



HAL
open science

Pilotage stable d'un planeur en utilisant une logique non monotone

José-Luis Vilchis, Pierre Siegel, Andrei Doncescu

► To cite this version:

José-Luis Vilchis, Pierre Siegel, Andrei Doncescu. Pilotage stable d'un planeur en utilisant une logique non monotone. Journées Francophones sur la Planification, la Décision et l'Apprentissage pour la conduite de systèmes, Jul 2017, Caen, France. 4p. hal-01566174

HAL Id: hal-01566174

<https://laas.hal.science/hal-01566174>

Submitted on 20 Jul 2017

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.

Pilotage stable d'un planeur en utilisant une logique non monotone

José-Luis Vilchis Medina¹

Pierre Siegel¹

Andrei Doncescu²

^{1 2} Aix Marseille Université, CNRS, LIF, Marseille, France

² LAAS, CNRS, Toulouse, France

jose Luis.vilchismedina@lif.univ-mrs.fr

Résumé

Le pilotage d'un avion à différentes étapes de vol est un problème de logique non monotone. Parce que les règles de pilotage peuvent changer en fonction des circonstances externes, telles que les perturbations atmosphériques ou les situations de sécurité ou d'urgence du pilote. Nous présentons une méthodologie pour mener un vol stable prenant en compte des facteurs d'incertitude et de contradiction des informations.

Mots Clef

Logique non monotone, Logique des défauts, Représentation des connaissances, Planification, Moteur-planeur, Règles de pilotage.

Abstract

Piloting an aircraft at different flight stages is a non-monotonic logic problem. Because rules of piloting may change depending on external circumstances, such as atmospheric disturbances or the safety or emergency situations of the pilot. We present a methodology for a stable flight taking into account factors of uncertainty and contradiction of information.

Keywords

Non monotonic logic, Default logic, Knowledge representation, Planning, Motor-Glider, Rules of piloting.

1 Introduction

La stabilisation des aéronefs dans les différentes étapes du vol, est une partie importante pour une trajectoire bien définie. Il y a 4 forces fondamentales qui font l'avion rester dans l'air : La force de portance, de traînée, le poids et la traction[2]. Une fois que l'aéronef est dans l'air, il doit s'orienter et se déplacer dans l'espace pour obtenir la trajectoire désirée. Généralement, les angles roll, pitch et yaw sont utilisées pour l'orientation. Les avions ont besoin d'utiliser des moteurs comme agents propulseurs, cela permet d'avoir la vitesse nécessaire pour décoller, mais aussi rester dans l'air[2].

Nous utilisons un planeur équipé d'un moteur. L'avantage d'utiliser ce type d'aéronef, c'est parce, il est l'un des

meilleurs aéronefs en termes de sa finesse(rapport entre la portance et la traînée). La finesse est liée aussi au rapport entre la distance horizontale et verticale parcourue. Par exemple, s'il s'est déplacé 40 km avec une perte d'altitude de 1 km, il a une finesse de 40. Dans les règles de pilotage, il y a des cas contradictoires ou incertains. Par exemple, si un pilote a une émergence(soit un problème technique, soit pour des raisons de sécurité, etc.), il doit violer certaines règles pour résoudre la problématique. Dans cette article nous allons présenter une méthode basée en logique non monotone pour la conduite du vol stable, que prendre en compte l'incertitude et l'imprécision de l'information[1][7].

Nous présentons dans la section suivante la représentation des connaissances et la logique classique. La définition et caractéristiques de la logique non monotone sont comprises dans la section 3. La représentation du système en logique des défauts est dans la section 4. Dans la section 5, nous décrivons une simulation d'une situation de vol avec toutes les solutions possibles de choix et finalement les conclusions sont décrites dans la section 6.

2 Représentation des connaissances

La représentation des connaissances est une partie importante en intelligence artificielle car elle donne une description de l'environnement pour qu'un ordinateur soit capable d'inférence. Dans ce contexte, grâce au capteur (Figure 1), nous disposons de la lecture de différentes variables physiques(vitesses et accélérations), avec laquelle nous pouvons décrire ou représenter les conditions du planeur, pour après inférer les actions pour attendre quelque soit l'objectif (décollage, atterrissage, tourner, ...).

2.1 En logique classique

Avec la logique classique nous pouvons décrire situations qui passent dans la vie réelle ou représenter le monde, l'environnement. Mais dans la vie réelle, il y a des situations ou règles contradictoires. Par exemple, les ovipares pondent des œufs. En logique, nous pouvons écrire la règle suivante : $layEggs(X) \rightarrow ovipare(X)$. Maintenant, si nous considérons $X = Ornithorynque$, par définition, l'ornithorynque est un mammifère qui pond œufs. En conséquence,

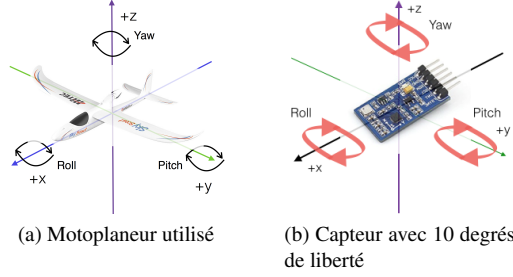


FIGURE 1 – Axes de référence du planeur(a) et du capteur(b), les dessins n'ont pas à l'échelle.

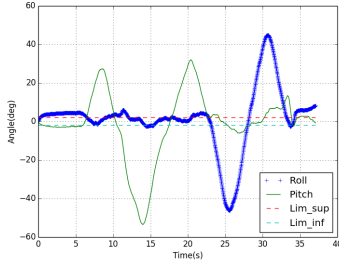


FIGURE 2 – Représentation dynamique du vol.

cette règle ne peut pas être utilisée. Donc la logique classique est limitée, car elle ne prend en compte certains aspects d'imprécisions ou d'exceptions. Pour traiter ce type de problématique, nous présentons la logique non monotone, en particulière, la logique des défauts.

3 Logique non-monotone

La logique non monotone est un type de logique qui permet de représenter règles d'incertitude, d'imprécisions ou d'exceptions. Les premières auteurs dans le domaine de la logique non monotone ont été à la fin des années 1970, J. Mc-Carthy[5], D. Mc-Dermott et J. Doyle, et R. Reiter[6].

3.1 Logique des défauts

Une théorie de défaut T consiste en un ensemble de faits W , qui sont des formules en logique de premier ordre et un ensemble de défauts D , qui sont des règles d'inférence [6]. L'outil de représentation principale est la règle de défaut. Un défaut est une règle d'inférence sous la forme : $\frac{A(X) : B(X)}{C(X)}$, où $A(X), B(X), C(X)$ sont des formules bien formées(logique de premier ordre). Où $X = (x_1, x_2, x_3, \dots, x_n)$ est un vecteur de variables libres (non quantifiées). $A(X)$ sont les prérequis, $B(X)$ sont les justifications et $C(X)$ sont les conséquences. Il est possible d'utiliser la notation suivante, $A(X) : B(X) \rightarrow C(X)$. Intuitivement, une règle de défaut signifie : "si $A(X)$ est vrai, et il n'y a aucune preuve que $B(X)$ soit faux, alors $C(X)$ est vrai". Il est défini comme un défaut normal, si $B(X) = C(X)$. Les règles des défaut sont utilisées pour créer des extensions.

3.2 Définition d'extension

Une extension de la théorie de défaut $\Delta = (D, W)$, est un ensemble E de formules logiques [6]. Une extension doit vérifier la propriété suivante : Si d est un défaut de D , dont le prérequis est dans E , sans que la négation de sa justification n'est pas dans E , alors la conséquence de d est dans E . Formalement, E est une extension de Δ si et seulement si :

- $E = \cup_{i=0}^{\infty} E_i$ with :
- $E_0 = W$ and for $i \geq 0$
- $E_{i+1} = Th(E_i) \cup \{C(X) \mid \frac{A(X) : B(X)}{C(X)} \in D, A(X) \in E_i \text{ and } \neg B(X) \notin E\}$

Où $Th(E_i)$ est l'ensemble de formules dérivés de E_i . La définition précédente est difficile à appliquer dans la pratique. Parce que $\neg B(X) \notin E$ suppose E est connu, mais E n'est pas encore calculé. Nous allons prendre la définition de défauts normaux, c'est-à-dire, $B(X) = C(X)$. Une extension est définie : E est une extension de Δ si et seulement si :

- $E = \cup_{i=0}^{\infty} E_i$ with :
- $E_0 = W$ and for $i \geq 0$
- $E_{i+1} = Th(E_i) \cup \{C(X) \mid \frac{A(X) : C(X)}{C(X)} \in D, A(X) \in E_i \text{ and } \neg C(X) \notin E_i\}$

Où $Th(E_i)$ est l'ensemble de formules dérivés de E_i . Selon Reiter [6], si tous les défauts sont normaux, il existe au moins une extension. Les extensions sont définies comme points fixes.

4 Représentation en logique des défauts

La base de connaissances du système est décrite par un ensemble des faits et des défauts.

4.1 L'ensemble des faits W

L'ensemble W sont faits vrais. Par exemple, $glider(decrease, t)$, signifie que le planeur diminue en altitude au temps t . Dans cet ensemble, nous pouvons aussi décrire des exclusions mutuelles, ce sont des situations qui ne peuvent pas se produire en même temps. Par exemple pour les actions, le planeur ne peut pas tourner à gauche et à droite en même temps t . Formellement, nous avons : $\forall t, \neg(yoke(pull, t+1) \wedge yoke(push, t+1))$.

4.2 L'ensemble des défauts D

L'ensemble des règles d'inférence D décrit des actions possibles, des informations incomplètes et contradictoires. Chaque règle est représenté sous la forme de défaut normal. Par exemple, $\frac{glider(decrease, t) : yoke(pull, t+1)}{yoke(pull, t+1)}$. Intuitivement, cela signifie, "Si $glider(decrease, t)$ est vrai, et si $yoke(pull, t+1)$ n'a pas de contradictions, alors $yoke(pull, t+1)$ est vrai".

4.3 Exemple de calcul

Voici, nous avons une fonction(F) qui représente faits vrais. Cette représentation décrit la situation du planeur.

$F : glider(decrease, t), glider(turn_left, t), glider(motor_off, t), glider(low_altitude, t), glider(low_airspeed, t)$. Nous pouvons dire en français que : le planeur décroît, il tourne à gauche, il n'a pas activé le moteur, il a une altitude basse et une vitesse basse au moment t . Que devons-nous faire pour avoir un vol stable? Avec cette information, nous pouvons calculer les extensions et enduire les actions possibles pour attendre l'objectif : vol stable. Ensuite, nous montrons les extensions avec les différents défauts obtenus :
 $E_0 = \{ glider(decrease, t), glider(turn_left, t), glider(motor_off, t), glider(low_altitude, t), glider(low_airspeed, t) \}$

$$d_6 = \frac{glider(decrease, t) : yoke(pull, t + 1)}{yoke(pull, t + 1)} \quad (1)$$

$$d_8 = \frac{glider(decrease, t) : motor(on, t + 1)}{motor(on, t + 1)} \quad (2)$$

$$d_{14} = \frac{glider(turn_left, t) : yoke(right, t + 1)}{yoke(right, t + 1)} \quad (3)$$

$E_1 = \{ glider(decrease, t), glider(turn_left, t), glider(motor_off, t), glider(low_altitude, t), glider(low_airspeed, t) \}$

$$d_6 = \frac{glider(decrease, t) : yoke(pull, t + 1)}{yoke(pull, t + 1)} \quad (4)$$

$$d_9 = \frac{glider(decrease, t) : motor(off, t + 1)}{motor(off, t + 1)} \quad (5)$$

$$d_{14} = \frac{glider(turn_left, t) : yoke(right, t + 1)}{yoke(right, t + 1)} \quad (6)$$

$E_2 = \{ glider(decrease, t), glider(turn_left, t), glider(motor_off, t), glider(low_altitude, t), glider(low_airspeed, t) \}$

$$d_7 = \frac{glider(decrease, t) : yoke(push, t + 1)}{yoke(push, t + 1)} \quad (7)$$

$$d_9 = \frac{glider(decrease, t) : motor(off, t + 1)}{motor(off, t + 1)} \quad (8)$$

$$d_{14} = \frac{glider(turn_left, t) : yoke(right, t + 1)}{yoke(right, t + 1)} \quad (9)$$

Nous pouvons constater que dans chaque extension calculée, nous avons des sous-systèmes cohérents. Maintenant, nous avons résolu le problème de contradiction, présenté au début. Nous pouvons ajouter des règles plus complexes pour une base de connaissances plus complète.

5 Simulation : décollage stable

Les principaux éléments qui conduisent à l'objectif souhaité sont :

- Etat : C'est la représentation de la réalité ou situation de l'objet à chaque instant du temps.
- Temps : C'est l'instant entre les états.
- Action : C'est l'activité à réaliser afin d'atteindre l'objectif.

Ces éléments sont clés pour inférer les actions suivantes, au temps $t + 1$. Ensuite, nous allons simuler une situation pour attendre un décollage comme objectif. Dans le calcul sont considérés les trois aspects précédents[3][4].

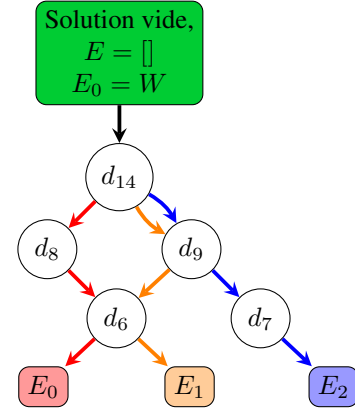


FIGURE 3 – Représentation graphique pour le calcul des extensions et des défauts.

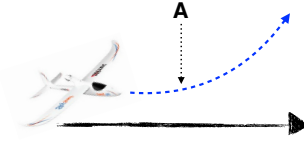


FIGURE 4 – Situation A.

Dans cette situation (Fig. 4) nous avons les informations suivantes, $G : glider(pitch_stable, t), glider(roll_stable, t), glider(motor_off, t), glider(low_altitude, t), glider(low_airspeed, t)$. Si nous appliquons la théorie des défauts, nous obtenons 5 différentes solutions.

$E_0 = \{ glider(pitch_stable, t), glider(roll_stable, t), glider(motor_off, t), glider(low_altitude, t), glider(low_airspeed, t) \}$

$$d_{16} = \frac{glider(roll_stable, t) : yoke_roll(neutral, t + 1)}{yoke_roll(neutral, t + 1)} \quad (10)$$

$$d_{17} = \frac{glider(pitch_stable, t) : yoke_pitch(neutral, t + 1)}{yoke_pitch(neutral, t + 1)} \quad (11)$$

$$d_{20} = \frac{glider(low_altitude, t) : motor(on, t + 1)}{motor(on, t + 1)} \quad (12)$$

$E_1 = \{ glider(pitch_stable, t), glider(roll_stable, t), glider(motor_off, t), glider(low_altitude, t), glider(low_airspeed, t) \}$

$$d_{16} = \frac{glider(roll_stable, t) : yoke_roll(neutral, t + 1)}{yoke_roll(neutral, t + 1)} \quad (13)$$

$$d_{17} = \frac{glider(pitch_stable, t) : yoke_pitch(neutral, t + 1)}{yoke_pitch(neutral, t + 1)} \quad (14)$$

$$d_{21} = \frac{glider(low_altitude, t) : motor(off, t + 1)}{motor(off, t + 1)} \quad (15)$$

$E_2 = \{ glider(pitch_stable, t), glider(roll_stable, t), glider(motor_off, t), glider(low_altitude, t), glider(low_airspeed, t) \}$

$$d_{16} = \frac{glider(roll_stable, t) : yoke_roll(neutral, t + 1)}{yoke_roll(neutral, t + 1)} \quad (16)$$

$$d_{19} = \frac{glider(low_altitude, t) : yoke(push, t + 1)}{yoke(push, t + 1)} \quad (17)$$

$$d_{21} = \frac{glider(low_altitude, t) : motor(off, t + 1)}{motor(off, t + 1)} \quad (18)$$

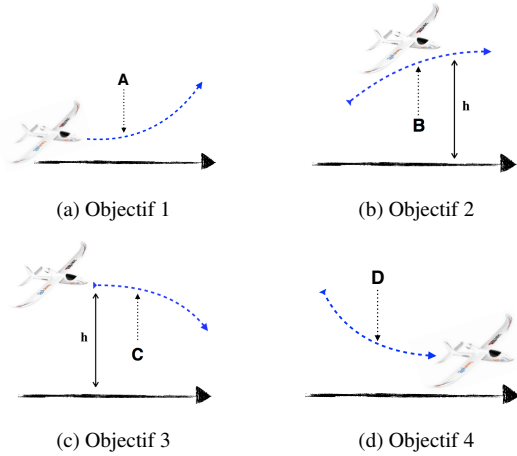


FIGURE 5 – Différents objectifs pendant le vol.

$E_3 = \{glider(pitch_stable, t), glider(roll_stable, t), glider(motor_off, t), glider(low_altitude, t), glider(low_airspeed, t)\}$

$$d_{17} = \frac{glider(pitch_stable, t) : yoke_pitch(neutral, t + 1)}{yoke_pitch(neutral, t + 1)} \quad (19)$$

$$d_{18} = \frac{glider(low_altitude, t) : yoke(pull, t + 1)}{yoke(pull, t + 1)} \quad (20)$$

$$d_{20} = \frac{glider(low_altitude, t) : motor(on, t + 1)}{motor(on, t + 1)} \quad (21)$$

$E_4 = \{glider(pitch_stable, t), glider(roll_stable, t), glider(motor_off, t), glider(low_altitude, t), glider(low_airspeed, t)\}$

$$d_{17} = \frac{glider(pitch_stable, t) : yoke_pitch(neutral, t + 1)}{yoke_pitch(neutral, t + 1)} \quad (22)$$

$$d_{18} = \frac{glider(low_altitude, t) : yoke(pull, t + 1)}{yoke(pull, t + 1)} \quad (23)$$

$$d_{21} = \frac{glider(low_altitude, t) : motor(off, t + 1)}{motor(off, t + 1)} \quad (24)$$

La meilleure solution dans la situation A (Fig. 4) parmi les autres, est l'extension E_3 qui a les combinaisons des actions(défauts) nécessaires pour avoir un décollage au temps $t + 1$. Le choix d'une extension(avec des défauts) s'est effectué sur une analyse multicritère[3], sous la définition suivante : $P(A_K/A_L) = \prod_{j=1}^m (a_{Kj}/a_{Lj})^{w_j}$ for $K, L = 1, 2, 3, \dots, m$.

Sur la Fig. 5, nous montrons les différents objectifs qui composent les étapes de vol, chacun avec multiple décisions à prendre.

Sur la Fig. 6, une représentation graphique des états pour la commande de vol stable. L'intérêt est de rester à l'état zéro où l'aéronef aura une position droite, parallèle au sol.

6 Conclusion

La représentation des connaissances joue un rôle importante en Intelligence Artificielle, avec laquelle nous pouvons représenter l'univers. Mais cependant, il y a des cas contradictoires, où la logique classique, telle que la logique des prédicats ne permet pas de travailler avec ces cas. La

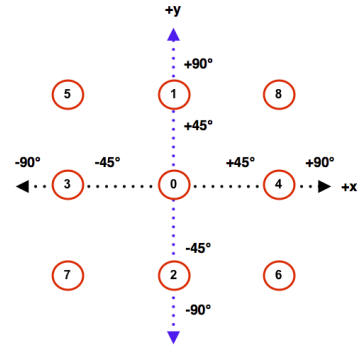


FIGURE 6 – Plan Roll vs Pitch.

logique non monotone prendre en compte divers aspects du raisonnement (dans l'incertain, temporel, etc.), mais aussi, elle peut gérer règles d'incertitude, d'imprécisions ou susceptibles des exceptions. La logique des défauts est un outil important capable de inférer avec information contradictoire et incomplète. Nous sommes en train d'améliorer les aspects de décisions, pour établir une modèle en utilisant le théorème de l'optimum de Pareto.

Références

- [1] A. Doncescu, P. Siegel, DNA Double-Strand Break-Based Nonmonotonic Logic, *Computational Biology, Bioinformatics and Systems Biology*, 2015.
- [2] Direction Général de l'Aviation Civile, *Manuel du pilote d'avion Vol à vue*, Direction Général de l'Aviation Civile, 1992.
- [3] I. Toulgoat, Modélisation du comportement humain dans les simulations de combat naval, *Thèse de doctorat*, 2011.
- [4] I. Toulgoat, P. Siegel, A. Doncescu, Modelling of Submarine Navigation by Nonmonotonic Logic, *BWCCA*, 2011.
- [5] J. McCarthy, Circumscription - A form of non-monotonic reasoning, *Artificial Intelligence*, 1980.
- [6] R. Reiter, A logic for Default Reasoning, *Artificial Intelligence*, pp. 81-132, 1980.
- [7] T. Le, A. Doncescu, P. Siegel, Utilization of Default Logic for Analyzing a Metabolic System in Discrete Time, *ICCSA*, 2013.