

DÉTECTION DE DÉFAUT À BASE DE SIMULATION : APPLICATION AUX PROCÉDÉS

N. OLIVIER-MAGET, G. HÉTREUX,
J.M. LE LANN

LGC, INPT – ENSIACET,
118 Route de Narbonne
F-31077 Toulouse Cedex 04, France
Nelly.Olivier@ensiacet.fr, Gilles.Hetreux@ensiacet.fr

M.V. LE LANN

LAAS
7 avenue du Colonel Roche
F-31077 Toulouse Cedex 04, France

RÉSUMÉ : *La complexité et la taille des procédés chimiques induisent la surveillance d'un nombre croissant de variables process. Leur connaissance est basée en général sur les mesures des variables système et des modèles physico-chimiques du procédé. Néanmoins ces informations sont imprécises à cause des bruits des mesures et de l'incertitude des modèles. C'est pourquoi les recherches visent à développer de nouvelles techniques de détection de fautes plus performantes. Dans ce travail, nous présentons une méthode basée sur la comparaison entre le système réel et le modèle de référence, utilisant un filtre de Kalman étendu. Le modèle de référence est simulé par le simulateur dynamique hybride, ProDHYS. Cette méthode est appliquée au travers de la simulation d'un exemple didactique.*

MOTS-CLÉS : *Détection de faute, Filtre de Kalman Étendu, Simulation Dynamique Hybride, Réseaux de Petri différentiels objets*

1. INTRODUCTION

Dans un contexte économique très concurrentiel, la flexibilité du système de production peut s'avérer être un avantage décisif. Généralement, cette flexibilité passe par la recherche d'une plus grande réactivité face à une demande fluctuante, mais aussi face aux nombreux aléas survenant lors de la fabrication. Dans un tel cadre, une simple défaillance est considérée comme nuisible. C'est pourquoi, le diagnostic de défauts fait l'objet d'un engouement particulier auprès de la communauté scientifique et des industriels. La volonté majeure est de ne plus subir les pannes mais de les maîtriser. Aujourd'hui, le diagnostic de défauts reste un vaste thème de recherche. La littérature cite de nombreuses méthodes de diagnostic et autant de domaines d'applications (Venkatasubramanian, et al., 2003). En général les méthodes de détection et diagnostic sont classifiées de la façon suivante :

- les méthodes à base de modèles quantitatifs telles que la redondance analytique (Chow, et al., 1984, Franck,1990), l'espace de parité (Gertler, et al., 2003,), l'estimation d'état (Willsky, 1976) ou l'utilisation de filtre (Chen, et al., 2003, Venkatasubramanian, et al., 2003),
- les méthodes à base de modèles qualitatifs telles que les méthodes causales : digraphes (Shi, et al., 1995a,b) ou arbre de défaillance (Venkatasubramanian, et al., 2003),
- et les méthodes sans modèles telles que les systèmes experts (Venkatasubramanian, et al.,

2003) et les classificateurs statistiques (Anderson, 1984)

Pour notre part, les systèmes de production visés sont les procédés batch ou semi-continus. Ces unités marient des séries d'opérations continues (qu'il s'agisse d'un régime transitoire ou permanent) avec des opérations discontinues. Constituées de nombreux appareils de traitement interconnectés et partagés, ce mode de production mixte implique une conduite événementielle de la recette et lui confère une grande dépendance vis-à-vis des paramètres à valeurs continues (quantités de matière utilisées, conditions opératoires, etc.).

De plus, elle est souvent spécifiée via des événements d'état (seuil de température ou de composition, etc.) et non par des durées ou des dates d'occurrence fixées a priori. Par conséquent, la modélisation des opérations et des transformations physico-chimiques des produits nécessite souvent l'implantation de modèles phénoménologiques. Dans ce contexte, l'utilisation des simulateurs à événements discrets (qui nécessitent des durées opératoires fixées) ainsi que les formalismes classiques intégrant seulement la notion d'événements temporels (automates temporisés (Alur, et al., 1994), réseaux de Petri temporisés (Sifakis, et al., 1977)) ou de vitesses de franchissement (réseaux de Petri continus (David, et al., 1990) ou hybrides (David, et al., 2001)) s'avèrent mal adaptés à cette classe de problème et l'utilisation de simulateurs dynamiques hybrides apparaît comme une solution adéquate.

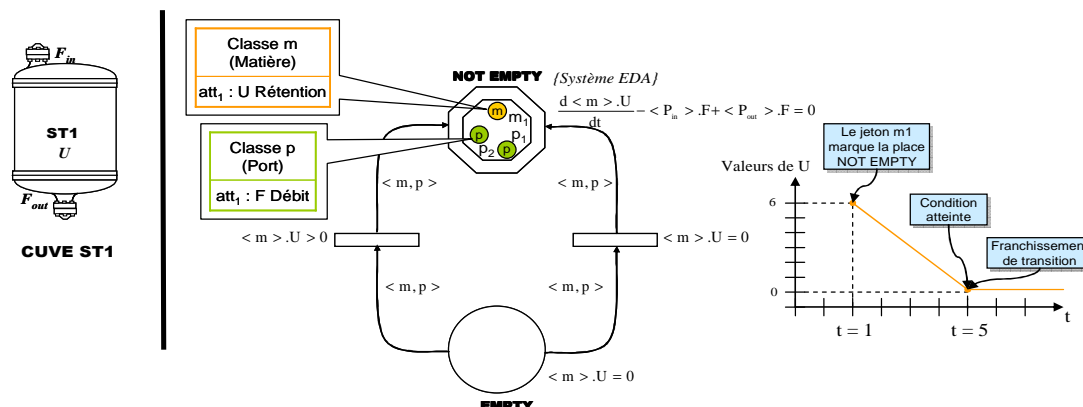


Figure 1. Principe d'évolution des RdPDO dans le modèle simplifié de la cuve

Dans ce cadre, l'unification des travaux de recherche en modélisation et simulation de procédés menés depuis plusieurs années au sein du Laboratoire de Génie Chimique a conduit au développement d'une plateforme de simulation dynamique hybride nommée ProDHYS. L'objectif de cette communication est d'abord d'illustrer les principes fondamentaux de ProDHYS à travers la mise en place d'un système hydraulique utilisé comme benchmark. Ensuite, le principe de la modélisation d'un procédé défectueux est présenté. Enfin, un exemple didactique illustre les potentialités de la simulation dynamique pour une étude de surveillance.

2. L'ENVIRONNEMENT ProDHYS

2.1. L'architecture logicielle

ProDHYS est conçu comme une bibliothèque de classes destinées à être dérivées afin de construire des éléments spécifiques plus complexes en exploitant les mécanismes propres à l'approche objet (composition, héritage, généricité). Son développement s'appuie sur un processus de conception et de codage unifié, basé notamment sur le formalisme UML et le langage C++ ainsi que sur la philosophie d'interfaces coopératives Orientées-Objet (notion de composants O-O) inspirée des recommandations du projet européen CAPE-OPEN et Global Cape-Open. Actuellement, cette bibliothèque regroupe plus de 1000 classes réparties en 2 couches fonctionnelles indépendantes. La couche interne de ProDHYS correspond au noyau de simulation et fournit les éléments de base utiles à la simulation de tout système dynamique hybride, quel que soit le domaine d'application. Quant à la couche supérieure, elle rassemble les classes utilisées pour la modélisation spécifique des procédés.

2.2. La spécification du modèle de simulation d'un procédé sous ProDHYS

La mise en point d'un procédé batch ou semi-continu porte autant sur l'analyse des phénomènes physico-chimiques qui s'y déroulent que sur la validation de la politique de conduite de l'unité. Pour cette raison, le

modèle de simulation d'un système est toujours divisé en deux parties distinctes :

- un modèle relatif à la partie « commande » (le superviseur) qui est, a priori, spécifique à la recette et à la topologie du procédé considéré.
- un modèle associé à la partie « opérative » (le procédé) qui est construit par assemblage des modèles spécifiques de chaque appareil constituant le procédé. Ces derniers doivent être, eux, réutilisables quelque soit le contexte d'utilisation (notion de composant).

Le modèle de simulation associé à chaque appareil est lui-même spécifié selon deux axes :

- un axe topologique qui consiste à décrire l'appareil en terme de connexions entre diverses structures composées hiérarchiquement dans lesquels circulent matière et énergie,
- un axe phénoménologique reposant sur les lois thermodynamiques et physico-chimiques ainsi que sur la notion de bilan.

À chaque état évoluant de manière continue est associé un système d'Équations Différentielles Algébriques (EDA). Afin de gérer de manière rigoureuse les dynamiques continues et discrètes, les Réseaux de Petri Différentiel Objet (RdPDO) ont été mis en oeuvre pour spécifier les séquences légalles de commutation entre les divers systèmes EDA décrivant le comportement de chaque composant (Olivier, et al., 2006, Perret, 2004). Ils permettent aussi de dissocier le modèle de la matière, de l'appareil qui la contient (notion de container) grâce à l'utilisation de jeton objet réutilisable dont le comportement est lui-même décrit par un RdPDO (Olivier-Maget, et al., 2007, Perret, 2004). La figure 1 montre un exemple d'évolution dans les RdPDO. Lorsqu'un jeton marque une place différentielle, le système EDA est instancié par substitution des variables formelles par les attributs du jeton. La résolution induit l'évolution continue de ces attributs ainsi que l'évaluation de la condition associée à la transition (événements d'état ou de temps). Lorsque cette condition est atteinte, l'action est exécutée et la transition est franchie.

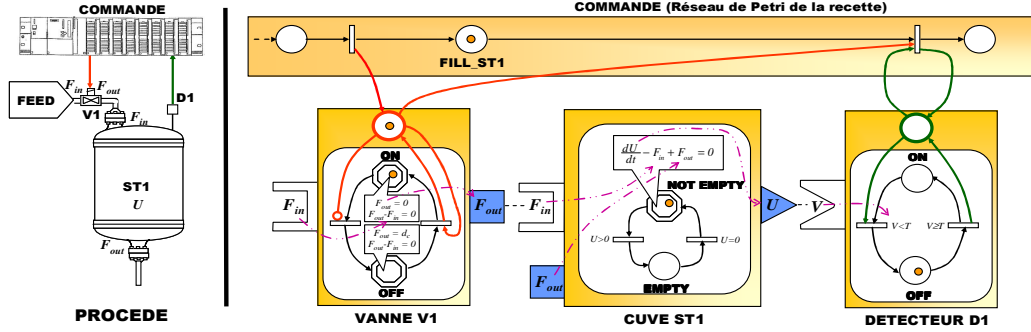


Figure 2. Interactions entre la partie commande et la partie procédé

Le modèle du niveau « commande » et le modèle de la partie « opérative » sont reliés uniquement grâce à des places dites de « signal ». Afin d'illustrer cet aspect, la figure 2 montre une séquence opératoire permettant d'alimenter un réacteur ST1 jusqu'à ce qu'un volume T soit atteint (modèles de simulation simplifiés). L'opération de remplissage du réacteur est commandée par la recette qui envoie un jeton sur la place signal de la vanne V1. L'alimentation est maintenue ouverte tant que cette place de commande reste marquée. Le marquage de la place signal du détecteur de niveau D1 indique que le volume de réactif a atteint la valeur cible T. La transition est alors tirée et provoque alors la fermeture de la vanne V1. De manière générale, le marquage d'une place de commande d'un dispositif *actif* induit l'évolution de son RdP, lui-même provoquant éventuellement l'évolution en cascade des dispositifs *passifs* auquel il est relié au travers du réseau constitué par la connexion des différents ports matière ou énergie.

supervision ont un rôle majeur. Certains systèmes industriels, tels que les procédés, rendent complexes ces fonctions en raison du nombre croissant de variables process mises en jeu et de la complexité de leurs relations. Le comportement du procédé est alors difficilement interprétable. De ce fait, les travaux de recherche en cours visent à concevoir un système de supervision couplé à un outil d'aide à la décision. Pour cela, le modèle de simulation de PrODHyS est utilisé comme modèle de référence pour mettre en œuvre les fonctions de détection et de diagnostic. L'architecture de ce système est montrée sur la figure 3. La supervision englobe les tâches de commande et de surveillance. La commande pilote l'exécution de la séquence opératoire de chaque recette et assure en temps réel la gestion des ressources nécessaires à cette exécution. La commande doit s'adapter à une déviation observée. Cela va de la modification du plan de fabrication par interaction avec le module d'ordonnancement, à la modification de conditions opératoires par interaction avec les niveaux de commande inférieurs. La fonction de surveillance complète alors parfaitement la tâche commande. Elle se réfère à une situation de fonctionnement normal que l'on cherche à maintenir pour assurer les conditions nominales de fabrication.

3. LE MODULE DE SURVEILLANCE : CONCEPTS

3.1. L'architecture

De nos jours, pour des raisons de sécurité et de performance, les fonctions de surveillance et de

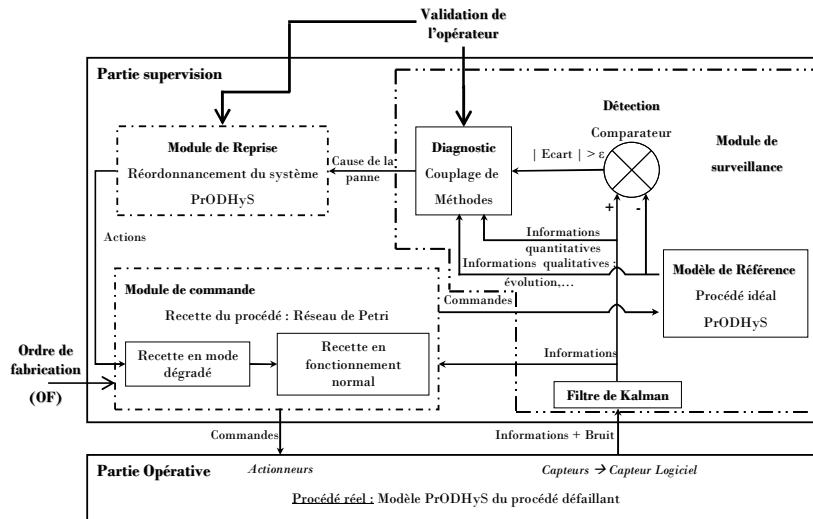


Figure 3. Architecture du système de surveillance

Ici la surveillance repose sur un modèle du procédé (modèle de référence) qui permet de comparer le comportement attendu, obtenu à l'aide du modèle, et le comportement réel, observé. En cas de déviation (détection), une cause compatible avec les informations disponibles sur le comportement réel du système et sur le comportement du modèle de référence (diagnostic) est trouvée. Le module de reprise est enfin lancé afin de déterminer la réaction adéquate. Notre idée est donc d'utiliser ProDHYS pour la simulation en temps réel : ce sera l'observateur du système physique. L'état complet du système sera disponible grâce aux résultats de la simulation en temps réel. L'outil étant en phase de prototypage, le procédé surveillé est pour le moment simulé au moyen de ProDHYS.

3.2. La détection

3.2.1 Les concepts

Le module de surveillance doit être capable de traiter aussi bien les défauts du système physique (fuites, pertes thermiques, etc.) que ceux des appareils de contrôle/commande (actionneurs, capteurs, etc.). La détection a pour rôle de révéler l'occurrence d'une possible défaillance. Comme définie dans (De Kleer *et al.* 1987), notre approche est orientée « bon fonctionnement ». Elle repose donc sur la comparaison du comportement attendu, obtenu à l'aide de la simulation du modèle de référence (valeurs des variables d'état) et le comportement réel observé (mesures provenant du procédé).

Par ailleurs, une détection par seuil fixe est utilisée. Cette méthode consiste à détecter l'instant pour lequel les grandeurs surveillées sortent d'une zone délimitée par des seuils autour d'une valeur objectif (valeur de la grandeur au sein du modèle de référence). Bien que des méthodes telles que la détection par seuil variable, soient plus rigoureuses et permettent la minimisation du nombre de détections erronées, (Duvilla, 2005) indique que la détection par seuil fixe reste plus robuste aux perturbations. Par ailleurs, elle permet d'obtenir une décision assez rapidement et est facilement implémentable et paramétrable.

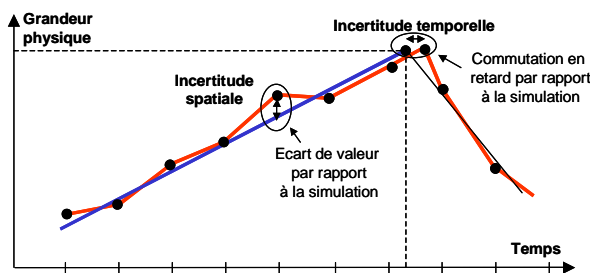


Figure 4. Les incertitudes sur la détection

Pour assurer le fonctionnement cohérent de cette fonction, il faut d'abord filtrer les mesures afin d'éliminer le bruit (filtre de Kalman (Welch, *et al.*, 1995)). Toutefois, il peut subsister une *incertitude*

spatiale. On note ε l'écart entre mesures et variables du modèle au-delà duquel on considère qu'un défaut est avéré. Cet écart reste un point délicat à évaluer (figure 4). Généralement, sa valeur est un compromis obtenu à partir d'une série de simulations dans lesquelles sa valeur est ajustée.

D'autre part, la simulation du modèle de référence doit évoluer en temps réel et de manière synchrone avec le procédé surveillé. En général, cette caractéristique ne peut pas être garantie à tout instant. En effet, une *incertitude temporelle* existe du fait des paramètres intrinsèques des modèles de simulation décrivant la dynamique du système. Ainsi, certains événements temporels ou d'état peuvent apparaître au niveau du modèle de référence simulé soit en avance, soit en retard par rapport aux observations faites sur le procédé surveillé. Ici encore, se pose le problème de déterminer à partir de quel écart ε peut-on considérer qu'une défaillance est potentiellement détectée. Dans tous les cas, une procédure de recalage du modèle de référence sur le procédé réel doit être réalisée pour pouvoir valider un test de détection. Tous ces facteurs soulignent la difficulté de l'étape de décision dans la détection. Notons enfin qu'à certains points de fonctionnement d'un système, un comportement défaillant peut même s'apparenter à un comportement normal, empêchant alors d'affirmer l'absence de défaut. Pour cette raison, même si l'occurrence d'un défaut est avérée, l'aboutissement de la phase de diagnostic reste soumise à l'incertitude liée à la véracité des tests de détection.

3.2.2 Une brève description du filtre de Kalman étendu

Le filtre de Kalman étendu a été développé par Schmidt (Mohinder *et al.*, 2001). Cette approche est probablement l'approche la plus commune et la plus populaire pour traiter les systèmes non-linéaires. Parmi les méthodes de diagnostic dites à base de modèles utilisées dans le passé, ce filtre est clairement l'un des plus étudiés. À l'origine, il a été exploité tant pour approximer les états et paramètres des procédés chimiques, que pour identifier les causes du comportement anormal du procédé (Watanabee *et al.*, 1984).

Le filtre de Kalman requiert un système à valeur continue en temps discret :

$$\begin{cases} X_{k+1} = f(X_k) + V_k \\ Z_{k+1} = h(X_{k+1}) + W_{k+1} \end{cases} \quad (1)$$

Où : $X_k \in \mathfrak{R}^n$ est le vecteur d'état du système,

f définit la dynamique du système,

$V_k \in \mathfrak{R}^n$ est le vecteur représentant les erreurs du modèle (bruit d'état),

$Z_k \in \mathfrak{R}^m$ est le vecteur d'observations (mesures),

h définit les mesures, c'est la matrice d'observation ; elle fait le lien entre le vecteur d'état et les mesures ;

et $W_k \in \mathfrak{R}^m$ est le vecteur qui représente le bruit des mesures,

Lors de l'utilisation de ce filtre, l'hypothèse posée consiste à dire que les bruits $\{V_k\}$ et $\{W_k\}$ sont des bruits blancs gaussiens de moyenne nulle et de matrices de covariances, respectivement Q_k et R_k :

$$\begin{aligned} E[V_k V_k^T] &= Q_k \\ E[W_k W_k^T] &= R_k \end{aligned} \quad (2)$$

La figure 5 illustre les différentes étapes de l'estimation de l'état du système. On distingue deux grandes étapes : la prédiction qui correspond à l'estimation a priori et la correction qui correspond à l'estimation a posteriori. Dans un premier temps, le vecteur d'état est estimé à partir des valeurs de l'étape précédente. Ensuite, il est corrigé grâce aux observations dans l'étape de correction. Ces deux étapes forment une boucle itérative :

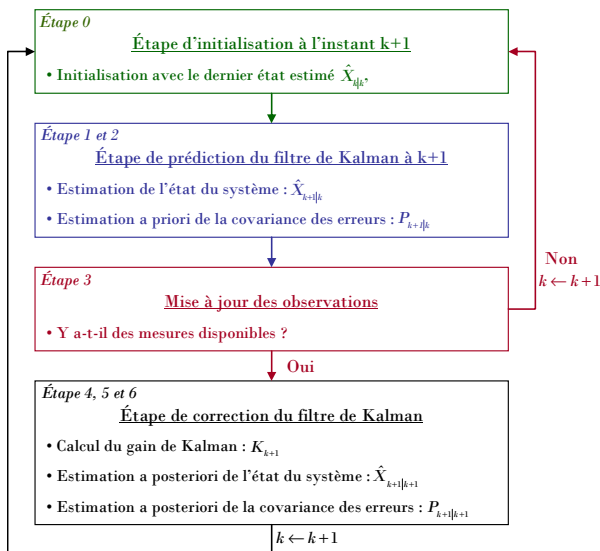


Figure 5. L'organigramme de principe du filtre de Kalman

La notation $k+1|k$ signifie que le calcul de la prédiction est faite à l'instant $k+1$ connaissant seulement l'état à l'instant k . Donc la notation $k+1|k+1$ correspond à la correction de l'estimation $k+1|k$ à partir des nouvelles informations de l'instant $k+1$ (mesures).

Les équations du filtre de Kalman sont algébriques et récurrentes. Par conséquent, le calcul d'une estimation nécessite peu de temps CPU, ce qui permet d'exploiter ce filtre pour du contrôle en temps réel. Dans l'algorithme, chaque nouvelle estimation est une combinaison entre la précédente estimation et les mesures actuelles. Les étapes suivantes correspondent à

une itération à l'instant $k+1$ de l'algorithme du filtre de Kalman :

Étape 0: Initialisation avec le précédent état estimé

$$\hat{X}_{k|k} : \hat{X}_{k|k} \leftarrow \hat{X}_{k+1|k+1}$$

Étape 1: Prédiction de l'état du système autour de $\hat{X}_{k+1|k}$

$$\hat{X}_{k+1|k} = f(\hat{X}_{k|k}) \quad (3)$$

Étape 2: Calcul de la matrice de covariance de l'erreur autour de $P_{k+1|k}$:

$$P_{k+1|k} = F_k \cdot P_{k|k} \cdot F_k^T + Q_k \quad (4)$$

Étape 3: Si de nouvelles mesures sont disponibles, alors allez à l'étape 4, sinon à l'étape 0

Étape 4: Calcul du gain de Kalman K_{k+1} par l'expression :

$$K_{k+1} = P_{k+1|k} \cdot H_{k+1}^T \cdot [H_{k+1} \cdot P_{k+1|k} \cdot H_{k+1}^T + R_{k+1}]^{-1} \quad (5)$$

Étape 5: Mise à jour de l'état estimé $\hat{X}_{k+1|k+1}$:

$$\hat{X}_{k+1|k+1} = \hat{X}_{k+1|k} + K_{k+1} \cdot [Z_{k+1} - h(\hat{X}_{k+1|k})] \quad (6)$$

Étape 6: Correction de la matrice de covariance de l'erreur :

$$P_{k+1|k+1} = [I - K_{k+1} \cdot H_{k+1}] \cdot P_{k+1|k} \quad (7)$$

Avec $F_k = \left. \frac{\partial f(x, u)}{\partial x} \right|_{x=\bar{x}_k}$ et $H_{k+1} = \left. \frac{\partial h(x)}{\partial x} \right|_{x=\bar{x}_{k+1}}$

4. L'INTÉGRATION DU MODULE DE SURVEILLANCE SOUS PRODHYS

4.1. Le module de détection

La méthode présentée précédemment a été adoptée pour la modélisation de la procédure sous ProDHyS. La figure 6 présente les modèles UML associées à la procédure. On retrouve ainsi la classe `Detection` qui génère automatiquement les résidus. Ceux-ci seront établis sur un horizon glissant représenté par la classe `Table`.

4.2. Le filtre de Kalman

La classe `KalmanFilter` (figure 7) possède deux attributs de types respectivement `SystemModelDynamic` et `MeasurementModelDynamic`, classes qui héritent toutes deux de la classe `modelDynamic`. L'un des attributs représente la dynamique du système et l'autre la dynamique des observations. La classe `modelDynamic` permet de prendre en compte tous les types de système : système linéaire ou non. Ainsi la procédure de calcul établi dans la classe `KalmanFilter`, procédure présenté dans le point précédent, est indépendante du système étudié. Les classe spécialisées de la classe `modelDynamic` sont appliquées à l'exemple étudié et est donc constituée d'un lien vers le solveur de ProDHyS qui contient les équations du système (classe `DISCoSolver`).

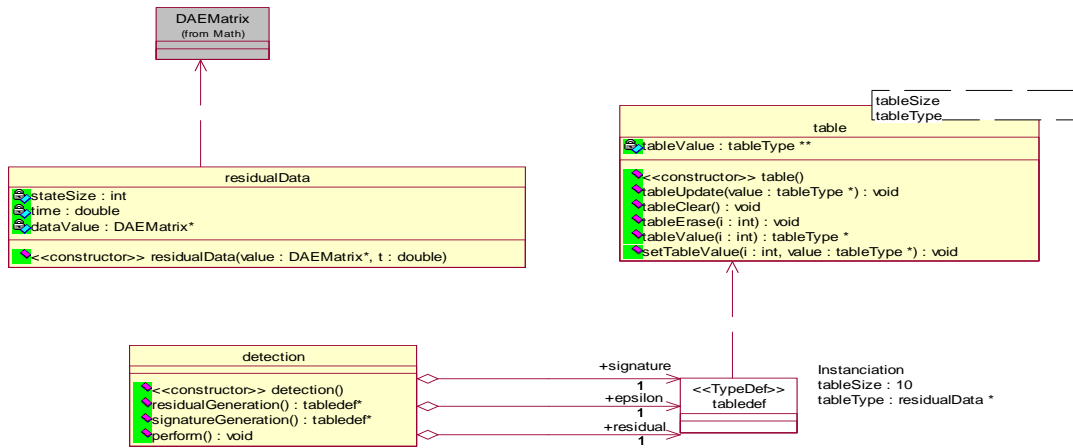


Figure 6. Diagramme de classe de la détection

5. LA SIMULATION DU PROCÉDÉ DÉFECTUEUX

Le module de surveillance doit être capable de traiter aussi bien les défauts du système physique (fuites, pertes thermiques, etc.) que ceux des appareils de contrôle/commande (actionneurs, capteurs, etc.). Dans notre travail, nous considérons les classes d'anomalies suivantes :

- les bruits : les bruits de mesure...
- les défaillances des capteurs et actionneurs : dérive d'un capteur, ou d'un actionneur, défaillance de la valeur seuil de sortie...
- les variations paramétriques du système : perturbations sur le procédé, température initiale faussée...
- les variations structurelles : usures mécaniques des pièces, défaillance d'un régulateur, fuite d'une conduite...

5.1. Le principe

L'outil étant en phase de prototypage, il n'est évidemment pas testé sur site avec un procédé réel. Pour cette raison, le procédé surveillé est actuellement simulé au moyen de PrODHyS. Le modèle de simulation

associé au procédé « réel » (avec potentiellement des défauts) est donc simulé en parallèle de celui du procédé de « référence » (sans défaut). Le procédé idéal (sans défauts) et le procédé surveillé (défauts probables) ont la même recette : la partie commande ne varie pas. La défaillance apparaît donc dans la modélisation des appareils.

5.2. La modélisation d'un appareil défectueux

La simulation du procédé « réel » nécessite donc de gérer les défaillances potentielles de chaque appareil le constituant. Pour cela, le modèle de simulation est construit à partir d'objets ElementaryDevice spécialisés dans lesquels les états de défaillances sont définis de manière intrinsèque et explicite. Ceci conduit à ajouter des places et des transitions dans le modèle de l'appareil « idéal » correspondant. La modélisation simplifiée d'une vanne tout ou rien et d'un détecteur est montrée sur la figure 8. Les éléments en trait continu correspondent au modèle de l'appareil « idéal » alors que les éléments en trait discontinu sont les éléments qui sont ajoutés dans les appareils avec défauts. Les transitions associées aux défauts sont sensibilisées au moyen de conditions (événements) activées par un gestionnaire de pannes.

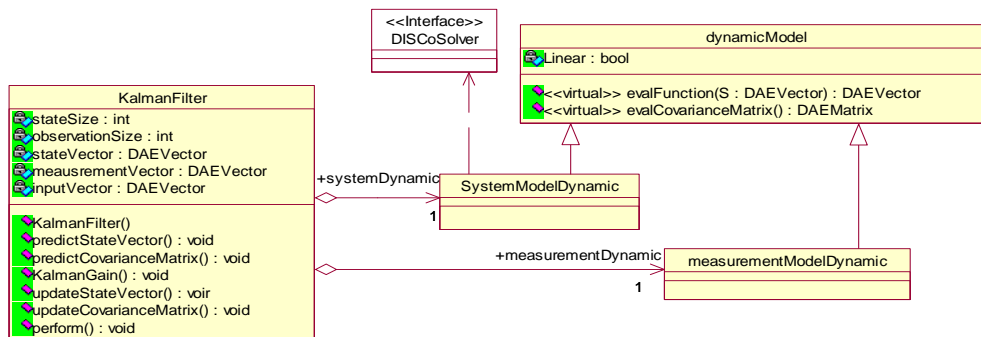


Figure 7. Diagramme de classe de la détection

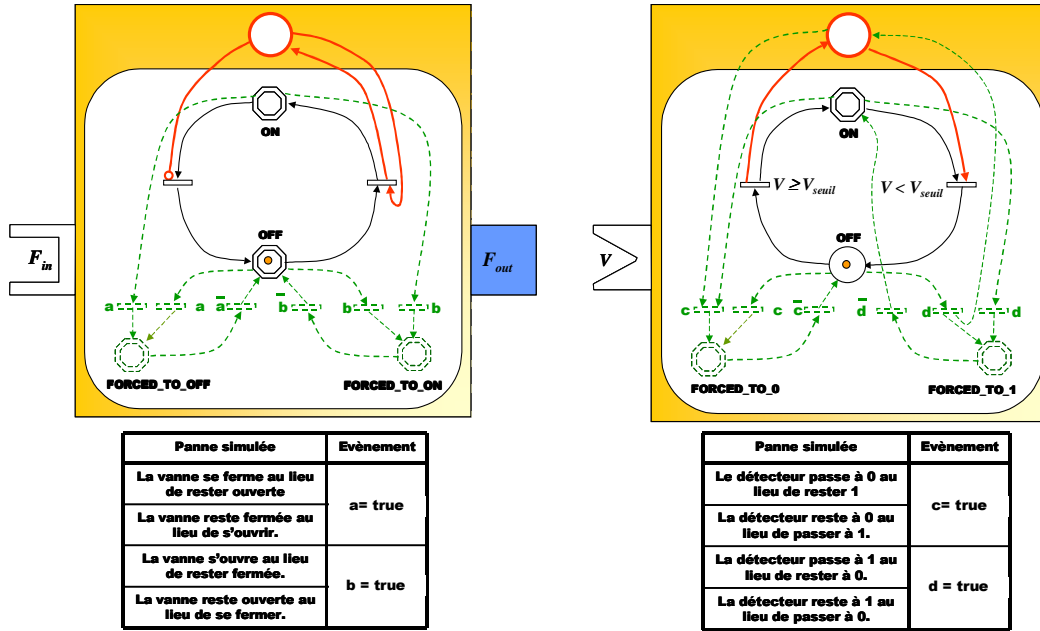


Figure 8. Modèle de simulation d'une vanne et d'un détecteur avec défauts

5.3. Le générateur de défaut(s)

En phase de test, les défauts seront évidemment générés de manière aléatoires. Par contre, en phase d'analyse du système, ils sont générés selon un calendrier fixé, chaque défaut étant définie par un identificateur, sa date d'occurrence et sa durée. Pour cela, un réseau de Petri est construit de manière automatique à partir des informations définissant chaque instance de défaut. La figure 9 montre le RdP correspondant à la génération de 3 défauts, les places différentielles modélisant ici des places temporisées.

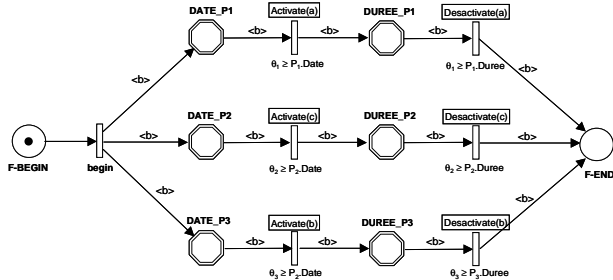


Figure 9. Réseau de Petri associé au générateur de défaut(s)

Aux franchissements des transitions, les pannes sont activées ou désactivées via une action. La figure 9 montre par exemple la gestion des événements a, b et c montrés sur la figure 8. Précisons bien que cette représentation rend possible l'apparition de plusieurs défauts simultanément.

Enfin, précisons que le procédé « idéal » et le procédé surveillé sont pilotés grâce à la même recette. Toutefois, deux instances de cette recette sont créées de manière à

pouvoir piloter de manière indépendante ces deux procédés.

6. LES RÉSULTATS

Le système hydraulique (figure 10) considéré est inspiré d'un benchmark défini par l'AS193 *Diagnostic des systèmes hybrides* (www.univ-lille1.fr/lail/AS193).

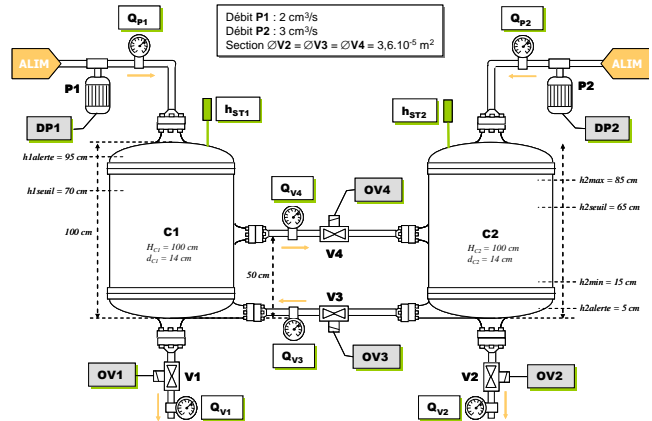


Figure 10. Flowsheet du Benchmark

Celui-ci est constitué de deux cuves cylindriques C1 et C2, reliés par deux conduites munies respectivement de vannes TOR V3 et V4. L'alimentation des cuves est réalisée par les pompes P1 et P2 pilotées en tout ou rien. Un soutirage au niveau de la cuve C2 est effectué grâce à la vanne TOR V2. La vanne V1 n'est pas utilisée ici. Enfin, l'instrumentation du procédé comprend 6 capteurs de débit et 2 capteurs de niveau dans la configuration maximale.

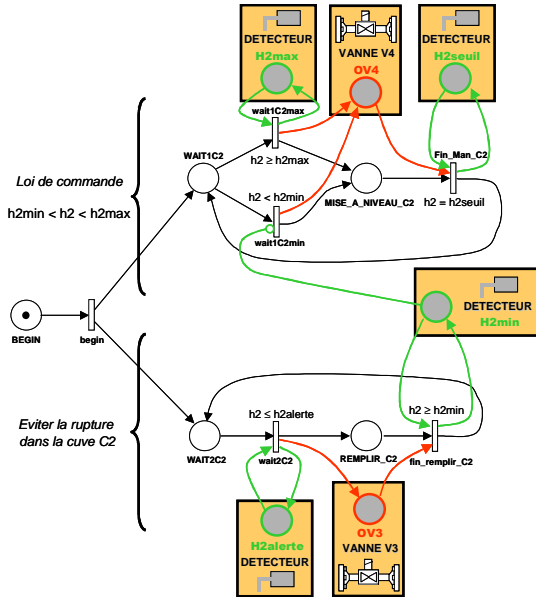


Figure 11. Réseau de Petri de la commande

Le système de commande a pour objectif principal de maintenir le niveau de liquide h_2 dans C2 entre les hauteurs h_{2min} et h_{2max} par action sur la vanne V4 (cf. réseau de Petri de la figure 10). Toutefois, la vanne V3 peut être ouverte pour éviter une rupture au niveau de C2 ($h_2 \leq h_{2alerte}$). Remarquons que la loi de commande mise en œuvre, volontairement rudimentaire, ne permet pas de garantir à tout instant ces objectifs car elle ne tient pas compte du niveau h_1 de C1.

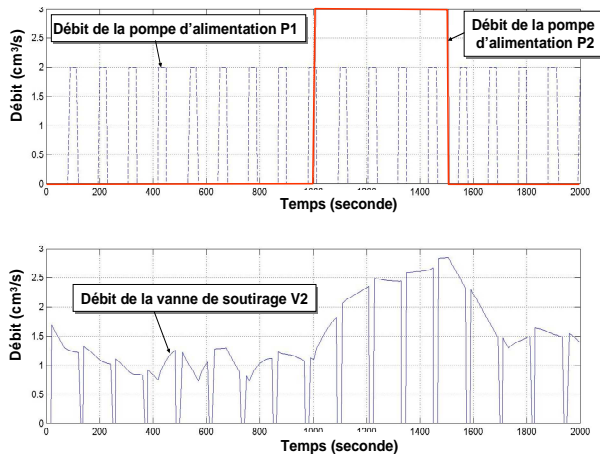


Figure 12. Les scénarios

Différents scénarii peuvent être simulés par action sur les pompes P1 et P2 et la vanne V2. Pour le jeu de paramètres indiqués sur la figure 10 et pour le scénario montré sur la figure 11, une défaillance de la vanne V3 est simulée : elle ne s'ouvre pas (figure 12). Avec seulement des mesures de débit, la détection de fautes basée sur un filtre de Kalman étendu est possible.

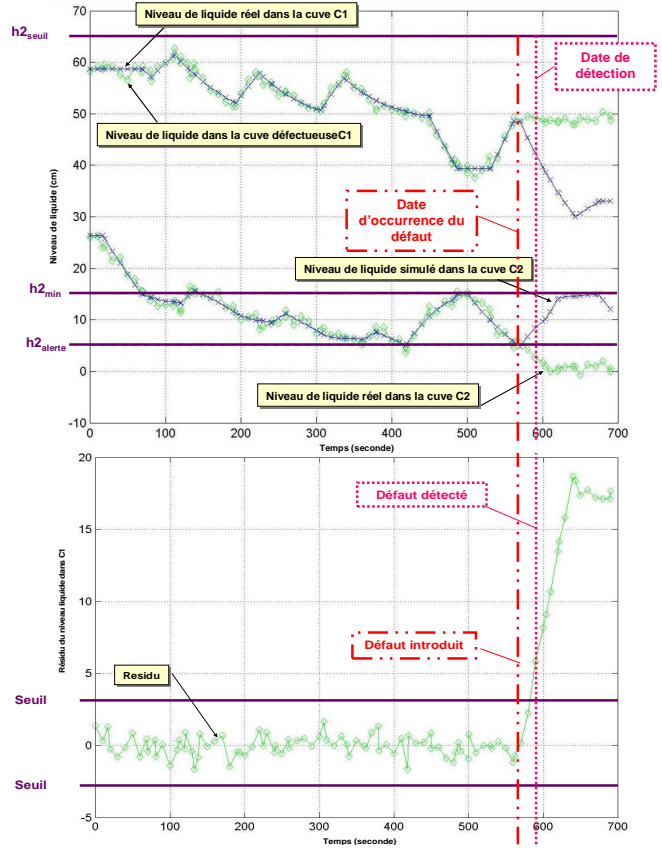


Figure 13. Résultats de la détection

7. CONCLUSION

L'approche orientée objet apporte de nombreux avantages en termes de qualité logiciel (extensibilité, réutilisabilité, flexibilité), mais surtout en termes de modélisation grâce à une description hiérarchique à la fois abstraite et proche de la réalité. Dans cet objectif, PRODHyS fournit des composants logiciels destinés à modéliser et à simuler plus spécifiquement les procédés chimiques. Par ailleurs, la modélisation de nombreux procédés nécessite la prise en compte de phénomènes hybrides. Tous ces éléments font donc de PRODHyS un outil opérationnel et évolutif. Les travaux en cours visent à intégrer le modèle de simulation avec un système de diagnostic. Le couplage entre les différentes méthodes de détection et localisation de fautes permet de tirer à profit les avantages de chacune et d'ainsi détecter l'occurrence d'un défaut le plus rapidement possible et ensuite de réduire le nombre de possibilités sur la cause de la défaillance observée et de valider le défaut éventuel.

REFERENCES

Alur R., Dill D.L. A theory of timed automata, *Theoretical Computer Science*, Vol. 126, No.2, 1994, pp. 183-235

Anderson T.W., An introduction to multivariate statistical analysis, New York: Wiley, 1984

Demeure I.,

- Farhat J., « Systèmes de processus légers : concepts et exemples », *Technique et Science Informatiques*, vol. 13, n° 6, 1994, pp. 765-795.
- Chen J., Patton R.J., Zhang H.Y., Design of unknown input observer and robust fault detection filters, *International Journal Control*, Vol. 63 (1) , 1999, pp. 85-105
- Chow E.Y., Willsky A.S., Analytical redundancy and the design of robust failure detection systems, *IEEE Transactions on Automatic Control*, Vol. 29 (7) , 1984, pp. 603-614
- David R., Alla H. Autonomous and Timed Continuous Petri Nets, *11th International Conference on Application and Theory of Petri Nets*, Saragosse, Espagne, Juin 1990
- David R., Alla H. On Hybrid Petri Nets, *J. of Discrete Event Dynamic Systems: Theory and Applications*, Vol. 11, 2001, pp. 9-40
- De Kleer J., Williams B. C., Diagnosing multiple faults, *Artificial Intelligence*, Vol. 32, 1987, pp. 97-130
- Duviella E. Conduite réactive des systèmes dynamiques étendus à retards variables. Cas des réseaux hydrographiques, Thèse de Doctorat, ENI Tarbes, France, 2005
- Frank P.M., Fault diagnosis in dynamic systems using analytical and knowledge- based redundancy – a survey and some new results, *Automatica*, Vol.26, 1990, pp. 459-474
- Gertler J., Singer D., A new structural framework for parity equation-based failure detection and isolation, *Automatica*, Vol. 26, 1990, pp.381-388
- Grewal M.S., Weill L.R. and Andrews A.P. Kalman filtering: theory & practice (using Matlab), 2nd edition, Wiley & Sons, 2001
- Olivier N., Hétreux G., LeLann J.M. Use of an Object Oriented Dynamic Hybrid Simulator for the Monitoring of Industrial Processes, *ADHS'06*, Alghero, Italy, 2006
- Olivier-Maget N., Hétreux G., LeLann J.M., LeLann M.V. Fault detection using a hybrid dynamic simulator: Application to a hydraulic system *IMSM-CMS'07*; Buenos Aires, Argentina, 2007
- Perret J., Hétreux G., LeLann J.M., Integration of an object formalism within a hybrid dynamic simulation environment, *Control Engineering Practice*, Vol. 12/10, 2004, pp. 1211-1223
- Shih R., Lee L., Use of fuzzy cause-effect digraph for resolution fault diagnosis for process plants I. Fuzzy cause-effect digraph, *Industrial and Engineering Chemistry Research* Vol. 34 (5), pp. 1688-1702, 1995
- Shih R., Lee L., Use of fuzzy cause-effect digraph for resolution fault diagnosis for process plants II. Diagnostic algorithm and applications, *Industrial and Engineering Chemistry Research* Vol. 34 (5), pp. 1703-1717, 1995
- Sifakis J. Use of Petri Nets for Performance Evaluation, Measuring, Modelling and Evaluating Computer Systems, H. Beilner et E. Gelenbe (dir.), *North-Holland Pub. Co*, 1977, pp. 75-93
- Venkatasubramanian V., Rengaswamy R., Yin K., Kavuri S. N. A review of process fault detection and diagnosis, *Computers & Chemical Engineering*, Vol. 27, 2003, pp. 293-346
- Watanabee K., Himmelblau D. M., Incipient fault diagnosis of nonlinear processes with multiple causes of faults, *Chemical Engineering Science*, 39 3, 1984, pp. 491-508
- Welch G., Bishop G., An introduction to the Kalman filter, Technical Report TR 95-041, University of North Carolina, 1995.
- Willsky A.S., A survey of design methods for failure detection in dynamic systems, *Automatica*, Vol. 12, pp. 601 – 611, 1976