



HAL
open science

Outil d'aide à l'élaboration de modèles hybrides de simulation pour les systèmes manufacturiers

Benoit Rohée, Bernard Riera, Véronique Carré-Ménétrier, Jean-Marc Roussel

► **To cite this version:**

Benoit Rohée, Bernard Riera, Véronique Carré-Ménétrier, Jean-Marc Roussel. Outil d'aide à l'élaboration de modèles hybrides de simulation pour les systèmes manufacturiers. Journées Doctorales du GDR MACS (JD-MACS'07), Jul 2007, Reims, France. Papier n°62. hal-00347047

HAL Id: hal-00347047

<https://hal.science/hal-00347047>

Submitted on 13 Dec 2008

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Outil d'aide à l'élaboration de modèles hybrides de simulation pour les systèmes manufacturiers

BENOIT ROHEE¹, BERNARD RIERA¹, VERONIQUE CARRE-MENETRIER¹, JEAN-MARC ROUSSEL²

¹ CReSTIC, Moulin de la Housse, BP 1039, 51687 Reims Cedex 2 France

Tel : +33 (0)3.26.91.32.26, Fax +33 (0)3.26.91.31.06

benoit.rohee, bernard.riera, veronique.carre @univ-reims.fr

² LURPA-ENS Cachan, 61, avenue du président Wilson, 94235 Cachan Cedex France

Tel : +33 (0)1.47.40.22.15, Fax +33 (0)1.47.40.22.20

jean-marc.rousseau@lurpa.ens-cachan.fr

Résumé — La simulation hors ligne des systèmes automatisés de production pilotés par des automates programmables industriels permet de tester une commande sans utiliser la partie opérative. Les avantages de l'utilisation de tels modèles sont multiples : réalisation simultanée de la commande et de la partie opérative, sécurisation de la commande, activités pédagogiques, ... La démarche proposée s'articule autour d'une modélisation continue de la partie opérative et des interactions discrètes avec la partie commande.

Mots clés — automate programmable industriel, simulation de partie opérative, Réseaux de Petri hybrides

I. INTRODUCTION ET OBJECTIFS

L'objectif de cette communication est de proposer une démarche de modélisation d'un système manufacturier en vue de simuler son comportement hors ligne. Les modèles proposés sont orientés vers une utilisation de type Systèmes à Evénements Discrets (SED) tels que les systèmes de production, transitiques [1], ... pilotés par des automates programmables industriels. Les applications de ce type de modèle sont multiples : élaboration et validation de la commande hors-ligne, réalisation de maquettes de simulation pour l'enseignement, de vitrines technologiques pour les fournisseurs de logiciels, mise à l'épreuve de commande [2], [3], ... Les modèles de simulation sont généralement élaborés en décrivant l'ensemble des séquences d'événements que la partie opérative peut accepter. Ces modèles doivent être indépendants de la commande [4] : le modèle de simulation de la partie opérative doit réagir à toutes les sollicitations possibles de la commande. La prise en compte du temps ne fait apparaître que l'ordre d'apparition des changements d'état des commandes et des capteurs. Ces modèles permettent de simuler une succession d'évolutions du moyen de production ou du flux de production [5]. En effet, le concepteur ne se contente pas d'une représentation discrète du système au travers des entrées / sorties lors des phases d'observation et de conception de la commande mais cherche à appréhender le fonctionnement continu du processus. A partir de cette information, il extrait des scénarios du comportement discret du système que la partie commande va suivre.

Nous proposons une démarche dans laquelle le modèle de simulation prend en compte le temps entre les évolutions et fait apparaître les inerties, les effets des changements de commande en cours de mouvement, les frottements, ... Tous ces paramètres ne peuvent être pris en compte que par une modélisation continue du processus physique. Les paramètres tels que les pannes, les perturbations aléatoires, ... qui génèrent un comportement non-déterministe ne sont pas pris en compte

dans le modèle ou, si on souhaite les faire apparaître, doivent être admis comme étant des entrées du modèle de simulation rendant ainsi le modèle déterministe.

L'outil de modélisation à retenir doit permettre de synchroniser l'évolution du modèle continu du processus avec les évolutions discrètes de la partie commande. Les Réseaux de Petri hybrides [6] permettent de modéliser ce comportement.

Tout d'abord, nous présentons succinctement l'outil de modélisation utilisé à savoir les Réseaux de Petri hybrides. Nous présenterons ensuite les modèles de la partie opérative issus d'un découpage fonctionnel sous la forme d'axes. Chaque élément de cet axe est modélisé indépendamment, assemblé. Enfin, un exemple de modèle est fourni pour illustrer la démarche avant de conclure.

II. OUTIL DE MODELISATION : RESEAUX DE PETRI HYBRIDES

L'outil de modélisation avec lequel nous avons choisi de présenter notre démarche est issu des Réseaux de Petri discrets et des Réseaux de Petri continus.

A. Réseaux de Petri discrets

Les Réseaux de Petri discrets [7] sont composés de places représentées graphiquement par des cercles simples, des transitions représentées par des traits simples reliés aux places par des arcs orientés. La condition pour qu'une transition soit franchie est que les places amonts à la transition soient marquées par le nombre de jetons indiqué sur l'arc entrant de la transition. A défaut, on admet que si rien n'est indiqué, il est nécessaire d'avoir un jeton dans la place amont. Le franchissement de la transition peut aussi être conditionné par un événement extérieur. Après franchissement de la transition, les places aval sont incrémentées du nombre de jetons indiqués sur les arcs sortant de la transition. Dans les Réseaux de Petri discrets autonomes, on admet que le franchissement de la transition est immédiat dès lors que toutes les conditions sont vérifiées.

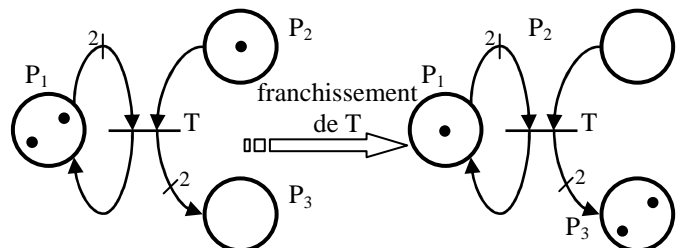


Figure 1. Franchissement d'une transition discrète

Dans l'exemple de la Figure 1, le franchissement de la transition T est conditionné par la présence de deux jetons dans la place P1 et d'un jeton dans la place P2. Au franchissement,

la place P1 est décrémentée de deux jetons et la place P2 est décrémentée d'un jeton. La place P1 est alors incrémentée d'un jeton et la place P3 de deux jetons.

B. Réseaux de Petri continus

Les Réseaux de Petri continus sont composés de places représentées graphiquement par des cercles doubles, des transitions représentées par des traits doubles reliés aux places par des arcs orientés. La condition pour qu'une transition soit franchie est que toutes les places amont à la transition soient marquées par un nombre réel strictement positif. Des proportions de flux liés au franchissement peuvent être indiquées sur les arcs. A défaut, on admet que, si rien n'est indiqué, les places sont diminuées d'un rapport. La vitesse de franchissement de la transition peut être conditionnée par une équation temporelle sur les valeurs des places amont. Pendant le franchissement progressif de la transition, les places amont sont proportionnellement diminuées avec les rapports indiqués sur les arcs entrant sur la transition. Les places aval sont proportionnellement augmentées avec les rapports indiqués sur les arcs sortant de la transition. Si les transitions sont régies par des équations différentielles, le franchissement de la transition est solution de cette équation. Il est cependant nécessaire de définir les conditions initiales sur les flux en plus du marquage initial.

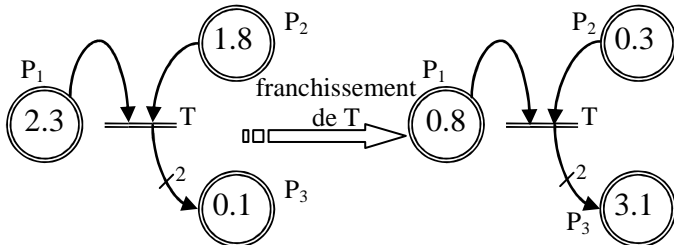


Figure 2. Exemple de trajectoire de franchissement d'une transition continue

Dans l'exemple de la Figure 2, le franchissement de la transition T est conditionné par la présence de marquage dans les places P1 et P2. Pour un franchissement de flux de 1.5, le marquage de la place P1 est diminué de 1.5 et la place P2 de 1.5. La place P3 est augmentée de 2×1.5 .

C. Réseaux de Petri hybrides

Les Réseaux de Petri hybrides reprennent la représentation graphique des Réseaux de Petri continus et discrets : les cercles doubles représentent les places continues et les cercles simples les places discrètes. Les transitions discrètes sont représentées par des traits simples reliés aux places par des flèches. Il est possible d'utiliser des transitions discrètes reliées à des places continues. Dans ce cas, le franchissement de la transition incrémente ou décrémente de la quantité indiquée sur l'arc respectivement les places avales ou amonts. L'incrément ou la décrément est discret : le marquage des places continues réalise un saut de valeur.

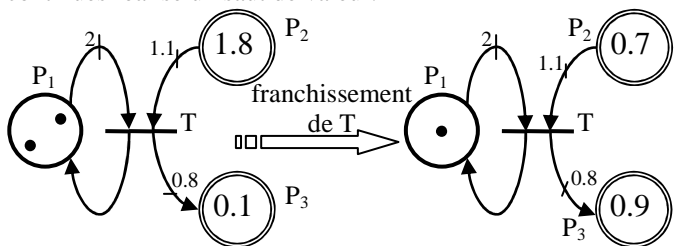


Figure 3. Franchissement d'une transition discrète dans un réseau de Petri hybride

Dans l'exemple de la Figure 3, le franchissement de la transition T est conditionné par la présence de deux jetons dans

la place P1 et de la présence d'un marquage de la place P2 supérieur à la valeur indiquée sur l'arc amont. Au franchissement, la place P1 est décrémentée de deux jetons et la place continue P2 est décrémentée de la valeur indiquée sur l'arc. La place P1 est simultanément incrémentée d'un jeton et la place P3 est augmentée de la quantité indiquée sur l'arc aval. Les transitions continues sont représentées par des traits doubles reliés aux places par des flèches. Il est possible d'utiliser des transitions continues reliées à des places discrètes. Cette notation est utilisée pour synchroniser la partie continue du modèle hybride d'un système avec la partie discrète. L'évolution de la partie continue est alors conditionnée par le marquage de la partie discrète. Un arc amont et aval lie la place discrète à la transition continue.

Dans l'exemple de la Figure 4, le franchissement de la transition continue depuis la place P2 vers la place P3 est conditionné par la présence d'un jeton discret dans la place P1. La transition ne peut pas être franchie s'il n'y a pas de jeton présent dans la place P1.

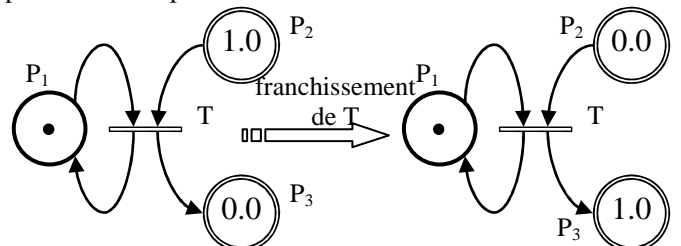


Figure 4. Synchronisation d'un modèle continu par un modèle discret

La partie discrète peut aussi être synchronisée par le modèle continu. La notion d'arc inhibiteur permet d'interdire le franchissement d'une transition discrète en fonction du marquage d'une place continue. La transition n'est franchissable que si la condition associée à l'arc inhibiteur n'est pas vérifiée. Le marquage de la place continue n'est pas affecté par la présence d'arcs inhibiteurs. L'arc inhibiteur est représenté par un arc orienté depuis une place continue pour aboutir à une transition où son extrémité est marquée par un cercle.

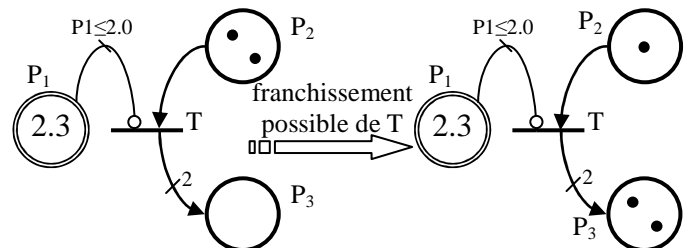


Figure 5. Exemple d'arcs inhibiteurs

Sur l'exemple présenté Figure 5, le franchissement de la transition T peut être inhibé suivant le marquage de la place P1. Le marquage de la place P1 est supérieur à la valeur 2.0. La condition $P1 \leq 2.0$ n'est donc pas vérifiée. La transition T peut donc être franchie. La place amont P2 est décrémentée d'un jeton et la place P3 est incrémentée de deux jetons.

La définition formelle de ces réseaux de Petri permet de les implémenter facilement pour les simuler. Nous proposons de les modéliser le fonctionnement des actionneurs par une modélisation continue. La partie discrète permettra de modéliser le comportement des préactionneurs. L'utilisation des Réseaux de Petri hybrides permet alors de synchroniser les modèles continus et discrets de ces différents éléments.

III. DECOMPOSITION ET MODELISATION DE LA PARTIE OPERATIVE

Les systèmes automatisés de production sont composés de préactionneurs, d'actionneurs et de capteurs. La chaîne d'action génère un effet sur la partie opérative à partir des commandes envoyées par la partie commande. La chaîne d'acquisition permet à la partie commande de recevoir des informations sur l'état de la partie opérative.

A. Définition d'un axe

Un axe [8], [9] est défini par un ensemble de matériels composé d'un actionneur, des préactionneurs l'alimentant, et des capteurs mesurant l'actionneur [10]. (Figure 6)

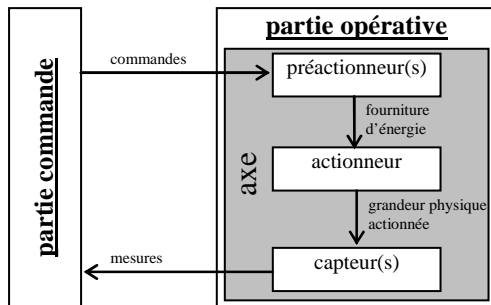


Figure 6. Description d'un axe

Les grandeurs physiques actionnées représentent les évolutions de caractéristiques physiques du système lorsque l'actionneur génère un effet sur le système réel. La grandeur physique représentée par l'axe peut être un positionnement relatif de sous-ensembles mécaniques, une température, un niveau de liquide, une pression ou toute autre grandeur modifiée sous l'effet d'un actionneur. Le préactionneur fournit de l'énergie à l'actionneur pour que celui-ci exerce un effet. On admet que, sous l'influence des commandes, le préactionneur ne peut prendre qu'un nombre fini d'états. Pour chacun des états pris par le préactionneur, l'actionneur va suivre une loi de fonctionnement. Les capteurs mesurent les conséquences des effets de l'actionneur.

B. Modélisation d'un axe par un réseau de Petri hybride étendu

Un axe peut être modélisé par un modèle hybride : il est composé à la fois d'une partie discrète (commande, mesures) et d'une partie continue (grandeur physique modifiée sous l'effet de l'actionneur). Nous supposons que, dans ces travaux, l'échantillonnage temporel effectué par l'automate programmable industriel lors de son fonctionnement cyclique est négligeable par rapport aux temps de réactivité de la partie continue. Les temps de commutation des préactionneurs sont supposés nuls.

1. Modélisation du préactionneur (Figure 7(a))

A partir de cette dernière hypothèse, on peut déterminer l'état

pris par le préactionneur et le modéliser par un réseau de Petri discret (Figure 8). Le modèle du préactionneur interprète les changements d'état des commandes et évolue si besoin. En fonction de l'état pris par le préactionneur, l'actionneur va suivre une loi de comportement modélisable par un modèle continu. Cette loi de comportement décrit « la loi de fonctionnement de l'actionneur ». Il peut s'agir, suivant le type et la technologie de préactionneur, d'une loi de fonctionnement en position, en vitesse, à énergie fournie constante, ... Les différentes lois de fonctionnement sont décrites dans le modèle d'actionneur.

Le préactionneur est modélisé par un réseau de Petri discret dans lequel les changements d'états des commandes sont associés aux transitions et les états du préactionneur sont associés aux places. Un seul jeton est toujours présent dans le modèle. L'état initial du préactionneur est supposé connu : en fonction de l'état initial des commandes et de leurs historiques lors de la préparation à la mise en énergie, on peut prévoir le comportement du préactionneur à l'état initial. L'état initial du préactionneur est associé au marquage initial du réseau de Petri modélisant le préactionneur.

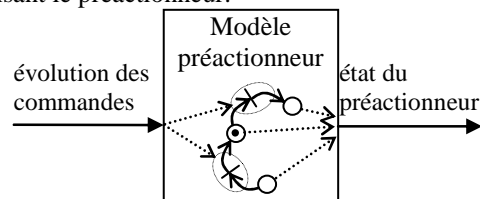


Figure 8. Modélisation du préactionneur par un réseau de Petri discret

2. Modélisation du comportement de l'actionneur (Figure 7 (b))

Modéliser le comportement de l'actionneur revient à décrire les évolutions de l'actionneur pour chacun des états pris par le préactionneur (Figure 9). Suivant le degré de finesse souhaité pour le modèle continu de l'actionneur, on peut identifier le comportement de l'actionneur sous la forme d'un modèle linéaire du premier ordre, du second ordre, avec ou sans perturbations, non linéaire, linéaire par morceaux, ...

Pour chaque configuration du préactionneur, l'actionneur est alimenté en énergie par le préactionneur et répond donc à un comportement donné. L'identification de ce comportement par un modèle déterministe permet de simuler le comportement de l'actionneur. Des hypothèses de fonctionnement sont nécessaires pour délimiter les conditions de validité du modèle identifié : alimentation générale en énergie, pas de dysfonctionnement du matériel, ... Le modèle doit être déterministe à partir des informations d'entrées du modèle. Pour représenter les évolutions de la grandeur physique sous la forme d'un axe par une variable numérique, il est nécessaire de définir une origine et un sens d'évolution.

Le comportement de l'actionneur peut être modélisé par un

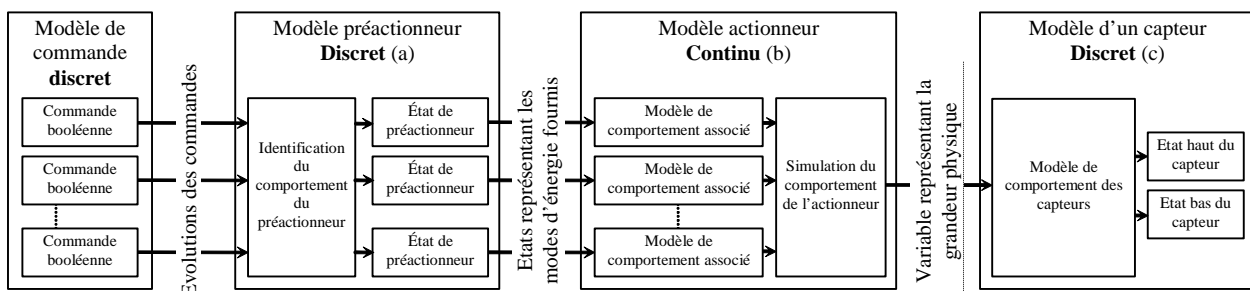


Figure 7. Modélisation de l'axe

réseau de Petri continu. Le modèle d'actionneur comporte deux places continues dont l'une d'elle représente la variable numérique de l'axe. Les Réseaux de Petri continus imposent que le marquage des places soit positif ou nul. La variable physique modélisée est donc minorée par la valeur 0. La seconde place continue permet de définir la limite maximale que peut prendre la variable physique : en associant un poids unitaire sur chacun des arcs, la somme des marquages des deux places est constante. Il s'agit d'un invariant de marquage. On définit une limite supérieure du marquage de la place modélisant la variable physique car la seconde place ne peut pas être marquée par une valeur négative. Le marquage des places continues évolue en fonction des lois de fonctionnement associées aux transitions. Chaque état pris par le préactionneur doit être associé à une transition dans le modèle d'actionneur. L'évolution du marquage des places continues peut être décrite par des équations temporelles sur les transitions. A chaque transition correspond une loi de fonctionnement associée à l'alimentation en énergie fournie par le préactionneur. On admet que la situation de l'actionneur est connue à l'initialisation. En fonction de cette situation initiale, le marquage des deux places est déterminé en respectant la valeur initiale de la variable physique et avec l'invariant de marquage.

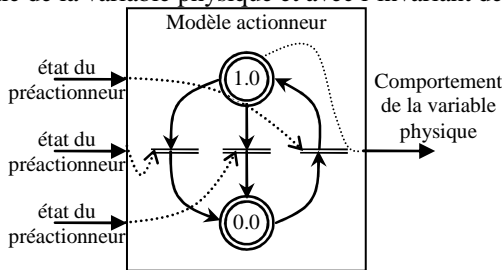


Figure 9. Modélisation de l'actionneur par un réseau de Petri continu

3. Modélisation du comportement des capteurs (Figure 7 (c)) Les capteurs Tout ou Rien changent d'état en fonction d'une grandeur physique. Chaque capteur est modélisé par un réseau de Petri discret synchronisé par la variable physique continue issue du modèle d'actionneur (Figure 10). On admet qu'un capteur n'est influencé que par un seul actionneur. Le changement d'état d'un capteur peut être modélisé comme un ou plusieurs passages par des seuils de la variable du modèle continu. Ces changements sont décrits par le sens d'évolution et le seuil de changement d'état. Grâce à ce type de modélisation, il est possible de représenter les effets d'hystérésis : le changement d'état du capteur dans un sens de variation de la grandeur physique n'implique pas toujours que le capteur change d'état dans l'autre sens de variation à la même valeur de seuil.

L'état initial des capteurs est supposé connu et en adéquation avec l'état dans lequel se trouve le modèle d'actionneur.

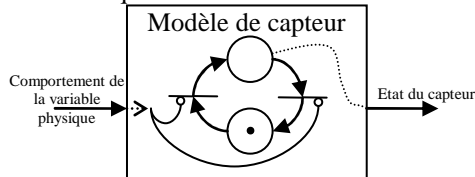


Figure 10. Modélisation d'un capteur par un réseau de Petri

4. Assemblage des différents modèles

L'assemblage des différents modèles permet d'obtenir un modèle de simulation de cet axe. Le modèle de préactionneur sélectionne le modèle de comportement de l'actionneur (Figure 11 (a) et (b)) ; le modèle d'un capteur est synchronisé avec le modèle de l'actionneur (Figure 11 (c)).

- le modèle de préactionneur sélectionne la loi d'évolution de la variable. (a)
-
- le modèle d'actionneur détermine l'état de la variable représentée sur l'axe. (b)
-
- le modèle de capteur synchronise le comportement d'un capteur en fonction de la variable représentée sur l'axe. (c)
-

Figure 11. Modélisation d'un axe

IV. MODELISATION DU COMPORTEMENT D'UN VERIN

Les systèmes de production automatisés manufacturiers comportent très souvent des actionneurs pneumatiques. Nous proposons dans cette partie d'appliquer la démarche de modélisation pour un axe de ce type. Le matériel choisi est un vérin simple effet piloté par un distributeur monostable 3/2 et équipé de deux capteurs de fin de course Tout ou Rien (Figure 12).

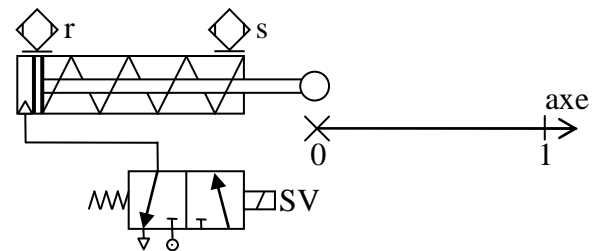


Figure 12. Eléments de partie opérative à modéliser

La commande SV permet de demander au distributeur d'alimenter le vérin en air comprimé pour sortir la tige. Les capteurs r et s mesurent les positions extrêmes de la tige du vérin. L'objectif est de modéliser et de simuler le comportement de cet axe.

A. Modélisation du distributeur

Le distributeur est de type 3/2 monostable. Il possède donc deux positions que l'on assimile aux états du préactionneur. Le modèle du préactionneur est présenté Figure 13. Les transitions du réseau de Petri représentent les deux événements associés à la commande SV : le passage de l'état bas à l'état haut (noté \uparrow SV) et le passage de l'état haut à l'état bas (noté \downarrow SV). Les places du réseau de Petri représentent les états pris par le

préactionneur pour fournir l'énergie pneumatique au vérin. Le vérin est alimenté pour sortir s'il a eu occurrence de $\uparrow SV$ et pour rentrer s'il a eu occurrence de $\downarrow SV$. A l'initialisation, on suppose que la commande SV est à l'état bas.

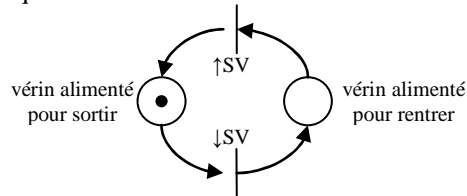


Figure 13. Modèle de préactionneur 3/2

B. Modélisation de l'actionneur

Sous l'effet de l'énergie pneumatique fournie par le préactionneur, la tige du vérin se déplace. La grandeur physique étudiée est la position de la tige du vérin par rapport au corps (Figure 14). On place l'origine de l'axe en associant la position « rentrée de la tige » à la valeur 0.0 de la position. L'échelle et le sens de l'axe sont définis en associant la valeur 1.0 à la position « sortie de la tige ». La course du vérin est limitée à ces deux positions extrêmes. Le domaine de validité de la variable modélisant la grandeur physique est donc de $[0.0, 1.0]$. La variable correspond au marquage continu d'une place. Pour simuler le comportement du vérin, il est nécessaire de modéliser et d'identifier le comportement de celui-ci par un modèle plus ou moins détaillé. La vitesse de déplacement de la tige du vérin peut être admise comme étant constante au cours du temps ou comme étant assimilable à un modèle continu du premier ordre pour tenir compte des inerties mécaniques de la tige du vérin. Ce modèle décrit l'évolution sur les transitions. Le marquage initial est déterminé par l'état initial de l'axe à la mise en marche supposé connu (vérin rentré).

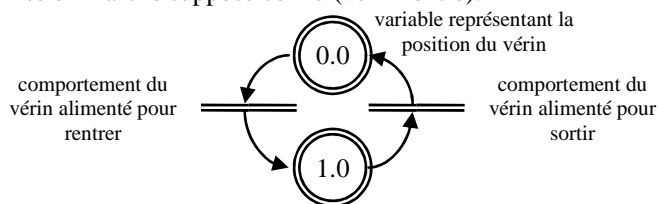


Figure 14. Modèle du vérin

Le modèle proposé peut tenir compte des éventuelles perturbations extérieures : blocage de la tige du vérin, frottements, masse variable entraînée par la tige, variation de la pression d'alimentation, ... Pour réaliser la simulation de la partie opérative, il est cependant nécessaire que le modèle d'actionneur soit déterministe : une seule évolution est possible par l'actionneur. Les perturbations peuvent néanmoins être assimilées à des entrées du modèle de partie opérative. Le blocage de la tige du vérin peut être commandé dans la simulation par l'utilisateur (défaillance) ou associé au fonctionnement d'autres actionneurs (collisions entre actionneurs). Une fois le type de modèle choisi pour représenter le comportement de l'actionneur, il faut identifier les paramètres de ce modèle au comportement réel du système. La variable représentant la position continue du vérin peut, si nécessaire, être rendue publique et être visualisée dans une animation graphique. Cette donnée n'est cependant pas observable par la partie commande avec la partie opérative réelle.

C. Modèle des capteurs

Les capteurs mesurent la position de la tige du vérin et donnent une information discrète de celle-ci. Modéliser le comportement des capteurs revient à synchroniser les

changements d'états des capteurs en fonction de l'état de l'actionneur. Le changement d'état d'un capteur dépend de l'état précédent dans lequel il se trouvait (mémorisation de l'état précédent) et des paramètres liés à l'actionneur. Dans le cas des capteurs de fin de course, les changements d'états de ces capteurs dépendent de la position de la tige du vérin et, éventuellement de la vitesse de déplacement de la tige du vérin. La synchronisation entre le modèle de capteur et le modèle d'actionneur est réalisée à l'aide d'arcs inhibiteurs (Figure 15). La transition est franchie si la condition associée à l'arc inhibiteur est fausse.

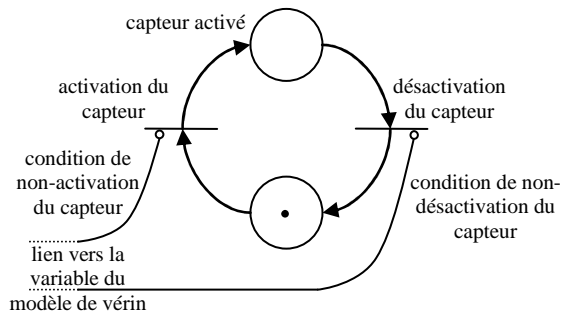


Figure 15. Modèle d'un capteur

Les conditions d'activation et de désactivation sont décrites par le sens de déplacement de la tige et le seuil d'activation. Les capteurs ne peuvent pas détecter une seule valeur de la position du vérin mais un intervalle. A partir du positionnement et de la technologie des capteurs utilisés, on modélise le changement d'état des capteurs fin de course de l'état bas à l'état haut si l'actionneur arrive à l'extrémité, par exemple, avec une marge de 3% de la course. Les capteurs passent à l'état bas si le vérin sort de l'extrémité avec, par exemple, une marge de 5%. Le comportement des capteurs associés à la variable modélisant la position de la tige est présenté Figure 16. A l'état initial, le capteur r est activé et le capteur s est désactivé. Ces conditions correspondent bien à l'état dans le modèle du vérin.

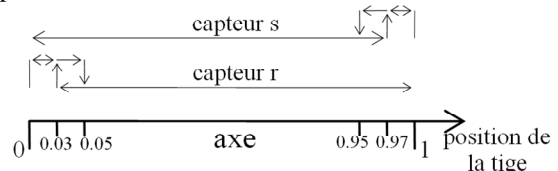


Figure 16. Evolution des capteurs r et s en fonction de la position de la tige du vérin

D. Le modèle complet de l'axe

En assemblant les modèles du distributeur, du vérin et des capteurs, on obtient le modèle présenté Figure 17. Le modèle de préactionneur (Figure 17 (a)) change d'état en fonction de l'état des commandes. A chaque état du modèle du préactionneur est associé un comportement des transitions du modèle d'actionneur (Figure 17 (b)). Le modèle d'actionneur est contraint à évoluer suivant les lois d'évolutions sélectionnées par le modèle de préactionneur. Le modèle d'actionneur fait évoluer les modèles de capteur (Figure 17 (c) et (d)) en fonction de l'évolution de l'actionneur. Le modèle complet simule le comportement complet d'un axe permettant ainsi de représenter les évolutions des capteurs en fonction des commandes en tenant compte de la dynamique de la partie opérative. Il est ainsi possible de tenir compte des inerties de déplacements, des frottements, ...

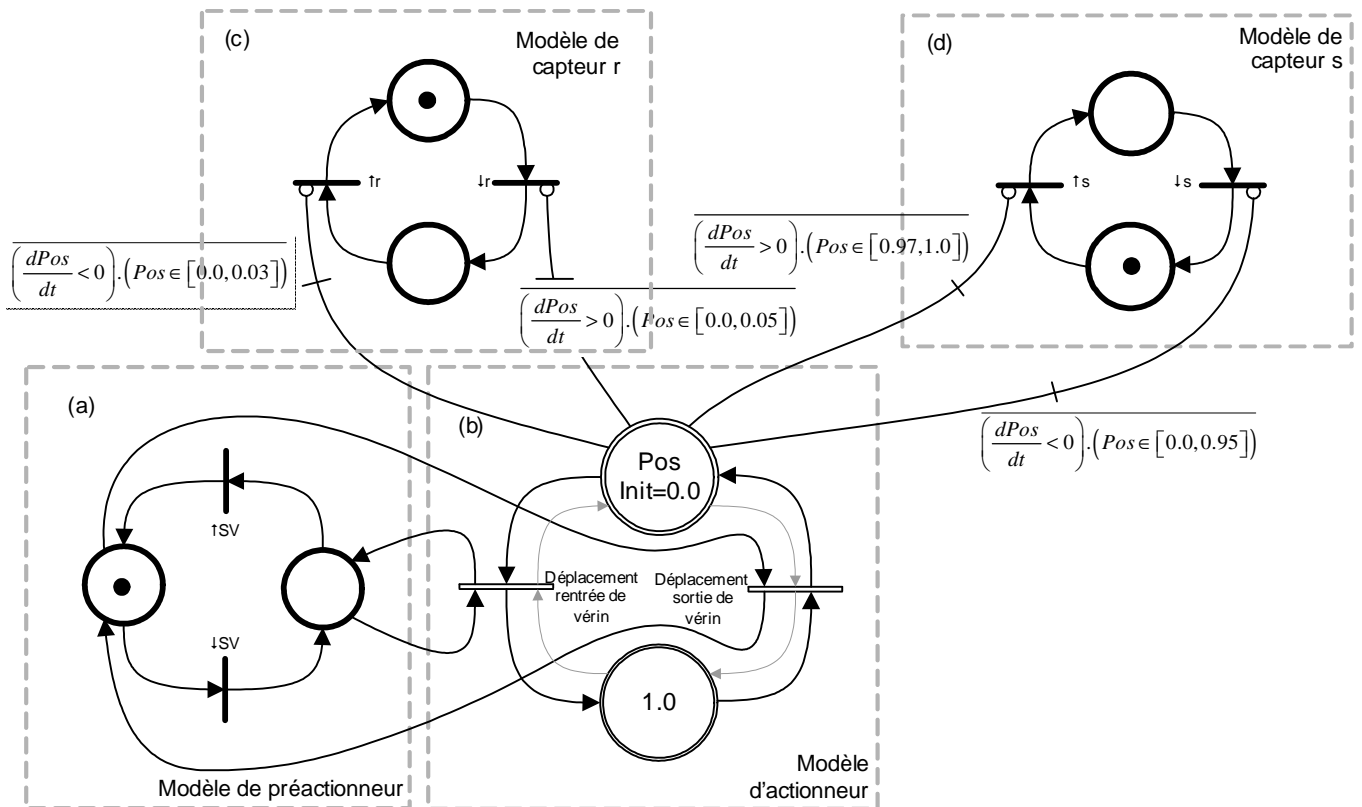


Figure 17. Modèle complet de l'axe

V. CONCLUSION ET PERSPECTIVES

La modélisation de la partie opérative des systèmes automatisés de production sous la forme de modèles discrets n'est pas toujours suffisante : le processus réel est continu et dépendant du temps. La simulation de la partie continue et l'étude des interactions entre le modèle continu et les données discrètes échangées avec la partie commande apportent des simulations plus réalistes que la simple simulation d'un modèle discret afin de tenir compte des inerties, des changements de commande en cours de mouvement, des frottements, ... permettent d'utiliser cette modélisation hybride pour mettre à l'épreuve une commande discrète sans utiliser une partie opérative réelle. Les applications de ce type de modèle sont multiples : élaboration d'une commande sûre, réalisation de maquettes pédagogiques, ... Pour valider notre approche, nous souhaitons implanter le résultat d'une telle modélisation dans des outils industriels de simulation afin de pouvoir juger des apports sur des cas concrets et de leur potentiel dans une problématique de supervision industrielle. Pour être à même de simuler l'ensemble des cas, il est nécessaire de pouvoir modéliser les interactions entre les actionneurs, le partage de capteurs entre plusieurs actionneurs, ... mais également les flux de production.

VI. REFERENCES

- [1] J-S. Mouchard, P. Berruet, J-L. Philippe, J-P. Guyomar, *Modeling, simulating and validation : an application to the design of transitive systems*, IFAC/IFIP/IEEE on Management and Control of Production and Logistics (MCPL'2000), Grenoble, France, 2000
- [2] R. Bardhol, C. Ermel, L. Ribiero, *A modular approach to animation of simulation Models*, 14th Brazilian symposium on software Engineering (SBES'00) Jao Pessoa, Brazilia 10/2000 p498-520
- [3] J. Perret, G. Hetreux, J-M Le Lann, *Modélisation des systèmes dynamiques hybrides basée sur le formalisme Réseaux de Petri prédicats-transitions-différentiels-objets*, 4^{ème} conférence francophone de modélisation et simulation MOSIM'03 04/2003 Toulouse
- [4] B. Rohee, B. Riera, V. Carre Menetrier, J-M. Roussel, *A methodology to design and check a plant model*, 3rd IFAC workshop discrete event system design DESDes'06 Rydzyna, Poland 09/2006
- [5] G. Frey, K. Moßig, M. Schnabel, *Assembly line sequencing based on Petri-net T-invariants*, 9th IFAC Symposium on information control in manufacturing. INCOM'98 Nancy-Metz 06/1998
- [6] Zaytoon Janan, *Systèmes dynamiques hybrides*, Traité IC2 Information, Commande, Communication. Hermès science 2001
- [7] R. David, H. Alla, *Du Grafcet aux Réseaux de Petri*, Traité des nouvelles technologies série automatique Hermès 1989
- [8] Simac, Prosyst / Schneider <http://www.prosyst.fr>
- [9] T. Meftah, H. Guéguen, N. Bouteille and V. Boutin, *Spécification par contraintes de la commande des systèmes automatisés*, Journées Doctorales et Nationales du GDR MACS, JDMACS 2005, Lyon, 09/2005.
- [10] S. Balemi, *Input/output discrete event processes and system modeling*, Proceedings of the joint workshop on discrete event systems (WODES'92) Prague, Tchecoslovaquie, 08/1992