Vers une méthodologie d'analyse pour la co-conception et la validation de systèmes commandés en réseau

Blaise Conrard, Laurent Cauffriez, Christophe Aubrun, Jean-Marc Thiriet

Vers une méthodologie d'analyse pour la co-conception et la validation de systèmes commandés en réseau

Cas-test du groupe ConecsSdF

Blaise CONRARD
LAGIS
Université Lille 1
Villeneuve d'Ascq, France
Blaise.Conrard@polytech-lille.fr

Laurent CAUFFRIEZ
LAMIH-SIC (Intelligent & Cooperative Systems)
Université de Valenciennes
Valenciennes, France
laurent.cauffriez@univ-valenciennes.fr

Christophe AUBRUN
CRAN
Université de Nancy
Nancy, France
christophe.aubrun@cran.uhp-nancy.fr

Jean-Marc THIRIET
GIPSA-Lab
Univ Grenoble Alpes, CNRS
Grenoble, France
jean-marc.thiriet@gipsa-lab.grenoble-inp.fr

***Résumé*—Cet article s'intéresse aux systèmes commandés en réseau ou NCS (Networked Control System) et à la validation formelle de leur comportement. En effet, la présence d'un réseau de communication au sein des NCS implique une validation approfondie de leur conception compte tenu de la distribution des traitements, de l'asynchronisme des horloges des instruments interconnectés et des retards variables dans l'acheminement des informations. Ces retards variables forment une nouvelle source de perturbation pour la commande du système et l'asservissement des grandeurs physiques qui le composent. Un challenge pour le concepteur d'un tel système est de montrer que le système restera, en exploitation, dans sa gamme de fonctionnement nominal en présence de perturbations internes au système ou externes induites par l'environnement. Cet article présente une telle étude et montre qu'une analyse formelle est réalisable pour des systèmes simples pour lesquels moins d'une dizaine de variables d'état sont à dénombrer.**

***Index Terms*—NCS, formal proof, safety assessment, dependability, control system design**

I. INTRODUCTION

L'utilisation de réseaux de terrain, de réseaux de capteurs, d'instruments intelligents donnent de nouvelles possibilités pour les systèmes commandés en réseau mais engendrent également de nouvelles contraintes en terme de sûreté de fonctionnement [1].

En effet, la distribution des traitements offre de nouvelles possibilités de reconfiguration et de tolérance aux fautes. Cependant, l'asynchronisme des différents calculateurs associé à des délais variables dans l'échange des informations rendent ces systèmes complexes dans la détermination de leur comportement. En dépit de cela, ces systèmes doivent satisfaire aux contraintes de sûreté et sécurité [2] [3]. Il s'agit donc de répondre à la problématique de la co-conception des «Systèmes Contrôlés en Réseaux Sûrs de fonctionnement» soit en langue anglaise «Safe-Networked Controlled Systems», dont un acronyme souvent usité est « Safe-NCS ») [4].

Dans ce papier, la conception de systèmes commandés en réseau est abordée en vue de prouver le bon comportement du système global. Cette approche prend en compte les effets induits par le réseau, ses défaillances et la distribution des traitements. En dépit de ces phénomènes, l'étude de sécurité proposée doit démontrer l'absence de risque ou en préciser les limites. Cette approche vise à donner au concepteur les moyens d'accroître la sûreté de fonctionnement de systèmes commandés en réseau.

II. METHODES D'ETUDE

A. Eléments à modéliser

La modélisation du comportement d'un système commandé en réseau est une activité relativement complexe de par la diversité des phénomènes à prendre en compte.

Pour de nombreux problèmes classiques d'automatique, il suffit de linéariser le système autour d'un point de fonctionnement pour procéder à l'étude de sa stabilité, de sa dynamique lors de petit changement de consigne ou de son comportement en présence de petites perturbations.

Cette technique n'est cependant pas adaptée à une étude du comportement global souhaité du système dans toute sa plage

de fonctionnement notamment dans le cas d'un système commandé en réseau.

D'une part, il s'agit de modéliser les non-linéarités du système. Un phénomène classique de non-linéarité est la saturation des actionneurs qui, bornés par leur puissance, ne peuvent avoir qu'une influence limitée sur le processus à commander. Un cas particulier de saturation est la non-réversibilité de certains actionneurs, par exemple un frein ou une pompe, qui ne peuvent travailler que dans un cadrant.

D'autre part, un autre phénomène classique en automatique, est l'intégration des perturbations environnementales dans le modèle. Ces perturbations environnementales, qui altèrent la qualité de commande, sont généralement modélisées par des variables aléatoires, associées souvent à un bruit blanc. A titre d'exemple, les perturbations environnementales peuvent être des bruits de mesures, une évolution de l'environnement, l'imprécision des actionneurs, des biais des instruments (capteurs ou actionneurs) ou encore la conséquence du pas de quantification des convertisseurs. En dépit de leur caractère aléatoire, ces signaux ont la particularité de rester dans des intervalles bornés, assez simples à estimer et peuvent être ainsi pris en compte dans la modélisation.

A ces perturbations externes, les ordres de l'utilisateur doivent être également intégrés au modèle. Ces ordres peuvent être assimilés à une "perturbation" de par leur caractère aléatoire, leur date d'apparition, ou leur amplitude.

Si les phénomènes précédents sont relatifs à des notions classiques de l'automatique et de la commande de processus, d'autres phénomènes sont intrinsèquement liés à l'utilisation d'une commande numérique distribuée. Un phénomène caractéristique est l'asynchronisme des différents calculateurs. En effet, chaque équipement (capteur, régulateur, actionneur) travaille avec sa propre horloge et sa propre période d'échantillonnage. Ces périodes d'échantillonnage sont par nature différentes et sont soit configurables soit fixées à la conception. Cet aspect, associé au décalage provoqué par l'asynchronisme des horloges, induit d'importants retards dans le délai de la chaîne de traitement des données : c'est-à-dire de l'instant de la mesure jusqu'à l'instant d'application de la commande correspondante sur l'actionneur. A noter que ces écarts ont une forte dynamique et peuvent varier fortement entre deux échantillonnages successifs.

L'emploi d'un réseau de communication amplifie ce phénomène. En effet, un certain délai s'écoule entre l'instant où un équipement a une information à transmettre et celui où le destinateur la reçoit. D'une transmission à l'autre, ce délai oscille dans un intervalle assez large dépendant du débit mais surtout du protocole employé et de la charge du réseau au moment des transmissions.

Enfin, les défaillances ou les fautes des composants doivent également être pris en compte lors de la conception. Ce peut être la perte d'une transmission, la réinitialisation intempestive d'un calculateur, la perte définitive d'un composant suite à sa défaillance intrinsèque.

Le modèle du Safe-NCS doit donc pouvoir intégrer ces événements aléatoires dont l'étude doit conduire à caractériser la tolérance aux fautes du système et à maîtriser les risques lors d'un comportement dangereux.

En conclusion, pour obtenir une modélisation suffisamment fine d'un système commandé en réseau, le modèle doit intégrer:

- les non-linéarités du processus,
- les perturbations de l'environnement,
- l'asynchronisme des calculateurs,
- les délais aléatoires de transmission,
- les possibles défaillances, fugitives ou permanentes des composants.

La sous-section suivante propose un tel modèle.

B. Modèle de base

Pour modéliser le comportement de systèmes commandés en réseau et pour y intégrer les différents phénomènes décrits précédemment, une représentation d'état semble la mieux adaptée. L'évolution du système peut ainsi être modélisée par la relation :

$x_{k+1} = f (x_k, e_k)$

où

- x est le vecteur d'état du système,
- f est la fonction qui permet d'évaluer, à partir de l'état courant, le vecteur d'état au prochain événement,
- e est le vecteur des variables exogènes.

Le vecteur d'état x du système regroupe l'ensemble minimaliste des variables du système qui permettent de décrire tout le système. Ces variables peuvent être des grandeurs physiques liées au processus (ex. : position d'un robot, niveau d'eau dans une cuve...), des variables numériques issues des algorithmes embarqués, le contenu de transmissions en attente de réception, les dates de réveil des traitements, l'arrivée d'un message en cours de transmission. On notera qu'inclure des dates dans un vecteur d'état est peu traditionnel en automatique, mais nécessaire ici pour la modélisation des phénomènes discrets.

La fonction f permet l'évaluation du vecteur d'état au prochain événement à partir de l'état courant. Ainsi, le système étudié et la représentation d'état associée peuvent être considérés comme un système hybride relevant à la fois de comportements continus et discrets. En effet, les grandeurs physiques du système suivent une évolution continue. On se propose de modéliser leur évolution sous la forme de relations algébriques liées au temps. Quant aux variables numériques du système de commande, elles changent de façon discrète suite à un événement spécifique (tel que la fin de l'exécution d'un traitement algorithmique, la prise d'une mesure ou la mémorisation d'une information reçue via le réseau). Ces changements de valeur peuvent être également représentés par une relation algébrique (éventuellement par un algorithme pour des opérations plus complexes), mais leur application et leur évaluation sont essentiellement liées à l'atteinte d'une date de réveil ou de déclenchement fourni par le vecteur d'état. Cette

date doit alors être mise à jour pour indiquer le prochain réveil du traitement.

Le vecteur e des variables exogènes modélise les perturbations, les délais de transmission, les ordres de l'utilisateur. Ce sont des variables aléatoires dont la distribution ou tout au moins les bornes sont supposées être connues. Si la plupart de ces variables sont de nature continue afin de modéliser des retards variables, des bruits de mesures..., certaines peuvent être aussi discrètes pour la modélisation par exemple de la perte d'un message, de la défaillance d'un équipement. Dans ce cas, la variable discrète se réduit à une variable booléenne.

C. Définition de l'analyse formelle proposée

A partir d'un tel modèle, et par simulation, une analyse statistique permet d'obtenir une distribution des valeurs les plus probables des grandeurs qui régissent le système. Cependant, cette analyse ne permet pas de prouver que des états redoutés ne sont pas atteignables. Or, cette preuve est essentielle pour une étude de sécurité. L'analyse formelle proposée tend ainsi à répondre à cette problématique. Elle vise ainsi à déterminer les frontières de l'espace des valeurs accessibles par les grandeurs du système et permet donc de vérifier les propriétés définies dans le cahier des charges défini par le maître d'ouvrage. En outre, cette démarche a pour but de vérifier que l'évolution de certaines grandeurs reste dans une région non-dangereuse dans le cas d'une étude de sécurité.

Plus précisément, la démarche d'analyse formelle proposée est itérative et recherche l'ensemble des états accessibles. Elle débute d'un état initial, et elle détermine, par pas successifs, les états atteignables quelles que soient les valeurs possibles du vecteur de variables exogènes. L'opération est ainsi renouvelée jusqu'à atteindre les frontières de l'espace accessible dans sa globalité. Il est à noter que cet espace fini n'existe que si le système étudié est stable.

III. Principe d'une analyse formelle pour un NCS

Cette section s'intéresse aux différents éléments pour mener une analyse formelle s'appuyant sur la modélisation d'un système commandé en réseau à base d'un modèle d'état.

A. Objectif et principe

L'approche utilisée pour déterminer l'espace des états accessibles par le système étudié requiert un état initial (ou un sous-espace initial). A partir de ce dernier, on détermine, en fonction des valeurs possibles des variables aléatoires, le sous-espace atteint au pas suivant [6]. A partir du sous-espace précédemment obtenu, on renouvelle l'opération pour obtenir un espace plus grand. Après une "infinité" de pas (cette notion est discutée après), l'espace atteignable et ses frontières sont alors déterminés, sous réserve que le système soit stable.

La validation du système étudié consiste ainsi à vérifier que le système satisfait un ensemble de contraintes (Cf. figure 1). Ces contraintes sont exprimées sous la forme de valeurs limites qui ne doivent pas pouvoir être dépassées par le système. Comme l'illustre la figure 1, le système est validé si l'espace des états atteignables reste bien dans les frontières fixées par les contraintes définies dans le cahier des charges.

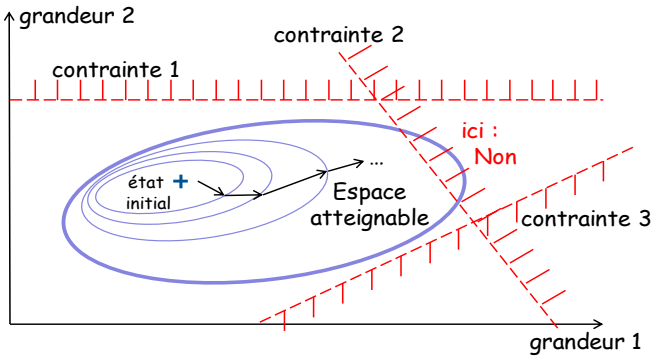

Fig. 1. Espace des états atteignables et satisfaction de contraintes

B. Problème des variables continues

Un des principaux obstacles à la méthode proposée réside dans le fait que les frontières de l'espace atteignable ne sont pas accessibles en un nombre fini de pas. Par exemple la fonction $X_{t+1} = X_t/2$ avec $X_0=1$, tend vers 0 sans jamais l'atteindre. L'informatisation de la démarche de recherche de l'espace atteignable n'aboutit donc pas à une solution acceptable dans la mesure où les frontières continuent de croître à chaque pas.

Pour résoudre ce problème, une discrétisation des valeurs continues s'est avérée nécessaire. Ainsi, lors de l'évaluation d'un état accessible, celui-ci est "arrondi" à un carré (un cube ou un hyper-cube, selon la dimension du vecteur d'état) où ses sommets sont les valeurs discrétisées des variables continues. Cette manière de procéder permet d'accroitre la vitesse de convergence vers les frontières de l'espace atteignables, codées sous la forme de sommets. Cependant, cette approximation à l'aide de pas de quantification conduit à une perte en précision et fournit une frontière plus grande de l'espace atteignable que la réalité. La figure 2 illustre cette technique.

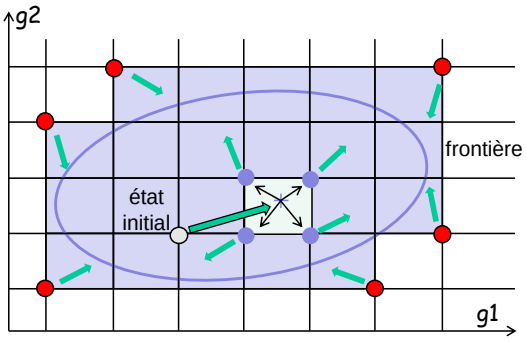

Fig. 2. Discrétisation de l'espace et exploration de celui-ci

C. Algorithme et outil d'analyse

Concernant l'informatisation de la méthode d'analyse, en l'absence d'outil connu de notre part, notre propre logiciel a été développé. Ecrit en C, pour optimiser sa rapidité d'exécution, il est basé sur l'algorithme suivant.

```
Etats ← x_initial / marqué "non-traité"
Tant que ∃ x ∈ Etats avec x marqué "non-traité"
    Pour chaque e_i ∈ Perturbation
        x_K+1 ← f ( x, e_i )
        S ← ens. des sommets-état discrétisés de x_K+1
                marqué "non-traité"
        Etats ← Etats ∪ S
    Marquer x comme "traité"
    Nettoyage Etats
            (= retrait des états à l'intérieur des frontières)
Retourne Etat
```

Dans son principe, l'ensemble *Etats* contient les états accessibles du système. Il est initialisé par un état initial. Les états sont marqués comme "traité" ou "non traité". Pour chaque état "non-traité", l'algorithme détermine à partir du modèle, les états accessibles (en un pas) et marque ensuite cet état comme "traité". La détermination de ces états accessibles est effectuée à partir de l'état courant **x**, du modèle **f** et des perturbations possibles $\mathbf{e_i}$ extraites d'un ensemble *Perturbation* fournissant différentes combinaisons des valeurs extrêmes de ces perturbations. Pour chaque nouvel état accessible $\mathbf{x_{K+1}}$, celui-ci est discrétisé en sommets ; ceux-ci sont alors inclus dans *Etats* comme des états accessibles à traiter. A chaque pas, afin d'éviter l'explosion combinatoire, un nettoyage de l'ensemble *Etats* est effectué pour ne retenir que les états de la frontière de l'espace accessible courant et retirer ceux à l'intérieur de cette espace. Lorsque tous les états ont été traités, les frontières extrêmes de l'espace accessible sont atteintes et cet espace final décrit par *Etats* est retourné sous la forme d'un ensemble de sommets.

D. Limites à l'approche proposée

La méthode proposée présente cependant deux limites exposées ci-après.

a) La première concerne l'explosion combinatoire. En effet, l'augmentation du nombre de sommets pour représenter l'espace accessible croît de façon exponentielle avec le nombre de paramètres du vecteur d'état. Ainsi, s'il faut au minimum 8 sommets pour définir un espace à 3 variables, il en faut 1024 ($=2^{10}$) pour des états à 10 paramètres et 10^6 sommets pour 20 paramètres. A ceci, une forme "arrondie" (et non carré) de ces espaces accroit encore plus le nombre de sommets et conduit à des temps de calcul rapidement rédhibitoires pour les systèmes informatiques actuels.

b) le choix des pas de discrétisation (ou de quantification) est délicat. Trop petit, ce pas accentue l'explosion combinatoire concernant le nombre de sommets pour décrire l'espace atteignable. Trop grand, il aboutit à une perte de précision trop importante sur l'obtention des frontières de l'espace.

La section suivante applique cette méthode à un cas d'étude.

IV. APPLICATION A L'ASSERVISSEMENT DE NIVEAU D'UNE CUVE

Cette section applique la méthodologie d'analyse formelle proposée sur un cas d'étude.

A. Présentation générale

Le système choisi est classique dans le domaine de la commande. Il concerne l'asservissement de niveau d'une cuve où un utilisateur externe prélève librement une quantité de fluide. Le système de commande implanté doit éviter l'événement redouté « cuve vidée » et, plus grave, l'événement redouté « débordement de la cuve ».

Concernant cette commande, deux instrument sont utilisés : un capteur de niveau et une pompe, tous deux intelligents c'est-à-dire pilotés par un microcontrôleur et communiquant via un réseau. Avec un réseau à accès aléatoire (tel que CAN, CSMA/CD...) [5] interconnectant d'autres équipements du site de production, l'acheminement des informations va être soumis à des retards variables. Ainsi, selon la charge du réseau et les besoins en communication, les messages échangés entre le capteur de niveau et la pompe vont être plus ou moins retardés. On suppose, cependant, que le délai de transmission reste borné et ne dépasse pas une période d'échantillonnage.

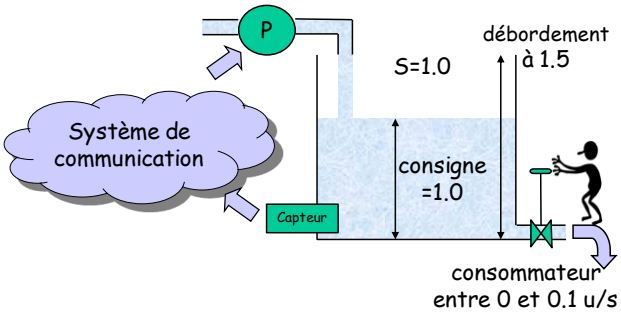

Fig. 3. Asservissement distribué de niveau d'une cuve

Les caractéristiques du processus physiques sont les suivantes :

- surface de la cuve : 1 unité (de surface)
- consigne d'utilisation ou point de fonctionnement : 1 unité (de longueur)
- hauteur de débordement : 1,5 unité (de longueur)
- période d'échantillonnage des équipements (capteur et actionneur) : 1 unité (de temps)

La perturbation environnementale choisie est une consommation variable du contenu de la cuve entre 0 et 0,1 unité (de volume), soit au maximum 1/10 du volume de la cuve consommé lors d'une période d'échantillonnage.

Pour un tel système, un correcteur de type PI (proportionnel intégral) est suffisant. Les gains du correcteur sont 0,2 pour le proportionnel, 0.01 pour l'intégrateur. Ces valeurs ont été choisies de sorte à obtenir un temps de réponse suffisant et ne pas risquer de débordement. Plus explicitement avec une cuve initialement vide, pour une consommation moyenne de l'utilisateur (0,05), le point de consigne est atteint en 10 unités de temps, tandis qu'en l'absence de consommation (le pire cas), le dépassement n'est pas suffisant pour entrainer un

débordement. A noter que pour ce dernier cas, l'erreur statique est due à la non-réversibilité de l'actionneur ; la pompe ne peut que remplir la cuve, pas la vider.

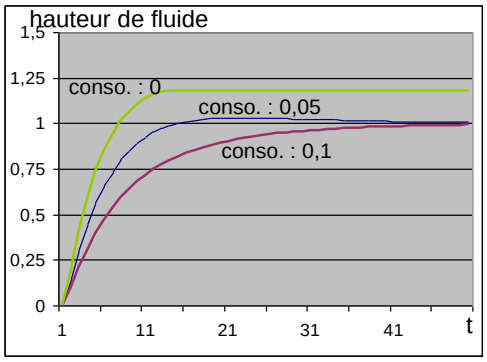

Fig. 4. dynamique du système pour différentes consommations

Enfin, le régulateur peut être aussi bien implanté sur le capteur que sur l'actionneur. Dans notre cas, on choisit la pompe. En effet, ce choix "au plus proche de l'actionneur" est plus judicieux dans la mesure où il permet d'adjoindre des mécanismes supplémentaires de sécurité tels qu'une mise en repli en cas de défaillance dangereuse détectée du réseau.

Avec ce système, l'objectif de l'étude ci-après est de déterminer si cette commande respecte les 2 propriétés suivantes et ce quelle que soit la consommation en fluide de l'utilisateur :

1) hauteur de fluide > 0 (c.-à-d. pas « l'événement redouté cuve vidée »)

2) hauteur de fluide < 1,5 (c.-à-d. pas « l'événement redouté débordement cuve »)

Dans le cas contraire, des solutions doivent être proposées.

B. Modélisation du système

Compte tenu de la description précédente du processus et des choix faits en conception, le système complet peut se modéliser par le schéma bloc fonctionnel de la figure 5.

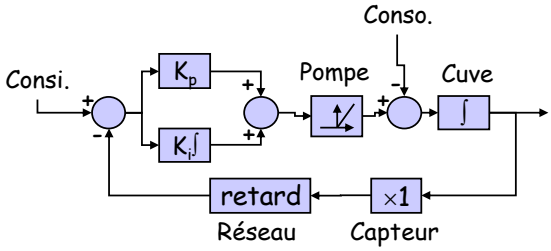

Fig. 5. Modélisation en schéma bloc fonctionnel

La modélisation sous la forme d'une représentation d'état, en tenant compte du réseau et de la distribution de la commande sur le capteur de niveau et la pompe fournit le vecteur d'état $\mathbf{x_k}$ donné par la table I, tandis que le vecteur des perturbations $\mathbf{e_k}$ est donné par la table II.

A chaque pas d'avancement du modèle, l'évaluation des paramètres de $\mathbf{x_{k+1}}$ est réalisé par les relations ci-après représentant la fonction vectorielle **f** du modèle :

$$x_{k+1} = f (x_k, e_k)$$

TABLE I. PARAMETRES DU VECTEUR D'ETAT

Nom du paramètre	Description
DateMes	Délai avant prochain réveil du capteur
DateReg	Délai avant prochain réveil du régulateur
DateArr	Délai avant arrivée du message
Niv	Niveau de la cuve
Cmd	Valeur de la dernière commande
ValInt	Valeur de l'intégrateur
MesReçu	Valeur de la dernière mesure reçue par le régulateur
MesEmis	Valeur de la dernière mesure transmise

TABLE II. PARAMETRES DU VECTEUR DES PERTURBATIONS

Nom du paramètre	Description	Intervalle
$consommation_k$	Quantité consommée par l'utilisateur	[0 ; 0,1]
$delai\_aléatoire\_de\_transmission_k$	Délai de transmission du prochain message	[0 ; 1]

Δt est la durée variable d'un pas. Il est déterminé par la date d'événement la plus proche. Ici, les 3 évènements considérés sont le réveil du calculateur du capteur (DateMes), le réveil du calculateur du régulateur-actionneur (DateReg) et l'arrivée d'une transmission à l'actionneur provenant du capteur.

Ainsi : $\Delta t = \text{minimum} (DateMes_k, DateReg_k, DateArr_k)$

L'avancement du modèle concernant les dates évènements est alors le suivant :

$$DateMes_{k+1} = DateMes_k - \Delta t$$
$$DateReg_{k+1} = DateReg_k - \Delta t$$
$$DateArr_{k+1} = DateArr_k - \Delta t$$

L'évolution de la grandeur continue, le niveau de la cuve est évaluée de la façon suivante :

$$Niv_{k+1} = Niv_k + (Cmd_k + consommation_k) \times \Delta t$$

où consommation est une valeur aléatoire de l'intervalle [0 ; 0,1], représentant la perturbation liée à la consommation variable de l'utilisateur et Cmd_k la valeur de la dernière commande reçue.

Si l'évènement associé au pas en cours est le réveil du calculateur du capteur, caractérisé par $DateMes_k = \Delta t$, entrainant la mesure du niveau de la cuve et sa transmission sur le réseau, les paramètres du vecteur d'état liés à cet évènement évoluent alors de la façon suivante :

si $DateMes_k = \Delta t$:

$MesEmis_{k+1} = Niv_k$

$DateArr_{k+1}$ = delai_aléatoire_de_tranmission$_k$
(valeur aléatoire dans l'intervalle [0 ; 1])

$DateMes_{k+1}$ = période_échantillonnage (ici 1)

De même, pour le réveil du calculateur de la pompe ($DateReg_k = \Delta t$) avec l'évaluation du correcteur et l'application de la nouvelle commande, on a :

$$ValInt_{k+1} = ValInt_k + (Consigne - MesReçu_k)$$
$$Cmd_{k+1} = Kp * (Consigne - MesReçu_k) + Ki * ValInt_k+1$$
$$DateReg_{k+1} = période\_échantillonnage$$

Enfin si l'évènement correspond à l'arrivée de la transmission de la mesure à la pompe ($DateArr_k = \Delta t$), on a :

$MesReçu_{k+1} = MesEmis_k$

$DateArr_{k+1} = \infty$

D'une manière globale, toutes ces relations fournissent l'évolution de l'état du système entre 2 évènements successifs.

C. Simplification du modèle pour l'analyse formelle

La recherche de l'ensemble des états atteignables peut être relativement longue malgré la puissance croissante des moyens informatiques. Une simplification du modèle par une réduction du nombre de variable accélèrera grandement l'évaluation des frontières de l'espace des états atteignables. Ainsi, le modèle précédent peut être réduit à 5 variables, dont 1 est transformée en une constante.

La première forme de simplification concerne la réduction du nombre d'évènements à traiter. En effet, un modèle équivalent peut être construit en alignant le pas d'évolution du modèle sur le réveil périodique du régulateur-actionneur.

En conséquence, la variable relative à la date d'arrivée d'une transmission peut être retirée. La figure 6 illustre cela, et met en évidence que le décalage des horloges entre le capteur et le régulateur-actionneur associé à des retards variables de transmission peuvent conduire à la perte de mesure et à l'utilisation d'une même mesure par le régulateur lors de 2 périodes d'échantillonnages consécutives.

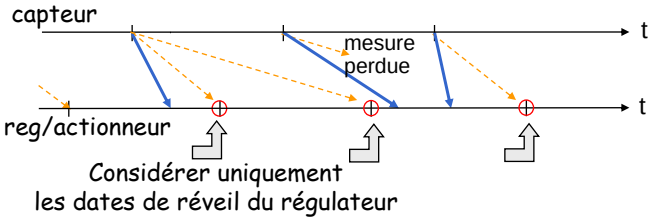

Fig. 6. Un scénario de communication avec un décalage des dates de réveil entre le capteur (source) et l'actionneur (récepteur)

Une autre simplification possible est de ne pas intégrer l'écart d'horloge entre le capteur et le régulateur-actionneur dans le vecteur d'état. Cet écart peut être alors considéré comme une constante pour lequel différentes valeurs pourront être testées.

On obtient alors le vecteur d'état suivant (cf. Table III):

TABLE III. PARAMETRES DU VECTEUR D'ETAT SIMPLIFIE

Nom	Description
Date	Délai avant prochain réveil du régulateur
Niv	Niveau de la cuve
ValInt	Valeur de l'intégrateur
DernMes	Valeur de la dernière mesure effectuée
MesReçue	Valeur de la dernière mesure reçue par le rég.

De ce vecteur d'état, à chaque pas, le système évolue de la façon suivante et définit la relation $x_{k+1} = f(x_k, e_k)$:

avec : $erreur = consigne - MesReçue_k$

$ValInt_{k+1} = ValInt_k + erreur$

$cmd = Kp \times erreur + Ki \times ValInt_{k+1}$

$Niv_{k+1} = Niv_k + cmd - consommation_k$

$nivMesure = Niv_k + \Delta h \times (cmd - consommation_k)$

$DernMes_{k+1} = nivMesure$

$MesReçue_{k+1} = nivMesure$ ou $DerMes_k$

Dans ces relations, $consommation_k$ est une valeur de l'intervalle [0 ; 0,1] et représente la perturbation, tandis que Δh est une constante représentant l'écart des horloges entre le capteur et le régulateur/actionneur.

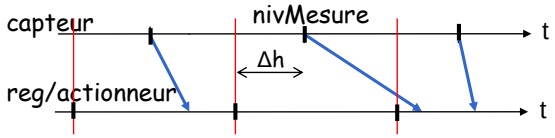

Fig. 7. Décalage d'horloge et retard de transmission dans le modèle

Avec ce modèle simplifié, une étude formelle peut être menée. Son objectif est de vérifier le respect de deux propriétés et ce quels que soient la consommation de l'utilisateur, les délais de transmission et la désynchronisation des horloges. La première propriété est l'absence de « cuve vidée » afin de vérifier le bon fonctionnement du système. La seconde s'intéresse à l'événement redouté « débordement de la cuve » pour un fonctionnement sécuritaire du système. Soit :

propriété 1 : $\forall k , Niv_k > 0$

propriété 2 : $\forall k , Niv_k < 1,5$

D. Un premier défaut : cas de l'événement redouté « cuve vidée »

L'étude de ce système montre un premier défaut du système concernant la vidange de la cuve et qui peut apparaître sous certaines conditions. Plus exactement, un scénario (tel que celui présenté à la figure 8) où l'utilisateur ne consomme pas de fluide suffisamment longtemps, alors que le niveau est au dessus de la consigne, mène à un risque de vidange.

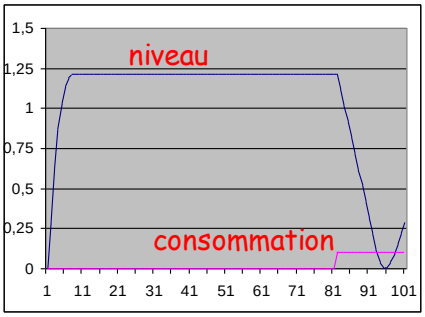

Fig. 8. Une séquence menant à la vidange

Après analyse, ce phénomène provient de l'intégrateur du régulateur PI. En effet, en raison d'un niveau trop élevé dans la cuve (ce qui correspond à une erreur négative entre consigne et mesure), l'intégrateur prend progressivement des valeurs trop basses (cf. figure 9) pour permettre le redémarrage de la pompe à la reprise de la consommation.

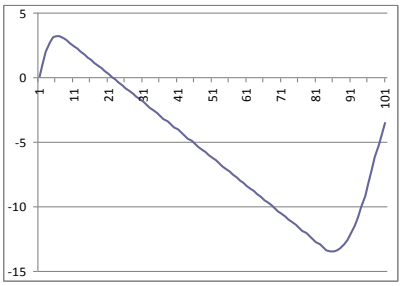

Fig. 9. Evolution de la valeur de l'intégrateur

Une solution à ce problème consiste à saturer l'intégrateur en le limitant à des valeurs uniquement positives. Dans les mêmes conditions, le niveau de la cuve reste alors toujours dans la moitié supérieure, comme le montre la simulation suivante.

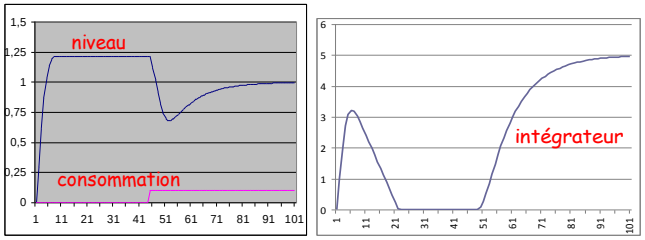

Fig. 10. Evolution du système avec anti-vidange

Avec cette correction, la fonction f du modèle formel évolue de la façon suivante :

$$ValInt_{k+1} = \min (ValInt_k + erreur , 0)$$

E. Un second défaut : l'événement redouté « débordement »

Une seconde analyse montre que la cuve peut également déborder. En effet, l'évaluation de l'espace des états atteignables donné par la table IV montre que le niveau de la cuve peut atteindre une hauteur de 1,8 unité sous l'hypothèse d'une cuve de hauteur infinie.

TABLE IV. RESULTATS DE L'ANALYSE FORMELLE

Variable	Valeur minimale	Valeur maximale	Pas de quantification
Date	1	1	0,1
Niv	0,4	1,8	0,05
ValInt	0	14,4	0,1
DernMes	0,4	1,8	0,05
MesReçue	0,4	1,8	0,05

Malgré l'imprécision liée au pas de quantification, le risque de débordement existe et la valeur de 1,8 unités trouvée ne permet pas de prouver que la hauteur de liquide dans la cuve ne peut pas atteindre le niveau de débordement (1,5).

Si l'on étudie plus précisément l'espace des états accessibles, on constate que deux facteurs principaux participent à ce risque de débordement. Le premier est lié au retard du réseau, avec lequel à un même instant, la valeur réelle du niveau de la cuve et la mesure utilisée par le régulateur peuvent être très différentes (cf. figure 11.a). Ceci peut entrainer une valeur assez importante de l'intégrateur activant la pompe et le remplissage de la cuve, alors que le niveau est assez haut (cf. figure 11.b)

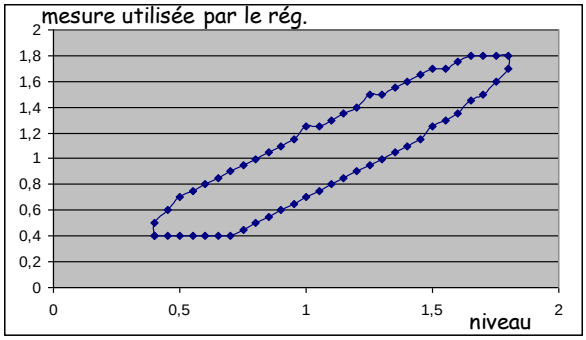

Fig. 11.a Sous-espace atteignable du vecteur d'état

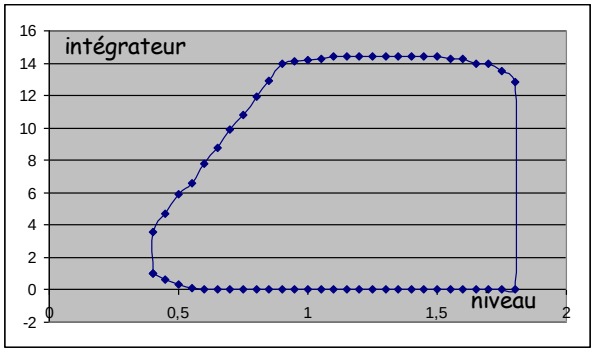

Fig. 11.b Sous-espace atteignable du vecteur d'état

Comme précédemment, une solution consiste à saturer la valeur de l'intégrateur, mais maintenant avec une valeur haute. Une première limite avec une valeur maximale de l'intégrateur à 10, fournit un résultat encore excessif avec une hauteur maximale de la cuve à 1,65. Après avoir testé différentes valeurs, saturer l'intégrateur à une valeur de 6 assure un niveau inférieur au risque de débordement. Plus précisément, on obtient les résultats de la table V.

TABLE V. RESULTATS DE L'ANALYSE FORMELLE

Variable	Valeur minimale	Valeur maximale	Pas de quantification
Date	1	1	0,1
Niv	0,45	1,4	0,025
ValInt	0	6	0,1
DernMes	0,45	1,4	0,025
MesReçue	0,45	1,4	0,025

F. Cas de la perte de message

Précédemment, les études menées considéraient un réseau avec des retards mais sans perte des messages transmis. Or, la perte de message, suite à une perturbation ou à une surcharge du réseau, n'est pas un événement rare.

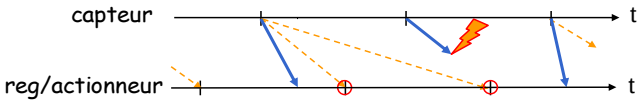

Fig. 11. Cas de la perte de message

La commande proposée précédemment utilise au niveau du régulateur-actionneur, la dernière mesure reçue et ce, quelle que soit sa date. Ainsi, en cas de perte d'un ou plusieurs messages successifs, la commande utilise une ancienne mesure qui ne correspond plus au niveau de la cuve. Ceci conduit rapidement au débordement ou à la vidange de la cuve.

Afin de gérer cette situation, on propose l'intégration d'un mécanisme de sécurité. Si entre deux échantillonnages consécutifs de la commande, aucune nouvelle mesure du niveau n'a été reçue, la valeur de celle-ci est affectée à une valeur arbitraire de 2, soit 50% plus élevé que le niveau de débordement à 1,5. La représentation du système sous la forme d'un diagramme bloc-fonction avec cette sécurité est décrite par la figure 12.

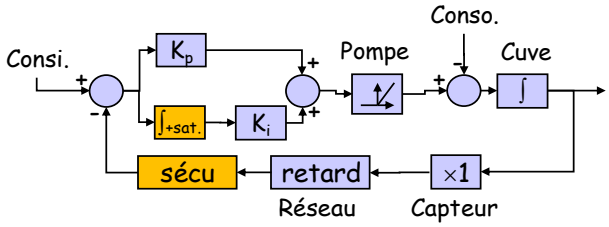

Fig. 12. Système intégrant une sécurité contre la perte de message

L'analyse de l'espace des états atteignables fournit alors les résultats de la table VI. Ils montrent que l'adjonction de ce mécanisme de sécurité interdit tout risque de débordement. Cependant, la vidange de la cuve ne peut être empêchée. En cas de perte de communication trop longue, le système se met en sécurité et selon la consommation de l'utilisateur, la cuve finit par se vider plus ou moins rapidement.

TABLE VI. RESULTATS DE L'ANALYSE FORMELLE

Variable	Valeur minimale	Valeur maximale	Pas de quantification
Date	1	1	0,1
Niv	0	1,425	0,025
ValInt	0	6	0,1
DernMes	0	2	0,025
MesReçue	0	2	0,025

G. Conclusion de l'étude

En conclusion, l'adjonction de 2 mécanismes relativement simples au niveau du régulateur permet de rendre sûr ce système d'asservissement de hauteur de fluide dans une cuve. Le premier mécanisme consiste à saturer (ou à borner) l'intégrateur par une valeur basse et une haute. Ceci évite le risque de débordement ou de vidange de la cuve vis-à-vis des perturbations internes liées aux retards de transmission du réseau ou de celles externes liées aux variations de consommation de l'utilisateur.

Le second mécanisme vise à assurer la sécurité et est l'équivalent d'un chien de garde quant à la réception des mesures par le régulateur du système. En cas de perte de communication entre le capteur et le régulateur/actionneur, le système n'a pas de comportements dangereux et évite le débordement de la cuve. Les effets de la perte de communication se limitent, au pire, à la vidange de la cuve.

V. CONCLUSION

Ce papier présente une façon d'aborder la sûreté de fonctionnement des systèmes commandés en réseau. Elle s'est attachée à montrer qu'une analyse formelle est réalisable pour éviter des comportements non désirés et dangereux. Elle considère le réseau de communication et la distribution des traitements comme une source de perturbation due aux retards de transmission et à la désynchronisation des horloges. Une des difficultés de ce type d'analyse formelle est liée aux caractéristiques de ces systèmes qui sont soumis à des perturbations très variées dans leur nombre et leur nature.

A ceci, il faut adjoindre la nature continue des variables et le risque d'explosion combinatoire qui limite ce type analyse à de petits systèmes. Cependant, l'étude menée démontre la possibilité de recourir à une méthodologie d'analyse formelle pour la co-conception et validation des systèmes commandés en réseau sûrs de fonctionnement (Safe-Network Controlled Systems).

ACKNOWLEDGMENT

Cette communication est le fruit des travaux du sous-groupe "atelier benchmark" du groupe ConecsSdF (Co-design de systèmes commandés en réseaux Sûrs de Fonctionnement) du GT ARC (GdR MACS). Les auteurs remercient, ainsi, les participants à ce groupe qui lors des réunions, par leurs remarques et leurs critiques, ont contribué à leur façon au contenu de cette publication.

REFERENCES

[1] Ghostine R., Thiriet JM, Aubry JF (2011) Variable delays and message losses: influence on the reliability of a control loop - *Reliability Engineering & System Safety* Vol 96, Issue 1 (2011) pp. 160-171.

[2] Cauffriez L., Benard V., Renaux D. (2006) A new formalism for designing and specifying RAMS parameters for complex distributed control systems: the Safe-SADT formalism, *IEEE Trans. on Reliability*. Vol. 55/3, pp.397-410.

[3] IEC 61508 (2010) *Sécurité fonctionnelle des systèmes électriques/ électroniques/ électroniques programmables relatifs à la sûreté*

[4] Aubrun C., Simon D., Song Y.Q. (2010). *Co-design Approaches for Dependable Networked Control Systems*" Ed. ISTE Wiley.

[5] Ciame (2009). *Réseaux de terrain - critères de sûreté de fonctionnement*. (Hermes-Lavoisier, Éd.) Traités IC2 (Information - Commande - Communication), série systèmes automatisés. Hermes-Lavoisier.

[6] Bhatia, A., & Frazzoli, E. (2004). *Incremental search methods for reachability analysis of continuous and hybrid systems* (pp. 142-156). Springer Berlin Heidelberg.