définisse non seulement la valeur de la grandeur mais également celles de ses dérivées significatives. Une autre extension de cette étude est de prendre en compte des modèles de fautes plus complexes comme la possibilité d'occurrence de rafales d'erreurs qui peut être intégrée dans des modèles Markoviens [3].

Références

[1] J. Barrenscheen, G. Otte, *Analysis of the Physical CAN Bus Layer*. In 4th International CAN Conference, ICC'97,Octobre 1997.

[2] S. Poledna, P. Barrett, A. Burns et A. Wellings, *Replica Determinism and Flexible Scheduling in Hard Real-Time Dependable Systems*, IEEE Transactions on Computers, 49(2), 2000.

[3] N. Navet, Y-Q. Song, F. Simonot, *Worst-Case Deadline Failure Probability in Real-Time Applications Distributed over CAN (Controller Area Network)*, Journal of Systems Architecture, Elsevier Science, 46(7), 2000.

[4] P. Ramanathan, *Overload Management in Real-Time Control Applications using (m,k) Firm Guarantee*, IEEE Transaction on parallel and distributed systems, 10(6), 1999.

[5] International Electrotechnical Comission, *61508-1 : Functionnal Safety of Electrical / Electronic / Programmable Electronic Safety-Related Systems*, IEC, 1997.

[6] E. Dilger,T Füher, B. Müller,S. Poldena, *The X-by-Wire Concept : Time Triggered Information Exchange and Fail Silence Support by new System Service*, SAE Internatinal Congress and Exhibition. Detroit, 1998.

[7] Projet PALBUS SP-AR 1999 :31, *Validation of Dependable Distributed Control Systems*, `http://www.sp.se/electronics/rnd/palbus/`, 1999.

[8] F. Jumel, *Définition et Gestion d'une Qualité de Service pour les applications temps réel*, thèse INPL, à paraître juin 2003