

## ***SUIVI DE L'ACTIVITE DE PILOTAGE PAR PREDICTION ET RECALAGE***

---

**Charles Lesire,**

Doctorant du laboratoire de l'ONERA-DCSD  
Charles.Lesire@cert.fr , + 33 5 62 25 22 12

**Adresse professionnelle**

ONERA-CERT, DCSD ★ 2 av. Edouard Belin ★ 31055 Toulouse Cedex

**Résumé** : La revue d'accidents aéronautiques tant civils que militaires a montré que l'apparition de conflits dans la gestion du vol est un facteur de dégradation de l'activité et est à l'origine de nombreux accidents. Ces conflits rendent souvent le pilote non réceptif aux alarmes classiques. Un nouveau système d'assistance au pilotage, GHOST, est en cours d'étude et permet de résoudre certains conflits par l'envoi de contre-mesures pertinentes à l'équipage. Ce système nécessite, pour être efficace, un bon suivi de l'activité de pilotage. Après une présentation du système GHOST et des différentes méthodes de suivi d'activités, nous présentons le modèle de représentation des procédures, fondé sur le formalisme des réseaux de Petri à objets, puis les principes de *prédiction* et *recalage* qui assurent le suivi de l'activité de pilotage.

**Summary** : The review of civilian and military accident reports has shown that a conflictual situation is a precursor to the loss of aircrew's situation awareness and is a major cause of air accidents. Conflicts often prevent pilots from being receptive to classic warnings. A new system for pilot assistance, GHOST, is under study to solve some of the conflicts by sending the crew relevant countermeasures. This system is grounded on the monitoring of the pilot's activity. GHOST and different methods for activity tracking are first presented then the Petri net based model of aeronautical procedures is described and finally the prediction and correction principles designed to track the pilot's activity are explained.

**Mots clés** : Réseaux de Petri, suivi d'activité, aide au pilotage.

# Suivi de l'activité de pilotage par prédiction et recalage

## 1 - INTRODUCTION

L'étude des rapports d'accidents aéronautiques (BEA, 1996) confirmée par des expérimentations en simulateur (Dehais, 2002) ont permis de mettre en avant des cas de *conflits* menant le pilote à persévérer dans de mauvais choix. Le phénomène de persévération est dû à un stress important (Cohen, 1993) et il est caractérisé par une perte de facultés cognitives du sujet : celui-ci se focalise sur son conflit et ne peut plus en retirer son attention. Dans de nombreux cas les alarmes classiques (visuelles ou auditives) sont inefficaces du fait de la focalisation du pilote sur un instrument ou une situation.

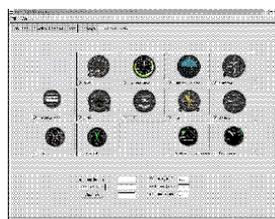
L'idée du système GHOST (Dehais, Tessier, Chaudron, 2003) est de mettre en place des contre-mesures afin d'enrayer le mécanisme de persévération et sortir le pilote de son conflit. Le principe des contre-mesures est fondé sur le retrait d'information : puisque le pilote ne peut détourner son attention volontairement du conflit, GHOST permet de faire disparaître l'objet sur lequel le pilote se focalise, de sorte qu'il puisse prendre conscience de la situation extérieure (alarmes, paramètres, contrôle aérien, etc.)

GHOST est constitué :

- d'un simulateur de vol<sup>1</sup> installé sur un portable muni d'un joystick ; le simulateur a été modifié de façon à permettre la disparition ou le clignotement d'instruments et l'affichage de messages ;
- d'un logiciel de suivi de l'avion sur une carte aéronautique<sup>2</sup> ;
- d'une interface de magicien d'Oz, permettant de modifier à distance les paramètres de vol et la météo, et d'envoyer des contre-mesures.



Simulateur FlightGear



Interface de magicien d'Oz

FIG. 1: Environnement GHOST

Après une série d'expérimentations qui ont montré la pertinence de l'envoi de contre-mesures (Dehais,

<sup>1</sup> FlightGear, open source <http://www.flightgear.org>

<sup>2</sup> Atlas, freeware <http://atlas.sourceforge.net>

Tessier, Chaudron, 2003), il s'agit maintenant de modifier GHOST en vue de l'envoi automatique de contre-mesures en fonction de l'activité de pilotage courante et prédite.

Le travail présenté ici concerne donc les principes et la mise en œuvre du suivi de l'activité de pilotage.

## 2 - SUIVI DE SITUATIONS

Cette section décrit différentes méthodes de reconnaissance d'actions ou de situations, principes sur lesquels est fondé le suivi de l'activité du pilote. Les recherches de Brémond (1997) et Rota et Thonnat (2000) portent sur la reconnaissance d'actions dans des séquences d'images et sont fondées sur la mise en correspondance d'événements observés avec des scénarios composés de listes d'événements et de contraintes portant sur les événements (logiques) ou leurs dates d'occurrence (numériques). L'approche proposée par Dousson (1991) et Ghallab (2002) utilise les *chroniques* dans la reconnaissance d'événements : elle est fondée sur un graphe temporel représentant les contraintes entre les différents événements (relations de précédence temporelle) et la phase de prédiction consiste en une propagation de contraintes sur ce graphe. Largouët et Cordier (2000) confrontent une suite d'observations (images satellites) avec des états références et mettent en œuvre un principe de prédiction et de postdiction pour réduire le nombre d'états retenus. Le projet CATS<sup>3</sup> (Callantine, 2002) est destiné au suivi de l'activité de pilotage et à l'analyse en ligne des performances de l'équipage. Un modèle d'activité est représenté par un arbre dont la racine est l'activité et ses fils les actions à réaliser. Lorsqu'une action est reconnue, le système repère les activités susceptibles d'être entreprises par le pilote et attend de celui-ci l'action suivante de façon à affiner la reconnaissance. La mesure des temps d'attente entre deux activités et l'écart par rapport à des temps de référence permet de donner une allure de la performance du pilote.

Nos travaux sont directement issus de l'estimateur symbolique Kalmansymbo - appelé ainsi par analogie avec le filtre de Kalman, (Cossart-Jaupitre, 1999 ; Tessier, 2003) - dont le but est d'élaborer et de mettre à jour une description de l'état d'un environnement (appelée situation), en fonction d'observations qui sont effectuées sur cet environnement grâce à des capteurs dits

<sup>3</sup> Crew Activity Tracking System

«symboliques» (c'est-à-dire qui délivrent des informations de même nature que celles que délivrerait un observateur humain). Une situation est un ensemble d'états possibles de l'environnement, ordonnés selon une relation de préférence : les états préférés sont ceux qui correspondent le plus vraisemblablement à l'état (réel) de l'environnement, les états moins préférés sont ceux qui correspondent moins à cet état. L'estimation est effectuée à partir de modèles d'évolution de l'environnement (plans) représentés sous forme de réseaux de Petri. L'estimation est un processus en deux étapes (voir figure 2) : (1) la prédiction permet, à partir d'une situation (donc une relation de préférence sur des états) et des modèles d'évolution (réseaux de Petri), de calculer la situation prédite, c'est-à-dire une relation de préférence sur les états qu'on est susceptible d'observer par la suite ; (2) le recalage permet, à partir d'une situation prédite et d'une observation, de calculer une nouvelle situation, plus conforme à l'observation que ne l'était la situation prédite.

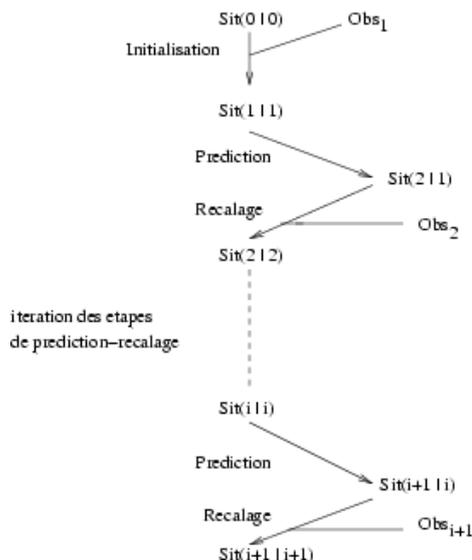


FIG. 2: Processus d'estimation de Kalmansymbo

### 3 - KALMANSYMBO-AÉRO

L'idée est d'appliquer les principes de Kalmansymbo à GHOST pour suivre et prédire l'activité de pilotage (Kalmansymbo-aéro), et aider ainsi à la prévention des conflits. Les particularités des procédures aéronautiques requièrent des adaptations de ces principes, qui prendront notamment en compte la présence de variables continues et de boucles dans les réseaux de Petri.

#### 3.1 - Exemple la mission Refuel

La mission servant à expérimenter les hypothèses faites lors de l'élaboration de Kalmansymbo-aéro sera la mission *Refuel* (figure 3).

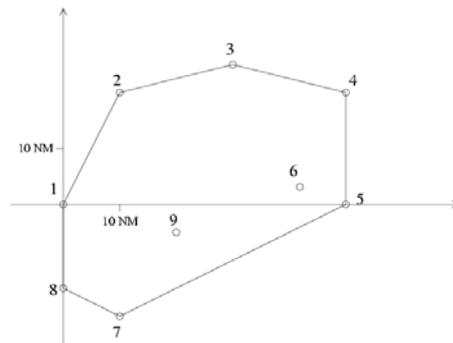


FIG. 3: Mission Refuel

Le pilote doit suivre le plan de vol suivant : décoller (la piste se trouve entre 1 et 8 et orientée vers 1), passer, dans l'ordre, par les points tournants 1, 2, 3, 4, 5, 7 et 8, et atterrir en 1. Le pilote a la possibilité de se ravitailler en passant à la verticale des points 6 et 9. Il peut se diriger vers 6 s'il est entre 4 et 5 ou entre 5 et 7. Il peut se diriger vers 9 s'il est entre 5 et 7.

#### 3.2 - Modélisation

Le formalisme que nous avons choisi est celui des réseaux de Petri à objets, dont la définition est issue de Valette (2000). Les paramètres de vol (latitude, longitude, vitesse, fuel, consommation) sont encapsulés dans le *jeton-mission*, unique instance de la classe *Jeton*.

#### Places et transitions

Comme introduit dans Tessier (2003), une activité est représentée par un *cube* (Chaudron, Maille, Boyer, 2003), conjonction de prédicats, associé à une place du réseau. Dans le cas de la mission *Refuel*, toutes les activités seront représentées par le prédicat d'arité 2 : *waypoint*. Ce prédicat indique la direction de l'avion (vers quel point tournant il vole) et si l'avion est considéré comme étant à la verticale de ce point. Ainsi *waypoint(2,unreached)* indique que l'avion se dirige vers le point tournant 2 et qu'il ne l'a pas atteint, alors que *waypoint(1,reached)* indique que l'avion a atteint le point tournant 1 (voir figure 4).

Les transitions des réseaux à objets possèdent une condition et une action de franchissement. On note graphiquement ces attributs en étiquetant les transitions sous la forme « *c / a* » où *c* est un cube de conditions et *a* une conjonction d'actions. La condition de passage entre activités représente la capacité de l'avion à se rendre au point suivant. Ce paramètre est dans notre cas modélisé par la quantité de carburant disponible et est représenté par le prédicat *fuel* : *fuel(2,OK)* indique que la quantité de carburant est suffisante pour atteindre le point tournant 2. Dans le cas inverse, la condition de passage est *fuel(2,BAD)*. L'action est représentée par le prédicat *goToWp*. Celui-ci agit sur le *jeton-mission* et modifie ses attributs pour

simuler un déplacement (dans le cas de  $goToWp(2)$ ), le déplacement de 1 à 2). La figure 4 illustre ce modèle :

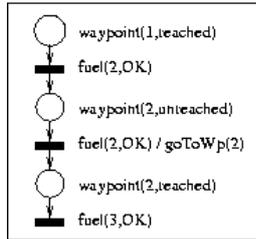


FIG. 4: *Modèle partiel de la mission Refuel*

### Les transitions de changement de but

Certaines transitions sont particulières car elles représentent un changement de but (et donc un conflit cognitif potentiel). Ces transitions correspondent à la sélection d'un nouveau but : un point tournant à atteindre différent de celui qui était initialement visé. Pour les différencier des transitions classiques représentées par un rectangle plein, on représente les transitions de changement de but par un rectangle vide. Dans la figure 5, la transition de changement de but est présente pour modéliser l'abandon par le pilote du but 5 ( $waypoint(5,unreached)$ ) pour se diriger vers le point tournant 6 ( $waypoint(6,unreached)$ ). La particularité de ces transitions est expliquée dans le cadre de la phase de prédiction (voir section 4).

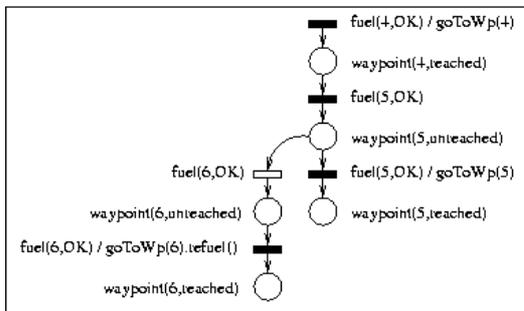


FIG. 5: *Transition de changement de but*

### Les boucles

Lors d'un vol, un pilote peut être amené à accomplir plusieurs fois la même activité. C'est pourquoi il est indispensable de gérer les boucles dans les réseaux de Petri. Les boucles posent cependant des problèmes dans la phase de prédiction au cours de laquelle elles peuvent théoriquement être parcourues une infinité de fois. On décide donc, dans le cadre de cette étude, de limiter le nombre de parcours des boucles à 1. On peut donc «casser» les boucles en dupliquant certaines places.

La figure 6 est un exemple de sous-réseau comportant une boucle, qui correspond au détournement de l'avion pour se ravitailler en 6 : le pilote se dirige vers 5 ( $waypoint(5,unreached)$ ), puis change de but pour aller en 6 ( $waypoint(6,unreached)$ ), se ravitaile

( $waypoint(6,reached)$ ), et finalement reprend sa route vers 5 ( $waypoint(5,unreached)$ ). On remplace ce modèle par celui de la figure 7, dans lequel l'activité  $waypoint(5,unreached)$  a été dupliquée pour distinguer les phases avant et après ravitaillement.

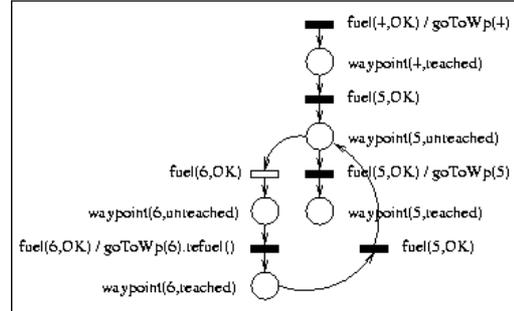


FIG. 6: *Boucle*

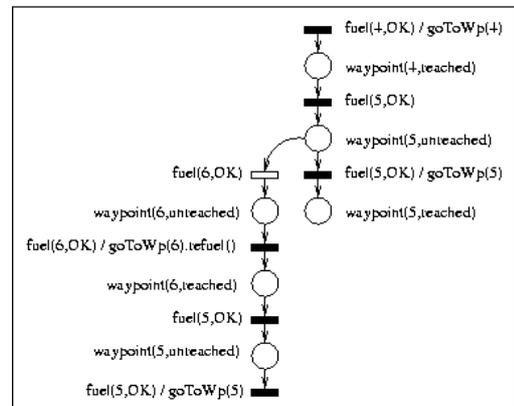


FIG. 7: *Boucle dépliée*

### Le temps

Afin de traiter des problèmes réels, qui font intervenir des paramètres continus, on introduit ici une notion temporelle. On assigne au *jeton-mission* une horloge, qui est modifiée par les actions des transitions pour simuler un déplacement de l'avion. L'utilisation de cette horloge sera exposée dans la section traitant de la phase de prédiction (section 4).

## 4 - PRÉDICTION

La prédiction permet, à partir d'une situation reconnue (modélisée sous forme de réseau de Petri marqué), de calculer la situation prédite, c'est-à-dire une relation de préférence sur les états qu'on est susceptible d'observer par la suite. Ce calcul se fait en construisant l'arbre des marquages accessibles du réseau.

Dans le cas de la mission *Refuel*, on peut constater que seul le *jeton-mission* est présent dans le réseau. Le marquage se limite donc à un seul jeton. On notera par  $\bullet$  la place marquée et par  $\times$  les places prédites.

#### 4.1 - Les transitions de changement de but

Dans le cas idéal, un pilote ne change pas de but. Il prévoit sa route lors de la préparation du vol. Ainsi, lors de la phase de prédiction, on suit ce modèle «idéal» et on ne prédit pas le franchissement des transitions de changement de but, sauf dans le cas où la place immédiatement précédente est *marquée* (avec un jeton ●).

Dans le cas de la mission *Refuel*, on laisse au pilote la possibilité d'aller se ravitailler (en 6 ou 9). La première transition de changement de but rencontrée est donc celle qui permet au pilote de se dérouter pour aller se ravitailler en 6. Ainsi, lorsque l'activité *waypoint(5,unreached)* est prédite, la transition de changement de but est non franchissable pour la prédiction (figure 8). Elle le devient lorsque cette activité est marquée (figure 9).

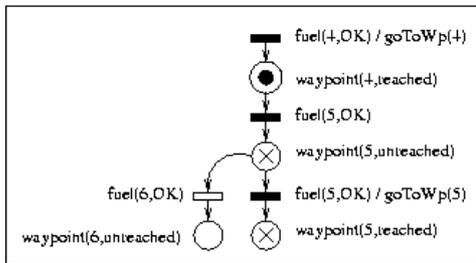


FIG. 8: Transition de changement de but non franchissable en prédiction

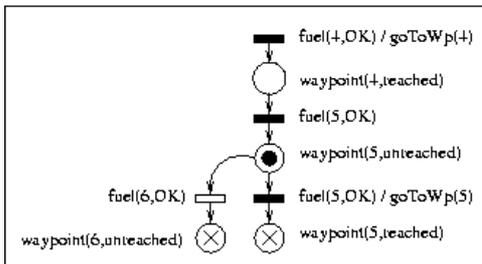


FIG. 9: Transition de changement de but franchissable en prédiction

#### 4.2 - Intervalles de prédiction

L'horloge du *jeton-mission* est modifiée par les actions des transitions. Par exemple, lors du franchissement de la transition qui porte l'action *goToWp(3)* (mission *Refuel*), le *jeton-mission* subit les actions suivantes :

- modification de ses coordonnées (déplacement géographique)  
 $position := coordonnées(3);$
- modification de son horloge (déplacement temporel)  
 $horloge := horloge + tempsPourAllerA(3);$
- consommation de carburant (déplacement temporel)  
 $niveau := niveau - conso * tempsPourAllerA(3).$

L'horloge du *jeton-mission* indique l'heure à laquelle l'avion est prévu dans une activité. Cette heure est communiquée à la place prédite. On

obtient donc un intervalle construit à partir de l'ensemble des heures où le jeton est prévu dans la place. On nommera ces intervalles les *intervalles de prédiction*. La figure 10 illustre ce principe :

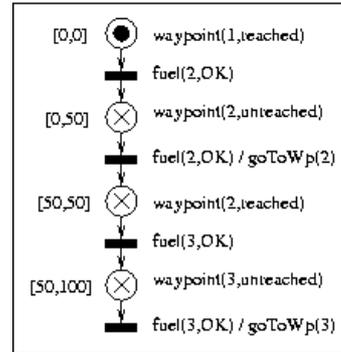


FIG. 10: Intervalles de prédiction

L'état initial correspond à l'activité *waypoint(1,reached)* à  $t=0$ . La première transition ne modifie pas la valeur de l'horloge du jeton car elle n'a pas d'action. L'intervalle de la première place est donc  $[0,0]$ . Le jeton entre dans la deuxième place à  $t=0$ , puis franchit la deuxième transition : son horloge est modifiée,  $t=50$ , temps nécessaire pour aller de 1 à 2, étant donné la vitesse de l'avion. On considère donc que le jeton a quitté sa place à  $t=50$ , et on construit l'intervalle  $[0,50]$ . On construit de la même façon les intervalles suivants.

Lors des phases de prédiction successives, on agrandit ces intervalles en y ajoutant les valeurs de l'horloge. Ainsi, une place possédant l'intervalle  $[0,50]$  et à laquelle la prédiction indiquerait un intervalle de  $[20,60]$  se verrait attribuer l'intervalle de prédiction  $[0,60]$ . L'intervalle de prédiction est toujours un intervalle (pas d'union d'intervalles et/ou de singletons disjoints).

#### 4.3 - Horizon de prédiction

La présence d'une horloge lors de la phase de prédiction permet de définir une nouvelle notion : l'*horizon de prédiction*. C'est une valeur temporelle qui permet de limiter la construction de l'ensemble des places accessibles : on arrête la construction lorsque l'horloge du *jeton-mission* va au-delà de l'horizon.

Dans le cas de la mission *Refuel*, on fixe la valeur de l'horizon à 3 mn. Cette valeur est pertinente car elle permet de prévoir les actions du pilote sur 4 à 5 activités, ce qui correspond environ à deux sélections de points tournants. On a ainsi un bon aperçu des états susceptibles d'être atteints sans surcharger l'ensemble des places accessibles.

#### 4.4 - Principe de prédiction

Si dans un premier temps, le fait de cocher sur un réseau l'ensemble des places prédites était suffisant, il faut maintenant revoir ce principe. Les réseaux n'étant pas linéaires, une place peut être marquée

plusieurs fois et donc posséder plusieurs intervalles de prédiction. On pourrait, comme on le fait lors de phases de prédiction successives, faire l'union de ces intervalles.

Cependant, une place souvent marquée aurait un intervalle de prédiction beaucoup trop grand, ce qui enlèverait tout l'intérêt de l'intervalle. De plus, les intervalles de prédiction seront utilisés lors de la mise en correspondance et leur construction a de l'importance.

On décide, plutôt que d'agrandir l'intervalle de prédiction, de dupliquer le réseau. Lorsqu'on est confronté à un choix (conflit au sens des réseaux de Petri), on crée une copie du réseau pour chaque transition franchissable.

Le problème qui subsiste alors est le suivant : lors de la phase de recalage suivante, il est possible que l'observation se mette en correspondance avec une place située en amont du conflit. On est alors amené à réitérer le processus, créant ainsi encore plus de réseaux (figure 11).

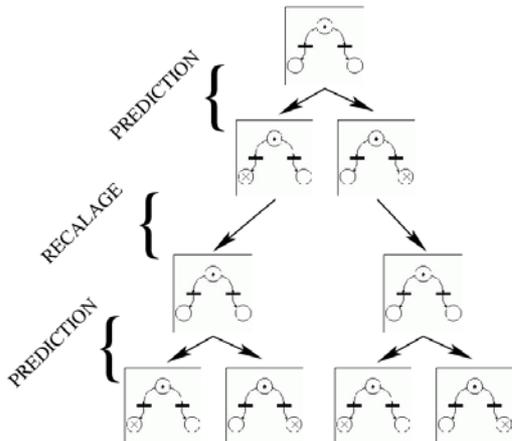


FIG. 11: Multiplication des réseaux en présence d'un conflit

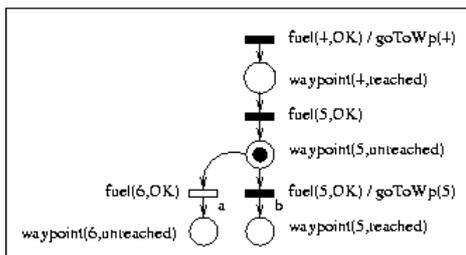


FIG. 12: Prédiction multiple : état initial

On remédie à ce problème en supprimant de chaque réseau les transitions conflictuelles, excepté celle qui est franchie. On obtient ainsi plusieurs réseaux sans interférence des intervalles de prédiction. Les figures 12, 13 et 14 illustrent ce principe : à l'état initial, on est confronté à un conflit mettant en cause deux transitions (notées *a* et *b*). On va donc construire deux réseaux correspondant chacun à une de ces deux transitions.

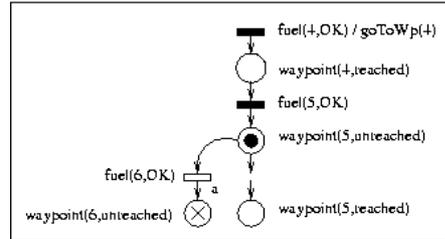


FIG. 13: Prédiction multiple : premier chemin

Le réseau de la figure 13 a été privé de la transition *b*. On continue donc la prédiction en franchissant la transition *a*.

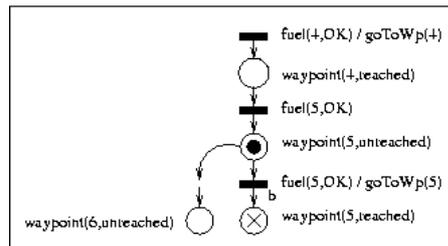


FIG. 14: Prédiction multiple : second chemin

On procède inversement pour le réseau de la figure 14.

## 5 - RECALAGE

La phase de recalage dispose des plans prédits et d'une nouvelle observation. La première étape est la *mise en correspondance* qui permet de reconnaître les activités correspondant à l'observation.

### 5.1 - Observation

Une observation est représentée par un objet dont la classe hérite de la classe *Jeton*. L'horloge indique l'heure de l'observation. L'observation possède de plus un cube de propriétés représentant l'état observé.

### 5.2 - Initialisation

Au lancement du système, celui-ci dispose d'une banque de plans non marqués représentant les plans de vol que le pilote peut suivre. Lors de l'arrivée de la première observation, les plans pouvant correspondre à l'activité observée sont alors sélectionnés.

Chaque place représente une activité et à ce titre porte un cube de propriétés. La mise en correspondance consiste en la comparaison du cube représentant l'observation avec les cubes associés aux places.

La mise en correspondance initiale est identique à celle de la phase de recalage, à ceci près qu'aucune correspondance temporelle n'est faite. Nous distinguerons donc mise en correspondance d'activités, mise en correspondance temporelle et mise en correspondance généralisée.

### 5.3 - Mise en correspondance d'activités

La mise en correspondance d'activités confronte l'observation aux cubes des activités prédites.

On notera  $\mathcal{A}$  l'ensemble des mises en correspondance d'activités. La définition de chacun des éléments de  $\mathcal{A}$ , tirée de Tessier (2003), est :

**Définition 1** (Mise en correspondance d'activités). Soit  $A$  le cube des propriétés de l'activité et  $O$  celui de l'observation. On dit que l'activité est en correspondance :

$$PARFAITE \Leftrightarrow A = O$$

$$PARTIELLE\ 1 \Leftrightarrow (A \subset O) \vee (O \subset A)$$

$$PARTIELLE\ 2 \Leftrightarrow (A \not\subset O) \wedge (O \not\subset A) \wedge (A \cap O \neq \emptyset)$$

$$IMPOSSIBLE \Leftrightarrow A \cap O = \emptyset.$$

**Exemple** (Mise en correspondance d'activités).

Prenons comme cube d'observation

$$O = \{\text{waypoint}(3, \text{reached}), \text{vitesse}(200)\}$$

que l'on doit confronter à deux activités prédites de cubes respectifs

$$A1 = \{\text{waypoint}(3, \text{reached})\}$$

$$A2 = \{\text{waypoint}(3, \text{reached}), \text{vitesse}(150)\}.$$

La mise en correspondance de  $O$  et  $A1$  est

$$PARTIELLE\ 1 : A1 \subset O.$$

En revanche, la mise en correspondance de  $O$  et  $A2$

$$\text{est } PARTIELLE\ 2 \quad (A2 \not\subset O, \quad O \not\subset A2 \quad \text{et} \\ A2 \cap O = \{\text{waypoint}(3, \text{reached})\}).$$

**Définition 2** (Relation de préférence).

La relation de préférence entre ces différents éléments est :

$$PARFAITE \leq PARTIELLE1 \leq PARTIELLE2 \leq IMPOSSIBLE$$

où  $A \leq B \Leftrightarrow A$  est préférée à  $B$ .

### 5.4 - Mise en correspondance temporelle

L'ajout d'un intervalle de prédiction aux places fait intervenir un nouveau critère, celui de la mise en correspondance temporelle.

L'ensemble des mises en correspondance temporelles est  $\mathcal{T}$  dont les éléments sont décrits par :

**Définition 3** (Mise en correspondance temporelle). Soit  $t$  l'instant de l'observation et  $I$  l'intervalle de prédiction de l'activité. On dit que l'activité est en correspondance :

$$IN \Leftrightarrow t \in I$$

$$POST \Leftrightarrow t > \text{sup}(I)$$

$$ANTE \Leftrightarrow t < \text{inf}(I)$$

Contrairement au cas de la mise en correspondance d'activités, on ne peut ici classer les différents cas selon un ordre total. On choisira un ordre partiel dans lequel  $IN \leq ANTE$  et  $IN \leq POST$ .

### 5.5 - Mise en correspondance généralisée

**Définition 4** (Mise en correspondance généralisée). La réunion des deux mises en correspondance ci-

dessus construit une relation d'ordre partiel. La mise en correspondance d'une observation et d'une activité est affectée de deux attributs (une paire  $(a, t) \in \mathcal{A} \times \mathcal{T}$ ).

On obtient ainsi  $12$  ( $\#\mathcal{A} \times \#\mathcal{T}$ ) mises en correspondance possibles que l'on peut ordonner partiellement d'après les relations d'ordre précédentes.

Les cas particuliers et délicats sont ceux obtenus lorsqu'il faut classer, par exemple, une mise en correspondance (PARFAITE, ANTE) et (PARTIELLE 2, IN). En d'autres termes, est-il préférable de croire à une erreur de manipulation (mauvaise indication de la fréquence de la prochaine balise  $\rightarrow$  PARTIELLE 2) ou à une erreur de prédiction (le pilote a volé plus vite que prévu  $\rightarrow$  ANTE) ?

Dans le contexte de ce travail, on privilégiera la mise en correspondance d'activités sur la mise en correspondance temporelle. On définit donc une relation de préférence partielle dont la règle est :

**Définition 5** (Relation de préférence).

Soit  $m=(a, t)$  et  $m'=(a', t')$  deux mises en correspondance. Les cas possibles sont :

$$a \leq a' \Rightarrow m \leq m'$$

$$(a = a') \wedge (t \leq t') \Rightarrow m \leq m'$$

$$(a = a') \wedge (t = t') \Rightarrow m = m'$$

$$(a = a') \wedge (t \text{ et } t' \text{ incomparables}) \Rightarrow m \text{ et } m' \text{ incomparables.}$$

### 5.6 - Principe de recalage

Lors de l'arrivée d'une observation, celle-ci est mise en correspondance avec les activités prédites. Les mises en correspondance IMPOSSIBLE sont écartées, tandis que les activités restantes (dites *recalées*) sont classées selon l'ordre de préférence exposé dans la définition. Ce nouvel ensemble de plans reconnus est ensuite livré à la phase de prédiction en attendant l'arrivée d'une nouvelle observation.

## 6 - RÉSULTATS

Afin de valider le modèle et les principes définis, nous avons mis au point un logiciel (figure 15) qui met en œuvre les différents algorithmes et qui permet de visualiser l'ensemble des plans manipulés lors des différentes phases (alternativement prédiction et recalage).

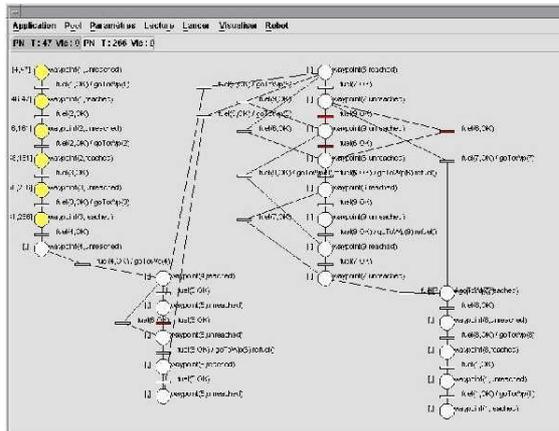


FIG. 15: *Logiciel de suivi d'activité*

À partir des paramètres de vol provenant d'expérimentations menées sur la mission *Refuel* (Dehais, 2003), on a pu suivre et prédire l'activité de plusieurs pilotes expérimentés confrontés à des conflits, et déterminer *a posteriori* les dates de ces conflits et les moments pertinents pour l'envoi de contre-mesures.

## 7 - CONCLUSIONS ET PERSPECTIVES

Dans l'état actuel, le modèle défini et les principes exposés de Kalmansymbo-aéro permettent le suivi et la prédiction d'activité sur le cas de la mission *Refuel*. La modélisation de différentes procédures (par exemple approche sur l'aéroport de Toulouse-Blagnac) permettra d'affiner ces principes. La poursuite de ces travaux fait l'objet d'une thèse au cours de laquelle les principes de prédiction seront approfondis, un lien entre filtres de Kalman numérique et symbolique sera étudié, et la connexion sera établie avec la détection de conflits et l'envoi des contre-mesures.

## 8 - BIBLIOGRAPHIE

B.E.A. (1996), *Statistiques générales 1994-1996*, <http://www.bea-fr.org>

Brémond, F. (1997), *Environnement de résolution de problèmes pour l'interprétation de séquences d'images*, Thèse, Sophia-Antipolis, France.

Callantine, T.J. (2002), *Activity tracking for pilot error detection from flight data*, NASA, USA.

Chaudron, L., Maille, N., Boyer, M. (2003), «The CUBE lattice model and its applications», *Applied Artificial Intelligence*, vol 17, n° 3, pp 201-242.

Cohen, H. (1993), *Neuropsychologie expérimentale et clinique*, Gaetan Morin.

Cossart-Jaupitre, C. (1999), *Un estimateur symbolique pour le suivi de situation*, Thèse, Onera-Supaéro, France.

Dehais, F. (2002), «Modelling cognitive conflict in pilot's activity», in *STAIRS*, Lyon, France.

Dehais, F., Tessier, C., Chaudron, L. (2003), «GHOST : experimenting conflicts countermeasures in the pilot's activity», in *IJCAI*, Acapulco, Mexico.

Dousson, C. (1994), *Suivi d'évolutions et Reconnaissance de chroniques*, Thèse, LAAS-CNRS, France.

Ghallab, M. (2002), *Planification et Raisonnement Temporel*, Notes de cours, LAAS-CNRS.

Largouët, C., Cordier, M.O. (2000), «Timed automata model to improve the classification of a sequence of images», in *ECAI'00*, Berlin, Allemagne.

Rota, N.A., Thonnat, M. (2000), «Activity recognition from video sequence using declarative models», in *ECAI'00*, Berlin, Allemagne.

Tessier, C. (2003), «Towards a commonsense estimator for activity tracking», in *AAAI*, Palo Alto CA, USA.

Valette, R. (2000), *Les Réseaux de Petri*, notes de cours, ENAC.