

Identification de système à un automate hybride : revue des travaux récents

Yan Monier, Gregory Faraut, Bruno Denis, and Nabil Anwer

LURPA, ENS Paris-Saclay, Université Paris-Saclay 4 av. des sciences, 91190 Gif-sur-Yvette, France
prénom.nom@ens-paris-saclay.fr

Abstract

Ce papier explore la manière dont le problème d'identification de système sous la forme d'un automate hybride est abordé dans la littérature récente. Nous expliciterons quelles sont les sous-problématiques inhérentes à l'identification ainsi que la manière dont elles sont traitées dans les différents travaux. Nous expliquerons leurs points communs, leurs divergences et les futurs défis à surmonter dans le domaine.

1 Introduction

Avec l'augmentation de la complexité des systèmes dynamiques, il est de plus en plus difficile de les caractériser, de prédire leur évolution et d'appréhender les actions à mener pour assurer leur bon fonctionnement. Pour y faire face, des modèles plus riches sont élaborés et plus récemment des jumeaux numériques sont proposés [24].

Pour décrire la dynamique continue d'un système, conjointement avec sa dynamique discrète ainsi que leurs interactions, les automates hybrides [3] connaissent un succès croissant dans les domaines les plus variés comme dans l'industrie automobile (véhicule autonome [11], régulateur de vitesse adaptatif [15], prédiction du comportement des piétons [12]), comme dans la biologie et le médical (pacemaker [2], métabolisme cellulaire [16] propagation des pandémies [22], interaction robot-humain dans la santé [14]), comme dans les réseaux de distribution d'électricité [33] ou de gaz [23], mais aussi comme dans la chimie [4], l'électronique [20], l'agriculture [17] ou encore la sociologie [21]. À partir d'un automate hybride qui modélise le comportement dynamique d'un système, le modélisateur peut faire une analyse par simulation, [6], la synthèse d'un contrôleur [26], la vérification formelle du modèle [5] ou l'étude de la sûreté de fonctionnement [31].

Cependant, se pose la question de l'obtention de ce modèle. Sa conception de manière experte est parfois difficile et/ou coûteuse. L'identification de système est une alternative intéressante qui consiste en l'obtention d'un modèle d'une classe préalablement définie, ici les automates hybrides, à partir de données observées sur le système (mesures). Le caractère fondamental de l'identification de système dans la science en fait un sujet développé dans de nombreuses communautés comme celles de la *théorie du contrôle* (*control theory*), de l'*informatique* (*computer science*), de l'*exploitation des données* (*data mining*) ou de l'*intelligence artificielle* (*artificial intelligence*). En corollaire, le terme identification qui a été introduit par Lotfi A. Zadeh [32] dans le domaine du contrôle peut se retrouver sous des vocables très différents, par exemple « learning » [25], « mining » [18], « recovering » [27], « data driven discovery » [30] ou « synthesis from time-series data » [7] (tous dans le contexte des automates hybrides).

Nous proposons ici une revue des travaux récents (2015-2023) portant sur l'identification de systèmes sous la forme d'un automate hybride. Nous avons listé des équipes de recherche autour des chercheurs suivants (par ordre alphabétique) : Fathiyeh Faghieh ([25], University of Tehran, Iran), Sebastian Fischmeister ([18], University of Waterloo, Canada), Sandeep Gupta

([13], Arizona State University, USA), Amit Gurung ([10], National Institute of Technology, Meghalaya, India), Thomas Henzinger ([9], [7], [8], ISTA, Austria), Taylor T. Johnson ([29], Vanderbilt University, Nashville, USA), Michael Mateas ([27], University of California, Santa Cruz, USA) et nos travaux au LURPA ([19], Université Paris-Saclay, France).

Ce papier est organisé de la façon suivante. La partie 2 présente les concepts qui définissent les automates hybrides et le problème de l'identification, puis la partie 3 propose un découpage du problème de l'identification en un ensemble de sous-problèmes issus de la revue de littérature. La partie 4 est consacrée à l'ordonnancement de résolution de ces sous-problèmes, pour finir sur la partie 5 qui aborde les futurs défis de l'identification à un automate hybride.

2 Concepts d'automate hybride et d'identification

La définition d'automate hybride (syntaxe et sémantique) et celle du problème de l'identification peuvent varier selon les travaux. Nous faisons ici l'inventaire des concepts de base communs à tous. Une représentation graphique d'un automate hybride, schématisant l'ensemble des concepts énoncés est présentée figure 1.

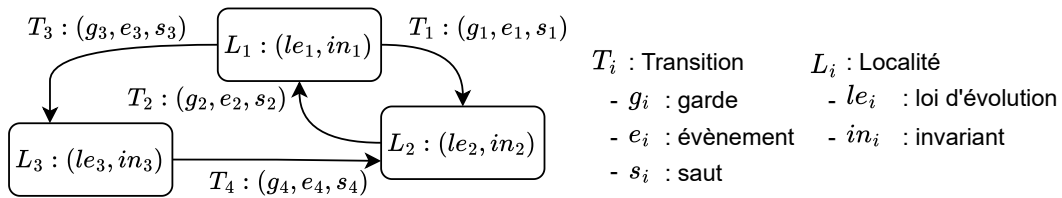


Figure 1: Représentation graphique d'un automate hybride

2.1 Syntaxe des automates hybrides

La définition d'un automate hybride repose sur des ensembles de variables, d'évènements, de localités, de transitions, d'invariants, de gardes, de sauts et sur leurs interrelations.

2.1.1 Variables

Les variables d'un automate hybride prennent leur valeur dans l'ensemble des réels \mathbb{R} et elles sont soit endogènes, soit exogènes. L'évolution des *variables endogènes* est entièrement déterminée par l'automate hybride tandis que l'évolution des *variables exogènes* est subie par l'automate et influencent son évolution. Dans la littérature, certains modèles d'automate hybride comportent seulement des variables endogènes ([7], [9], [8]), ces modèles sont dits autonomes.

2.1.2 Localités (locations)

Un automate hybride est construit autour de ses localités, représentant chacune un état discret ou un mode de fonctionnement du système. Elles sont les nœuds d'un graphe (L_i dans la figure 1). À chaque localité sont associés une loi d'évolution des variables endogènes et un invariant sur l'ensemble des variables.

Lois d'évolution (flows) – Chaque localité est associée à une loi d'évolution qui décrit l'évolution continue de l'ensemble des variables endogènes. Un système d'équations différentielles couplées est généralement utilisé à cet effet.

Invariants de mode (invariants) – Chaque localité peut être associée à un invariant qui décrit un domaine de valeurs acceptées pour les variables endogènes et exogènes. Un invariant est souvent écrit sous la forme d'un prédicat sur les valeurs des variables.

Dans la littérature, tous les automates hybrides comportent une loi d'évolution pour chacune de leurs localités. Quant aux invariants, certains types d'automates hybrides en font l'impasse en considérant que les transitions sont franchies dès qu'elles sont franchissables ([10], [18], [25], [29]).

2.1.3 Transitions

Les transitions de l'automate hybride représentent la capacité du modèle à changer d'état discret ou de mode de fonctionnement. Les transitions sont représentées dans un graphe comme un arc orienté (T_i dans la figure 1) partant d'une localité source vers une localité cible. Chaque transition peut être associée à une garde, un événement et un saut.

Gardes (guards) – Chaque transition doit être associée à une garde. Une garde représente un domaine de valeurs accepté des variables endogènes et exogènes du modèle. Une garde est souvent écrite sous la forme d'un prédicat sur les valeurs des variables.

Évènements (events) – Chaque transition peut être associée à un événement. L'évènement, souvent représenté sous la forme d'un label, symbolise, lorsqu'il est observé, le moment de franchissement de la transition. On peut noter deux types d'évènements. Les *évènements endogènes (Endogenous Event)* sont générés par l'automate au moment du franchissement d'une transition et ils représentent des événements de sorties du système. Les *évènements exogènes (Exogenous Event)* sont subis par l'automate, ils conditionnent le moment de franchissement d'une transition. Ils représentent les événements d'entrées du système.

Sauts (jumps) – Ils représentent la valeur imposée aux variables endogènes à la suite du franchissement de sa transition. Les variables endogènes peuvent alors garder la valeur qu'elles avaient avant le franchissement, ce qui conserve le caractère continu de l'évolution de ces variables, ou bien changer de valeur et introduire une discontinuité.

Dans les différents travaux, au moins l'un des deux éléments entre la garde et l'évènement est toujours présent sur les transitions. Dans certains cas, la garde, considérée comme urgente, est confondue à la notion d'évènement ([10], [13], [18], [25], [29]), dans d'autres, seulement la garde, non urgente, est utilisée afin d'obtenir un modèle dans lequel le moment de franchissement d'une transition n'est pas précisément déterminé ([7], [9], [8]). Les sauts quant à eux sont souvent implicites en considérant que l'évolution des variables endogènes est continue dans le temps ([7], [9], [13], [18], [25], [8], [29]).

2.2 Sémantique des automates hybrides

Dans la littérature, plusieurs concepts relatifs à l'évolution d'un automate hybride sont présents. Nous allons ainsi décrire quelques notions et propriétés des différentes variantes d'automates hybrides qui nous semblent pertinentes.

2.2.1 Évolution continue de l'automate

Parmi l'ensemble des localités de l'automate hybride, une des localités dites "localité courante de l'automate" représente le mode de fonctionnement du système modélisé à un instant donné. Pendant la durée durant laquelle cette localité reste la localité courante, l'ensemble des variables endogènes évoluent en suivant la loi d'évolution associée à cette localité. Le modèle ne peut changer de localité (ou le système de mode de fonctionnement) qu'au moment d'une commutation.

2.2.2 Commutation d'une localité à une autre

La commutation représente un changement de localité courante de l'automate hybride. Le modèle peut évoluer d'une localité courante L_1 vers une nouvelle localité L_2 seulement s'il existe une transition de source L_1 et de cible L_2 , que la valeur courante de l'ensemble des variables est incluse dans le domaine de la garde associée à cette transition et que la valeur courante de l'ensemble des variables est incluse dans le domaine de l'invariant associé à la localité cible L_2 . Si ces conditions sont vérifiées, la transition est dite franchissable. Lors d'un changement de localité, on dit qu'une transition est franchie, ou qu'une commutation de mode s'est produite. À la suite d'une commutation, les variables endogènes prennent pour valeur celle indiquée par le saut associé à la transition. Si la transition est associée à un événement contrôlant, elle ne peut être franchie et a pour obligation d'être franchie que si cet événement est observé et qu'elle est franchissable au moment de l'observation de cet événement. Si la transition est associée à un événement contrôlable, l'événement est émis au moment de la commutation. Si la transition est franchissable et que l'invariant associé à sa localité source (L_1) est sur le point de ne plus être vérifié, la transition doit aussi être franchie. Certains auteurs ajoutent pour toutes les gardes la notion d'obligation ou d'urgence de franchissement, cela signifie que la transition doit être franchie dès que le domaine de la garde inclut la valeur courante de toutes les variables et que la localité source de la transition est la localité courante ([10], [13], [18], [29], [25], [27]). Ces travaux font alors souvent abstraction de la notion d'invariant et d'événement.

2.2.3 Notions de chemin et d'exécution (path, run)

Afin de décrire l'évolution d'un automate hybride, la notion de trace est introduite, et plus précisément la notion de chemin et d'exécution.

Chemin (path) – Il représente le parcours à l'intérieur d'un graphe dont les nœuds sont les localités de l'automate hybride et les arcs orientés ses transitions. Durant son évolution, l'automate peut changer de mode de fonctionnement en franchissant une transition pour passer d'une localité à une autre, c'est une commutation. Le chemin est l'historique de l'ensemble de ces commutations, c'est une succession en alternance de localités et de transitions. Les chemins sont introduits sous la notion de "trace discrète" ([10]).

Exécution (run) – Elle représente l'évolution des variables endogènes lors de l'évolution de l'automate hybride. Une exécution contient alors toutes les valeurs prises par les variables endogènes durant l'évolution de l'automate. Des exemples de définition formelle d'exécutions peuvent être trouvés dans certains travaux ([7], [8]).

2.2.4 Déterminisme ou non-déterminisme du modèle

Dans le cas où un modèle serait utilisé pour du diagnostic de fautes, il peut être intéressant d'utiliser un modèle non déterministe qui représente un ensemble d'évolutions acceptables du

système ([7], [9], [8]). On nommera ce type de modèle, un modèle accepteur. Dans le cas où le modèle devrait représenter un système de commande, la commande étant généralement un processus déterministe, il est intéressant que son modèle le soit aussi ([13], [25], [27]). On nommera ce type de modèle, un modèle de commande.

2.3 Identification de système à un automate hybride

L'identification de système consiste en l'obtention d'un modèle d'une classe prédéfinie, ici les automates hybrides, à partir de données observées sur le système (mesures). En fonction du contexte des travaux considéré, les problèmes d'identification énoncés peuvent différer.

Identification passive ou active – Une méthode d'identification passive n'a pas d'influence sur l'évolution du système identifié, elle observe ses entrées et ses sorties en le laissant évoluer au gré des sollicitations de son environnement. À l'opposé, une méthode d'identification active sollicite les entrées du système et observe ses sorties pour en explorer le comportement. Les travaux sur l'identification de système hybride sous la forme d'automate hybride s'inscrivent principalement dans un contexte sous lequel l'identification doit pouvoir se faire sans perturber le fonctionnement du système. Bien qu'il existe des travaux portant sur l'identification active d'automate dans le domaine des systèmes à événements discrets, nous n'avons pas trouvé de travaux traitant spécifiquement le problème d'identification active d'automate hybride. De ce fait, les travaux présentés dans ce papier proposent uniquement de l'identification passive.

Identification en ligne (online) ou hors ligne (offline) – Une identification hors ligne fonctionne à partir d'un jeu unique d'observations ([8] [10], [13], [18], [29], [27]), tandis qu'une identification en ligne est capable de modifier le modèle déjà obtenu à partir d'un nouveau jeu d'observations ([7], [25], [9]). Il faut noter que si les identifications en lignes citées sont capables d'enrichir l'automate hybride existant (ajout de localité, de transitions...), elles ne permettent pas une restructuration plus profonde du modèle, qui réorganiserait la structure du graphe localité-transition.

3 Sous problèmes à résoudre pour l'identification de système à un automate hybride

L'identification de système sous la forme d'un automate hybride est non triviale et demande de résoudre plusieurs sous-problèmes interdépendants. Nous proposons ici un découpage du problème d'identification en 12 sous-problèmes dans lequel peuvent s'inscrire les travaux répertoriés et portant sur l'identification de systèmes sous la forme d'automates hybrides. Chacun de ces sous problèmes traite un problème particulier, allant de la préparation des données utilisées pour l'identification, en passant par le traitement de ces données, afin de déterminer les différents éléments constituant l'automate hybride, tels que ses transitions, ses localités, ses gardes, ses événements, ses sauts, ses lois d'évolution continues, et ses invariants. La Liste de ces sous-problèmes est présentée ci-dessous:

- | | |
|---|---|
| SP-1 – Acquisition des données | SP-7 – Identification des invariants |
| SP-2 – Sélection des données | SP-8 – Identification des gardes |
| SP-3 – Segmentation des données | SP-9 – Identification des sauts |
| SP-4 – Regroupement des segments | SP-10 – Identification des événements |
| SP-5 – Regroupement des dates de commutation | SP-11 – Inférence de la structure de l'automate |
| SP-6 – Identification des lois d'évolution continue | SP-12 – Validation de l'identification |

Chacun de ces sous problèmes peut être mis sous la forme d'un problème d'optimisation. Chaque problème est formalisé par ses entrées, ses sorties et sa méthode d'optimisation.

SP-1&2 – Acquisition et sélection des données

Définition du problème – Au moment de l'acquisition des signaux provenant du système à identifier, plusieurs problèmes peuvent être rencontrés. Le premier d'entre eux est le problème d'acquisition des données. Dus à des fréquences d'horloges différentes, les différents signaux peuvent être acquis avec une fréquence et un pas d'échantillonnage différents et parfois même avec un délai. Il est alors nécessaire de rendre chaque acquisition cohérente temporellement. Se pose de plus le problème de sélection des données. En lien étroit avec l'usage du modèle identifié, il est nécessaire de bien définir les entrées et les sorties du système. En prenant en compte les relations de causalité régissant l'évolution des différents signaux, une mauvaise affectation des entrées et des sorties peut mener à un échec du processus d'identification.

Exemple de méthodes utilisées – Il est supposé dans l'ensemble des travaux étudiés que ces problématiques sont des étapes en amont de la problématique d'identification. L'ensemble des séries temporelles observées sont synchronisées, échantillonnées d'un pas d'échantillonnage constant et labellisée en tant qu'entrée ou sortie du système. Cependant, nous pensons que la résolution de ces problèmes est fortement liée au problème d'identification. Les choix effectués lors de ce prétraitement pouvant avoir un fort impact sur le modèle identifié, ces problèmes font partie intégrante du problème d'identification d'automate hybride.

SP-3 – Segmentation des données

Définition du problème – L'objectif est de fournir, pour chaque série temporelle de chaque variable observée, un découpage en segments temporels correspondant à des lois d'évolution supposées différentes. Les instants séparant chaque segment sont nommés dates de changement. Ce problème est connu dans la littérature sous le nom de "changepoint detection". En entrée de ce sous-problème se trouve un ensemble de séries temporelles. En sortie de ce sous-problème se trouve un ensemble de segments et un ensemble de dates de changement.

Exemple de méthodes utilisées – Plusieurs méthodes sont utilisées afin de résoudre le problème de recherche des dates de changement, et pas seulement dans le domaine de l'identification d'automate hybride ([28]). Concernant l'identification sous la forme d'automate hybride, la majorité des méthodes reposent sur une méthode élémentaire de segmentation qui consiste à déterminer un changement abrupt de la valeur des variables ou de leurs dérivées ([7], [29], [13]). Ces méthodes sont peu coûteuses en calcul, mais sont à la fois fortement sensibles au bruit du signal et fortement décorréélées des formes de modèles des lois d'évolution identifiées par la suite. D'autres travaux utilisent des méthodes de fenêtres glissantes ([25], [10]). D'autres travaux dérivent les dates de changement du résultat de l'algorithme de Ramer-Douglas-Peucker ([8]) tandis que d'autres proposent leur propre heuristique basée sur une minimisation de critères, tel que le "Bayesian Information Criterion" ou la "Minimum Description Length" ([27]). Ces méthodes sont plus coûteuses, mais un peu moins sensibles au bruit et peuvent dans certains cas être adaptées au type de loi d'évolution que l'on cherche à obtenir par la suite.

SP-4 – Regroupement des segments (clustering)

Définition du problème – Un autre problème fondamental à résoudre est le problème de regroupement ou de clustering des segments. L'intérêt de modéliser un système sous la forme

d'un automate hybride vient du fait que le système observé évolue suivant un nombre fini de modes de fonctionnement différents. Lors de l'observation du système, ces modes de fonctionnement réapparaissent de manière récurrente. Il est intéressant de regrouper l'ensemble de ces modes de fonctionnement similaires, généralement dans une volonté d'avoir un modèle compact du système. Le problème de regroupement des segments peut être basé sur le regroupement de segments soumis à la même loi d'évolution, ou bien soumis au même invariant, ou même les deux. En entrée de ce sous-problème se trouve un ensemble de segments. En sortie de ce sous-problème se trouve un ensemble de groupes de segments.

Exemple de méthodes utilisées – Afin de résoudre ce problème, différentes solutions sont proposées. Certains travaux s'inspirent d'un algorithme de clustering K-mean sur les paramètres des lois d'évolution, couplé à des méthodes de détermination du nombre de clusters ([8], [18]). Un certain nombre de travaux s'intéressent à du clustering basé sur la distance fournie par des méthodes de "Digital Time Warping" ([25], [10]). Cette distance a l'avantage de considérer à la fois un écart spatial et temporel entre les séries temporelles simulées du modèle et celles mesurées sur le système. Une autre méthode consiste à résoudre un problème d'inégalité matricielle linéaire (LMI) pour déterminer l'intersection d'espaces de solution de loi de comportement sous la forme de polytopes ([29]). D'autres proposent leur propre heuristique ([7]) basée par exemple sur la vérification de "satisfiability modulo theories problem" ([9]), ou sur des critères statistiques comme l'information de Fisher ou la borne de Cramér-Rao ([13]), ou encore d'une minimisation de critères de pénalité ([27]).

SP-5 – Regroupement des dates de commutation

Définition du problème – De la même manière qu'il est pertinent de regrouper les segments susceptibles d'être modélisés par les mêmes lois d'évolution et les mêmes invariants, il est pertinent de regrouper les dates de changement pouvant être considérées comme des dates de commutation et être associées à la même garde, évènement ou saut. Dans le même esprit que le regroupement des segments, cela permet d'obtenir un modèle compact. En entrée de ce sous-problème se trouve un ensemble de dates de changement. En sortie de ce sous-problème se trouve un ensemble de groupe de dates de commutation.

Exemple de méthodes utilisées – Généralement, dans les travaux étudiés, le regroupement de dates de changement est un résultat direct du problème de regroupement de segments. Les dates de changements sont regroupées si elles ont le même groupe de segments qui les précède et le même groupe de segments qui les suit ([13], [25]). D'autres travaux proposent leur propre heuristique ([7], [8], [27]). Un autre papier utilise la méthode RANSAC ([29]).

SP-6 – Identification des lois d'évolution

Définition du problème – Afin de modéliser l'évolution continue des variables endogènes, la méthode d'identification doit déterminer la loi qui régit l'évolution des variables endogènes pour chaque segment ou groupe de segments préalablement identifiés. En entrée de ce sous-problème se trouve un groupe de segments. En sortie de ce sous-problème se trouve la loi d'évolution des variables endogènes pour l'ensemble des segments du groupe.

Exemple de méthodes utilisées – Différentes méthodes ont été proposées pour résoudre ce problème, la principale étant basée sur une estimation des paramètres d'une équation différentielle par régression ([10], [18], [25], [27]). D'autres utilisent l'intersection d'espace de solution d'équations différentielles ordinaires ([9], [7], [29], [8]), tandis que d'autres s'intéressent à la

détermination des paramètres d'une équation différentielle ordinaire en utilisant des critères statistiques comme l'information de Fisher et la limite de Cramér-Rao ([13]).

SP-7 – Identification des invariants

Définition du problème – En fonction de la variante retenue pour l'automate hybride, il est nécessaire d'identifier les invariants associés à chaque segment ou groupe de segments préalablement identifiés. En entrée de ce sous-problème se trouve un groupe de segments. En sortie de ce sous-problème se trouve l'invariant pour l'ensemble des segments du groupe.

Exemple de méthodes utilisées – Considérant les modèles comme évoluant sur gardes urgentes, peu de papiers se sont intéressés aux invariants ([10], [18], [29], [25], [27]). Ceux qui se sont intéressés à l'identification d'invariants l'ont directement dérivé de l'espace des valeurs prises par les variables sur le groupe de segments considéré. ([7], [9], [13], [8]).

SP-8 – Identification des gardes

Définition du problème – Un autre élément nécessaire à l'identification d'un automate hybride est l'identification des gardes qui devront être vérifiées au moment d'une commutation. En entrée de ce sous-problème se trouve un groupe de dates de commutation. En sortie de ce sous-problème se trouve la garde associée à l'ensemble des dates de commutation du groupe.

Exemple de méthodes utilisées – Afin de déterminer les gardes, plusieurs méthodes ont été utilisées. La méthode RANSAC introduite dans le clustering de date de commutation fournit aussi l'équation de garde ([29]). D'autres travaux se sont intéressés à l'intersection d'espaces de solution de garde sous la forme de polytopes pour trouver les paramètres d'une équation de garde ([9], [7]). Un papier s'est intéressé à l'identification de garde par inférence de condition temporelle ([18]), un autre s'est basé sur des "support vector machine" pour résoudre le problème ([10]). D'autres travaux ont développé leur propre heuristique pour trouver leurs gardes ([8], [13] [25], [27]).

SP-9 – Identification des sauts

Définition du problème – Quand cela est nécessaire, les sauts doivent aussi être identifiés. Un saut décrit l'initialisation des valeurs des variables endogènes à chaque commutation. En entrée de ce sous-problème se trouve un groupe de dates de commutation. En sortie de ce sous-problème se trouve le saut de chaque variable endogène à l'ensemble des dates de commutation du groupe.

Exemple de méthodes utilisées – Plusieurs travaux considèrent que leurs sauts sont implicites et maintiennent la valeur des variables endogène à la suite d'une commutation ([7], [9], [13], [18], [29], [25], [8], [27]). Seuls des travaux récents se sont intéressés à identifier les sauts à l'aide d'une méthode basée sur les "support vector machine" ([10]).

SP-10 – Identification des évènements

Définition du problème – Pour les automates hybrides qui les prennent en compte, les évènements qui doivent être associés aux transitions doivent aussi être déterminés. En entrée de ce sous-problème se trouve un groupe de dates de commutation. En sortie de ce sous-problème se trouve l'évènement qui se produit à l'ensemble des dates de commutation du groupe.

Exemple de méthodes utilisées – Dans certains travaux ayant pour but d’identifier des modèles accepteurs, aucun évènement n’est identifié ([7], [8], [9]). Dans les autres, les évènements sont confondus avec les gardes considérées comme urgentes ([10], [13], [18], [29], [25], [27]).

SP-11 – Inférence de l’automate hybride

Définition du problème – Le problème d’inférence de l’automate hybride consiste à obtenir la structure discrète de l’automate hybride. En superposant les résultats de ce sous-problème avec les résultats des précédents sous problèmes, un automate hybride peut alors être construit. En entrée de ce sous-problème peuvent se trouver les données sélectionnées, les groupes de segments avec leur loi d’évolution continue et leurs invariants, les groupes de dates de commutation avec leurs gardes, sauts et évènements. En sortie de ce sous-problème se trouve la structure discrète de l’automate hybride.

Exemple de méthodes utilisées – Beaucoup de travaux considèrent que la structure de l’automate est obtenue comme résultat direct des étapes de regroupement de segments (une localité par groupe de segment) et de regroupement des dates de commutation (une transition par groupe) ([7], [9], [10], [13], [8], [25]). Afin de préserver le déterminisme de l’automate hybride, une méthode basée sur le concept de "Prefix Tree Acceptor" a été proposée ([29]), ou encore une heuristique de regroupement de localité ([27]). Une autre méthode est basée sur une méthode d’inférence d’automate de Mealy: LearnLib ([18]).

SP-12 – Validation de l’identification

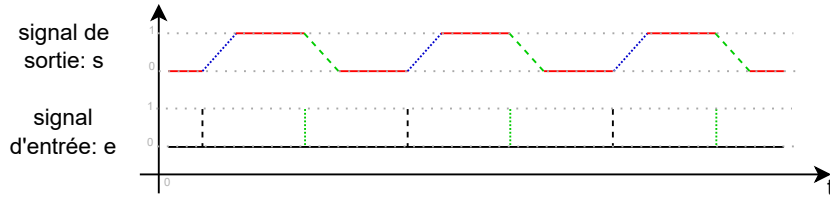
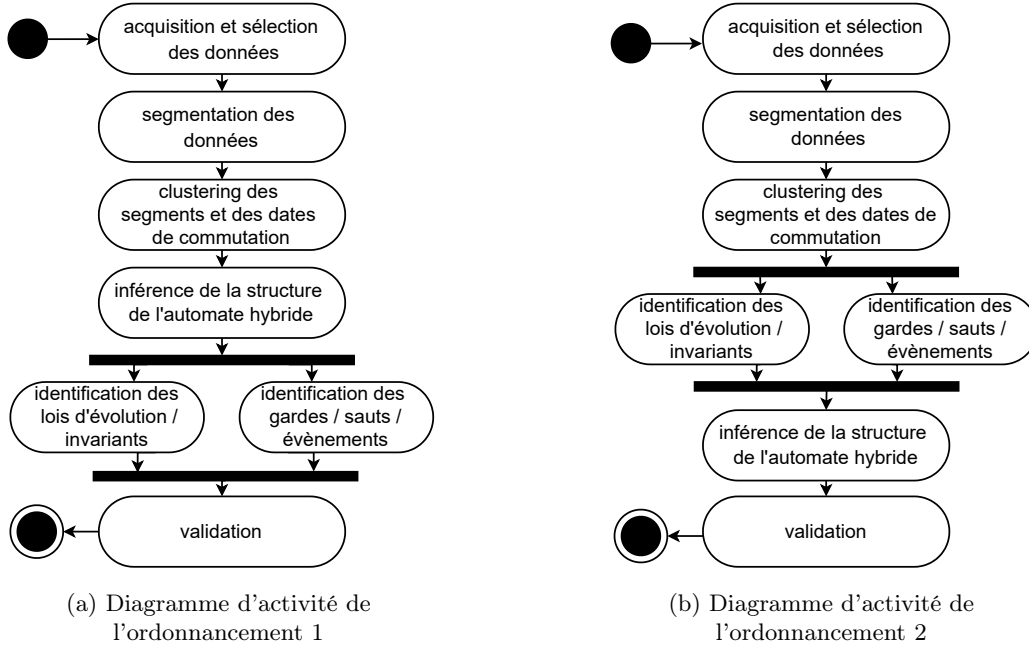
Définition du problème – Après avoir inféré l’automate hybride identifié, le modèle doit être comparé au système identifié à l’aide d’une métrique au préalable décidée par le modélisateur afin de vérifier que le modèle respecte les propriétés voulues.

Exemple de méthodes utilisées – Dans la littérature, en fonction du rôle porté au modèle, plusieurs validations sont proposées. Dans un objectif d’avoir un modèle dont la simulation est fidèle au système identifié, un grand ensemble des travaux s’intéresse à la distance maximale ou distance moyenne entre les sorties simulées du modèle identifié et les sorties mesurées sur le système sous la condition d’une même excitation du système et du modèle. La distance au sens du "Digital Time Warping" ([10],[25]) ou des moindres carrés ([13],[18]) peut être utilisée pour caractériser cet écart. D’autres travaux se sont intéressés au concept de "tube epsilon" caractérisant la capacité du modèle à accepter une évolution ou non en respectant une distance maximale "epsilon" au sens des moindres carrés ([7], [8], [9]). Ces caractéristiques sont d’ailleurs particulièrement adaptées à une fin de détection et identification de fautes sur le système. Enfin, d’autres travaux [29] se sont intéressés à la comparaison d’accessibilité d’états entre le modèle identifié et le système original ainsi qu’à un critère de conformité entre systèmes hybrides basé sur un écart spatio-temporel spécifique [1].

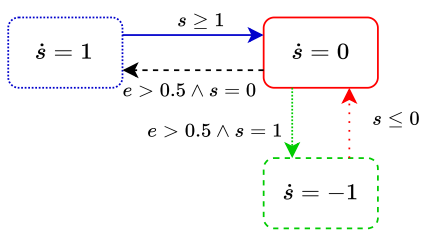
4 Ordonnancement de résolution des sous-problèmes

Dans les différents travaux étudiés, nous avons remarqué que l’ordre de résolution des sous problèmes n’est pas toujours le même. L’ordre de résolution des sous problèmes semble dépendre principalement de l’objectif de l’identification, des propriétés que doit avoir le modèle d’automate hybride recherché, ainsi que des méthodes utilisées pour résoudre chacun des sous problèmes. Cependant, en simplifiant légèrement, il est possible regrouper la manière dont les

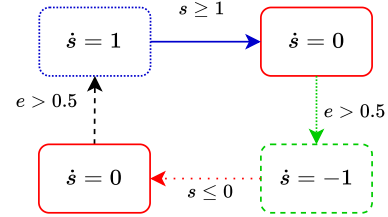
travaux ordonnant la résolution des sous problèmes en deux familles présentées figure 2, sous le nom d'ordonnement 1 (fig 2a) ([7], [8], [9], [10], [13], [18], [25]) et ordonnancement 2 (fig 2b) ([27], [29]).



(c) Exemple de signal segmenté et regroupé



(d) Exemple de modèle résultant de l'ordonnement 1¹



(e) Exemple de modèle résultant de l'ordonnement 2¹

Figure 2: Les 2 ordonnancements de résolution émergeant de la synthèse bibliographique

Globalement, l'ensemble des méthodes proposées commencent par supposer le problème d'acquisition et de sélection des données effectué en amont du processus d'identification. La sec-

onde étape est toujours l'étape de segmentation des données, chaque segment formé représente un comportement d'évolution continu des variables endogènes du système. Une fois que le signal est segmenté, il faut pouvoir regrouper les segments qui représentent la même loi d'évolution et les dates de commutation représentant les mêmes commutations. À partir de cette étape, la plupart des travaux considèrent que la structure de l'automate est déjà établie par les méthodes de regroupement (un groupe de segments par localité et un groupe de dates de commutation par transition) et il ne reste alors plus qu'à déterminer les gardes, les sauts, les lois d'évolution, les invariants et les événements (fig 2a). Cependant, certains travaux ([29], [27]) ont fait le choix de déterminer d'abord les gardes, les sauts, les lois d'évolution, les invariants et les événements avant de déterminer la structure de l'automate (fig 2b). Il nous semble en effet plus cohérent de déterminer la structure de l'automate après avoir effectué l'ensemble des étapes d'identification, principalement si l'on veut obtenir un modèle déterministe et facilement interprétable. Dans l'exemple donné figure 2c, une fois que les lois d'évolution, gardes, invariants, sauts et événements sont identifiés, il est possible de modifier la structure de l'automate afin de rendre l'automate déterministe (fig 2e). En adoptant l'ordonnancement 1 (fig 2a), le déterminisme ne peut être atteint qu'en adaptant les gardes (fig 2d), événements et invariants. Ce déterminisme risque alors d'être un artifice mathématique entraînant la perte du véritable sens de causalité des commutations et empêchant l'expression de variables internes cachées, modélisées par la structure de l'automate. Il peut de plus arriver que les méthodes d'identification de garde ne soient pas capables de déterminer des gardes déterministes, ce qui peut rendre dans certains cas l'obtention d'un modèle déterministe impossible.

5 Futurs défis

Au vu des différents travaux proposés, la problématique d'identification de systèmes sous la forme d'un automate hybride est jeune (premier papier datant de 2015). Les sous-problématiques liées à l'identification semblent globalement définies et à quelques détails prêts récurrents dans chacun des travaux. Cependant, la manière et l'ordre dans lequel ces sous-problématiques sont résolues pourraient encore être améliorés.

5.1 Coopération entre les résolutions de chaque sous problème

L'un des premiers points d'amélioration relevés provient de l'ordre de résolution des sous-problèmes. Les approches proposées tendent vers une résolution séquentielle des sous-problèmes, perdant au passage des informations utiles pour la résolution de chacun d'entre eux. Chaque imprécision de modélisation effectuée par chacune des étapes est propagée aux étapes suivantes sans pouvoir être corrigée. Par exemple, les méthodes proposées pour déterminer des dates de commutation ne se basent que sur un traitement des séries temporelles données en entrée de la méthode d'identification. De cette manière, en fonction des paramètres, seuils utilisés, ou même du bruit du signal, cette étape d'identification n'est pas à l'abri de fournir un ensemble de dates de commutation en trop, en moins, ou légèrement décalées par rapport aux réelles dates de commutation du système. Il pourrait alors être intéressant de développer des méthodes de détermination des dates de commutation utilisant les informations résultantes de l'étape de détermination des lois d'évolution, des sauts, des gardes, des invariants, des événements et même de la structure de l'automate, afin d'améliorer la précision de cette étape d'identification. En allant plus loin, il pourrait être intéressant de développer chacune des étapes d'identification

¹Afin d'alléger le graphique, les sauts, événements et invariants n'ont pas été représentés

pour que chacune d'entre elles base son résultat sur l'optimisation d'un critère global, calculé à partir des résultats de chacune des étapes (fig 3a). Par exemple, ce critère pourrait être la minimisation du nombre de relations 'causes/effets' présentes dans le modèle identifié (cela correspond au nombre de triplets de garde/saut/loi d'évolution successifs). Ce critère pourrait aussi être une minimisation du nombre de localités et transitions pour faciliter l'interprétabilité du modèle identifié, ou encore une minimisation du nombre de lois d'évolution, de gardes ou de sauts différents. Cela aurait sûrement pour défaut de rendre la méthode d'identification plus lente, mais permettrait d'obtenir une méthode d'identification plus robuste, reconsidérant des choix de modélisation peu judicieux qui ont pu être retenus à un moment où peu de données relatives au modèle identifié étaient disponibles. Un exemple d'ordre de résolution itératif répondant à ce problème est présenté figure 3a. Dans l'exemple donné figure 3b, est présenté le cas où l'on cherche à réduire le nombre de lois d'évolution différentes. Il est possible grâce à la méthode itérative de supprimer des dates de commutation afin de réduire le nombre de lois d'évolution à un nombre de 2, ce qui fournit le nouveau modèle déterministe figure 3c.

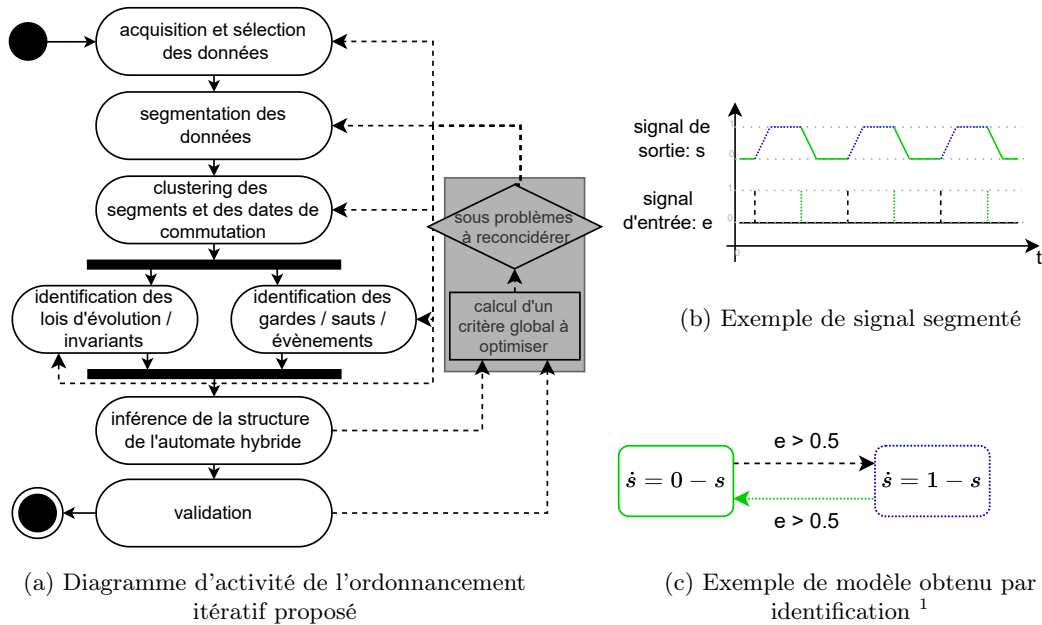


Figure 3: Exemple de nouvel ordonnancement itératif proposé

5.2 Identification évolutive

Dans le contexte de plus en plus présent du jumeau numérique, la nécessité de faire évoluer le modèle durant le cycle de vie du système pose la question de l'identification évolutive. Il s'agit de faire évoluer le modèle au cours du temps pour que ce dernier continue à représenter fidèlement le système dont les évolutions de comportement peuvent venir de dégradations dues à son vieillissement ou de fonctionnalités ajoutées ou supprimées au gré des besoins. Ce problème est connu en tant que problème d'obtention d'un "Digital Shadow" du système [24]. Il pourrait être intéressant d'adapter les méthodes d'identification de système sous la forme d'un automate hybride, afin qu'en réaction à l'ajout de nouvelles données (représentant potentielle-

ment une nouvelle dynamique du système) ou la suppression d’anciennes données (représentant potentiellement d’anciens comportements obsolètes), la méthode d’identification puisse mettre à jour le modèle d’automate hybride sans le reconstruire de zéro, à l’instar de l’identification en ligne, mais sans faire l’hypothèse que le comportement du système est invariant [19].

6 Conclusions

Nous avons proposé une revue des travaux récents (2015-2023) portant sur l’identification de système sous la forme d’un automate hybride. Nous avons défini les concepts relatifs à la définition d’un automate hybride et son exécution. Nous avons ensuite segmenté le problème d’identification en 12 sous-problèmes transversaux à l’ensemble des travaux étudiés. Nous avons analysé comment ces sous-problèmes sont résolus dans les différents travaux et l’ordre dans lequel ils y sont résolus. Enfin, nous avons proposé une courte discussion sur de futurs défis à relever pour obtenir des modèles toujours plus fidèles et suivant l’évolution du système, permettant de répondre par exemple aux besoins actuels des jumeaux numériques.

References

- [1] Houssam Abbas, Hans Mittelmann, and Georgios Fainekos. Formal property verification in a conformance testing framework. In *2014 Twelfth ACM/IEEE Conference on Formal Methods and Models for Codesign (MEMOCODE)*, pages 155–164, Lausanne, Switzerland, 2014. IEEE.
- [2] W. Ai, N. Patel, P. Roop, A. Malik, S. Andalamp, E. Yip, N. Allen, and M. Trew. A Parametric Computational Model of the Action Potential of Pacemaker Cells. *IEEE Transactions on Biomedical Engineering*, 65(1):123–130, 2018.
- [3] R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *Theoretical Computer Science*, 138(1):3–34, 1995.
- [4] R. Bohrer. Chemical Case Studies in KeYmaera X. In *Proceedings of Formal Methods for Industrial Critical Systems*, Lecture Notes in Computer Science, pages 103–120, Warsaw, Poland, 2022.
- [5] L. Bu, J. Wang, Y. Wu, and X. Li. From Bounded Reachability Analysis of Linear Hybrid Automata to Verification of Industrial CPS and IoT. In *5th SETSS*, Chongqing, China, 2019.
- [6] L. Desgeorges, P.-Y. Piriou, T. Lemattre, and H. Chraïbi. Formalism and semantics of PyCAT-SHOO. *Reliability Engineering & System Safety*, 2021.
- [7] M. García Soto, T. Henzinger, and C. Schilling. Synthesis of hybrid automata with affine dynamics from time-series data. In *Proc. of HSCC2021*, Nashville Tennessee, 2021.
- [8] M. García Soto, T. Henzinger, and C. Schilling. Synthesis of Parametric Hybrid Automata from Time Series. In *Proc. of ATVA*, Beijing, China, 2022.
- [9] M. García Soto, T. Henzinger, C. Schilling, and L. Zeleznik. Membership-Based Synthesis of Linear Hybrid Automata. In *Proc. of CAV2019*, pages 297–314, New York, USA, July 2019.
- [10] A. Gurung, M. Waga, and K. Suenaga. Learning nonlinear hybrid automata from input–output time-series data, 2023. arXiv:2301.03915.
- [11] Z. Huang, D. Chu, C. Wu, and Y. He. Path Planning and Cooperative Control for Automated Vehicle Platoon Using Hybrid Automata. *IEEE trans Intell Transp Syst*, 20(3):959–974, 2019.
- [12] S.K. Jayaraman, L.P. Robert, X.J. Yang, and D.M. Tilbury. Multimodal Hybrid Pedestrian. *IEEE Access*, 9:27708–27722, 2021.
- [13] I. Lamrani, A. Banerjee, and S.K.S. Gupta. HyMn: Mining linear hybrid automata from input output traces of cyber-physical systems. In *Proc of IEEE ICPS*, St. Petersburg, Russia, 2018.

- [14] L. Lestingi, M. Askarpour, M.M. Bersani, and M. Rossi. Formal Verification of Human-Robot Interaction in Healthcare Scenarios. In *Proc. of SEFM 2020*, Amsterdam, Netherlands, 2020.
- [15] Q. Lin, S. Verwer, and J. Dolan. Safety Verification of a Data-driven Adaptive Cruise Controller. In *2020 IEEE Intelligent Vehicles Symposium*, pages 2146–2151, 2020.
- [16] L. Liu and A. Bockmayr. Formalizing Metabolic-Regulatory Networks by Hybrid Automata. *Acta Biotheoretica*, 68(1):73–85, 2020.
- [17] C. Lozoya, A. Favela-Contreras, A. Aguilar-Gonzalez, and L. Orona. A Precision Irrigation Model Using Hybrid Automata. *Transactions of the ASABE*, 62(6):1639–1650, 2019.
- [18] R. Medhat, S. Ramesh, B. Bonakdarpour, and S. Fischmeister. A framework for mining hybrid automata from input/output traces. In *2015 IEEE EMSOFT*, Amsterdam, Netherlands, 2015.
- [19] Y. Monier, G. Faraut, B. Denis, and N. Anwer. Inferring Moore Machine for adaptive online Hybrid Automaton Identification. In *Proc. of IFAC World Congress*, Yokohama, Japan, 2023.
- [20] K. Naftahi, A. Abouloifa, Z. Hekss, S. Echalih, F. Ait bellah, and I. Lachkar. Three-Phase Four-Wire Shunt Active Power Filter Based on the Hybrid Automaton Control with Instantaneous Reactive Power Theory. In *Proc. of 14th IFAC ALCOS 2022*, Casablanca, Morocco, 2022.
- [21] L. Nasti and P. Milazzo. A Hybrid Automata model of social networking addiction. *Journal of Logical and Algebraic Methods in Programming*, 100:215–229, 2018.
- [22] E.M. Navarro-López and N. Çabukoğlu. A Novel Hybrid Automaton Framework for Multi-Phase Epidemic Modelling. In *Proc. of ANNSIM*, Fairfax, VA, USA, 2021.
- [23] A. Ostroukh, L. Berner, M. Karelina, P. Kravchenko, and Se. Evtiukov. Intelligent control system for distributed gas transport facilities. *Transportation Research Procedia*, 57:376–384, 2021.
- [24] Q. Qi, F. Tao, T. Hu, N. Anwer, A. Liu, Y. Wei, L. Wang, and A.Y.C. Nee. Enabling technologies and tools for digital twin. *Journal of Manufacturing Systems*, 58:3–21, 2021.
- [25] I. Saberli, F. Faghil, and F.S. Babil. A Passive Online Technique for Learning Hybrid Automata from Input/Output Traces. *ACM Trans. on Embedded Computing Systems*, 22(1):9:1–9:24, 2022.
- [26] V. Sinyakov and A. Girard. Formal controller synthesis from specifications given by discrete-time hybrid automata. *Automatica*, 131:109768, September 2021.
- [27] A. Summerville, J. Osborn, and M. Mateas. CHARDA: Causal Hybrid Automata Recovery via Dynamic Analysis. In *Proc. of the 26th Int Joint Conf on AI*, Melbourne, Australia, 2017.
- [28] C. Truong, L. Oudre, and N. Vayatis. Selective review of offline change point detection methods. *Signal Processing*, 167, 2020.
- [29] X. Yang, O. Beg, M. Kenigsberg, and T. Johnson. A Framework for Identification and Validation of Affine Hybrid Automata from Input-Output Traces. *ACM Trans. on Cyber-Physical Syst*, 2021.
- [30] Y. Yuan, X. Tang, W. Zhou, W. Pan, X. Li, H.T. Zhang, H. Ding, and J. Goncalves. Data driven discovery of cyber physical systems. *Nature Communications*, 10(1), 2019.
- [31] C. Yuehua, J. Liang, J. Bin, and L. Ningyun. Useful life prediction using a stochastic hybrid automata model for an ACS multi-gyro subsystem. *J. Syst. Eng.*, 30(1), 2019.
- [32] L. Zadeh. From Circuit Theory to System Theory. *Proc. of the IRE*, 50(5):856–865, 1962.
- [33] F. Zahn and V. Hagenmeyer. Modelling power systems as flat hybrid automata for controlled line switching. In *Proc. of ACM e-Energy'22*, New York, NY, USA, 2022.