

Contrôle de Vol d'un Planeur Basé sur 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

joseluis.vilchismedina@lif.univ-mrs.fr

Résumé

Le pilotage d'un avion consiste à faire divers choix d'actions pour réaliser une trajectoire. Mais les différentes phases de vol ont un environnement changeant, en conséquence on peut avoir des informations incertaines. Les situations peuvent changer en fonction des situations externes, telles que les perturbations atmosphériques, ou situations de sécurité et urgence du pilote. Le problème d'un pilote est alors contrôler l'aéronef dans des situations incertaines. Donc la commande de vol est un problème non-monotone. Nous présentons une méthode pour contrôler un planeur prenant en compte des facteurs d'incertitude et de contradiction.

Mots Clef

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

Abstract

Aircraft piloting consist of making various choices of actions to realize a trajectory. But different phases of flight have a changing environment, therefore, We can have uncertain information. Situations may change depending on external conditions, such as atmospheric disturbances or safety and emergency situations of the pilot. The problem of a pilot is then to control the aircraft in uncertain situations. So flight control is a non-monotonic problem. We present a method for controlling a glider taking into account factors of uncertainty and contradiction.

Keywords

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

1 Introduction

La stabilisation des aéronefs, à différentes étapes du vol, est une partie importante pour une trajectoire bien définie. Il y a quatre forces qui maintiennent l'avion 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 réaliser la trajectoire désirée.

Généralment, les angles «roll», «pitch» et «yaw» sont utilisées pour mesurer l'orientation et l'inclinaison. Les avions ont besoin des moteurs comme agents propulseurs, cela permet d'avoir la vitesse nécessaire pour décoller et rester dans l'air [2]. Dans la plupart des cas, le contrôle de l'aéronef sont modélisés par des équations différentielles et des matrices de rotation [8]. Nous abordons le problème du point de vue de l'intelligence artificielle. Pour aborder cette problématique nous utilisons une théorie de la logique non-monotones. Nous utilisons un motoplaneur (planeur disposant d'un moteur d'appoint). L'avantage d'utiliser ce type d'aéronef, c'est parce que, il est l'un des meilleurs en termes de sa finesse (rapport entre la portance et la traînée). La finesse est liée aussi au rapport entre à la distance au sol et l'altitude perdue. Par exemple, s'il s'est déplacé de 40 km avec une perte d'altitude de 1 km, alors sa finesse est de 40. Dans les règles de pilotage, il y a des cas contradictoires ou incertains, si un pilote a une urgence (soit un problème technique, soit pour des raisons de sécurité, etc.), il doit violer certaines règles pour résoudre la problématique. Dans cet article nous allons présenter une méthode basée sur une logique non monotone pour la conduite d'un vol stable, prenant 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 les 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 et finalement les conclusions 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 qui sera interpréter par un ordinateur. Dans ce contexte, grâce au capteur IMU¹ (Fig. 1), nous disposons de la mesure de différentes variables physiques (vitesses et accélérations, Fig. 2), avec lesquelles nous pouvons représenter les conditions spatio-temporelles du planeur. Ces informations vont permettre d'inférer les

1. Inertial Measurement Unit.

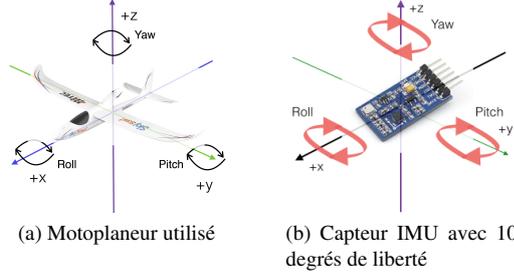


FIGURE 1 – Axes de référence du planeur(a) et du capteur(b), les dessins n'ont pas à l'échelle.

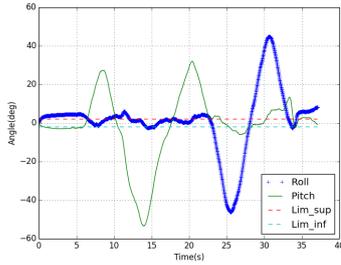


FIGURE 2 – Représentation dynamique du vol.

actions pour atteindre un objectif : décollage, atterrissage, tourner à droite ou à gauche, etc.

2.1 En logique classique

Avec la logique classique nous pouvons décrire des situations qui se passent dans la vie réelle ou encore représenter le monde. Dans la vie réelle, il y a des situations ou règles contradictoires. Si on prend l'exemple suivant, on peut représenter en logique classique le constat : "Les ovipares pondent des œufs", avec la règle : $layEggs(X) \rightarrow ovipare(X)$. Maintenant, si nous considérons $X = Ornithorynque$, par définition, l'ornithorynque est un mammifère, or il pond des œufs. En conséquence, la règle précédente ne peut pas être satisfaite. La logique classique est donc limitée, car elle ne prend en compte ni les aspects d'imprécisions ni ceux d'exceptions. Afin de traiter ce type de problématique, nous proposons d'utiliser la logique non monotone, plus particulièrement, la logique des défauts.

3 Logique Non-monotone

J. Mc-Carthy [5], D. Mc-Dermott et J. Doyle, et R. Reiter [6] ont été les premiers auteurs à travailler avec la logique non monotone dès la fin des années 1970. Une logique monotone ne peut pas gérer les incertitudes (comme vu l'exemple présenté dans la section 2.1). La logique non-monotone est une logique formelle qui ne respecte pas la propriété de la monotonie, définie comme : $A \vdash w, A \cup B \vdash w$. Si on a un modèle A , et on ajoute des informations de B , on ne peut pas réduire les conclusions de w . Ainsi, elle permet de capturer et représenter le raisonnement par défauts (faits incertains et contradictoires). C'est

un problème classique en Intelligence Artificielle. Une des logiques les plus connues et étudiées pour traiter cette problématique est la logique des défauts. C'est cette logique que nous présenterons dans la suite de ces travaux.

3.1 Logique des défauts

Une théorie des défauts 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 principal 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 (en logique de premier ordre). Où $X = (x_1, x_2, x_3, \dots, x_n)$ est un vecteur de variables libres (non quantifiées). $A(X)$ est le prérequis, $B(X)$ est la justification et $C(X)$ est la conséquence. 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 défauts sont utilisées pour calculer des extensions.

3.2 Définition d'extension

Une extension de la théorie des défauts $\Delta = (D, W)$, est un ensemble E des 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 ne soit dans E , alors la conséquence de d est dans E . Formellement, E est une extension de Δ si et seulement si :

$$\begin{aligned} & - E = \cup_{i=0}^{\infty} E_i \text{ avec :} \\ & - E_0 = W \text{ et pour } i \geq 0 \\ & E_{i+1} = Th(E_i) \cup \{C(X) \mid \frac{A(X) : B(X)}{C(X)} \in D, \\ & A(X) \in E_i \text{ et } \neg B(X) \notin E_i\} \end{aligned}$$

Où $Th(E_i)$ est l'ensemble de formules dérivées 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 des défauts normaux, c'est-à-dire, $B(X) = C(X)$. Une extension est définie : E est une extension de Δ si et seulement si :

$$\begin{aligned} & - E = \cup_{i=0}^{\infty} E_i \text{ avec :} \\ & - E_0 = W \text{ et pour } i \geq 0 \\ & E_{i+1} = Th(E_i) \cup \{C(X) \mid \frac{A(X) : C(X)}{C(X)} \in D, \\ & A(X) \in E_i \text{ et } \neg C(X) \notin E_i\} \end{aligned}$$

Où $Th(E_i)$ est l'ensemble de formules dérivées 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 ou solutions.

4 Représentation en Logique des Défauts

Dans cette section nous présentons les ensembles de faits W et de défauts D . Nous donnons également un exemple de calcul des extensions.

4.1 L'ensemble des faits W

W est un ensemble de faits vrais. Par exemple, $glider(decrease, t)$, signifie que le planeur a perdu de l'altitude au temps t par rapport au temps $t - 1$. 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 simultanément au même temps t . Formellement cette exclusion mutuelle est représentée de la façon suivante : $\forall t, \neg(yoke(pull, t + 1) \wedge yoke(push, t + 1))$.

4.2 L'ensemble des défauts D

L'ensemble des règles d'inférence D décrit des actions possibles en prenant en compte les informations incomplètes et contradictoires de l'environnement. Chaque règle est représenté sous la forme de défaut normal (Section 3.1). Par exemple, $\frac{glider(decrease, t) : yoke(pull, t + 1)}{yoke(pull, t + 1)}$. Intuitivement, cela signifie, "Si le $glider(decrease, t)$ est vrai, et si c'est possible que $yoke(pull, t + 1)$, alors $yoke(pull, t + 1)$ ". Autrement dit, si c'est possible de faire une action, on la fera.

4.3 Exemple de calcul

Voici, nous avons une fonction F qui représente les faits vrais. Cette représentation d'écrit la situation du planeur au temps t (pour la fonction $glider(X, t)$, elle sera notée comme $g(X, t)$).

$$F : g(decrease, t), g(turn_left, t), g(motor_off, t), \\ g(low_altitude, t), g(low_airspeed, t).$$

Nous pouvons dire en français que : le planeur perd de l'altitude, il tourne à gauche, il a le moteur coupé et une vitesse basse au moment t . Nous nous posons la question suivante : Quelles actions devons-nous faire pour avoir un vol stable ? Avec cette information, nous pouvons calculer les extensions (ensemble des actions) possibles pour atteindre l'objectif : vol stable. Ensuite, nous montrons les extensions avec les différents défauts obtenus (Fig. 3) :

$$E_0 = \{g(decrease, t), g(turn_left, t), g(motor_off, t), \\ g(low_altitude, t), g(low_airspeed, t)\}$$

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

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

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

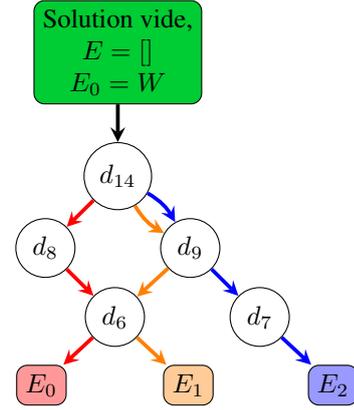


FIGURE 3 – Représentation graphique de calcul des extensions.

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

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

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

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

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

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

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

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

Nous pouvons constater que dans chaque extension calculée, nous avons des sous-systèmes cohérents [3][4]. Maintenant, nous avons résolu le problème de contradiction, présenté au début (Section 1). Nous pouvons ajouter des règles plus complexes pour une base de connaissances plus complète.

5 Simulation : Décollage

Dans cette section nous simulons une situation de décollage et nous calculons toutes les solutions possibles pour cette phase de pilotage. Les principaux éléments qui conduisent à l'objectif souhaité sont :

- État : C'est la représentation de la situation de l'objet (planeur) 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.

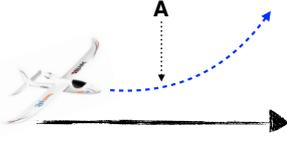


FIGURE 4 – Situation A.

Ces éléments sont clés pour inférer les actions suivantes. Dans le calcul sont considérés les trois aspects précédents. Dans cette situation (Fig. 4) nous avons les informations suivantes (pour la fonction $glider(X, t)$ vue précédemment, elle sera notée comme $g(X, t)$) :

$$G : g(pitch_stable, t), g(roll_stable, t), g(motor_off, t), \\ g(low_altitude, t), g(low_airspeed, t).$$

Si nous appliquons la théorie des défauts, plus particulièrement la logique des défauts (Section 3.1), nous obtenons 5 différentes extensions ou solutions.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

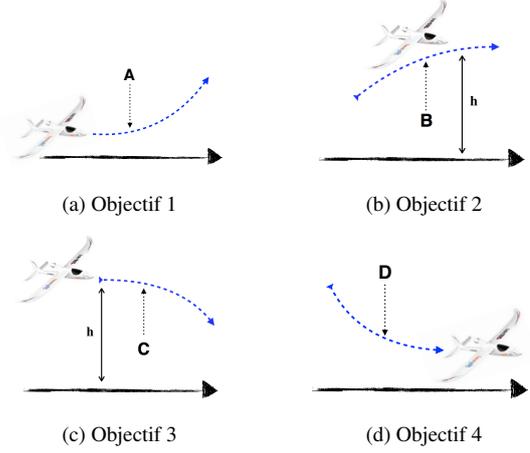


FIGURE 5 – Différents objectifs pendant le vol.

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

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

La meilleure extension et/ou solution de la situation A simulée précédemment (Fig. 4) est l'extension E_3 qui a les combinaisons des actions $\{yoke(pull, t + 1), yoke_pitch(neutral, t + 1), motor(on, t + 1)\}$ nécessaires pour atteindre l'objectif : **décollage**. Puisque l'extension E_0 a : $\{yoke_roll(neutral, t + 1), yoke_pitch(neutral, t + 1), motor(on, t + 1)\}$, le résultat est un vol droit que ce n'est pas l'objectif. L'extension E_1 a : $\{yoke_roll(neutral, t + 1), yoke_pitch(neutral, t + 1), motor(off, t + 1)\}$, il n'y a pas de mouvement sur le planeur car le moteur est coupé. L'extension E_2 a : $\{yoke_roll(neutral, t + 1), yoke(push, t + 1), motor(off, t + 1)\}$, même résultat que l'extension précédente, pas de mouvement car le moteur est coupé. Et finalement, l'extension E_4 a : $\{yoke(pull, t + 1), yoke_pitch(neutral, t + 1), motor(off, t + 1)\}$, même résultat que l'extension précédente, pas de mouvement car le moteur est coupé. Le choix d'une extension (ensemble des actions) est basé sur une analyse multicritère [3]. Nous avons choisi le modèle de produits pondérés car il permet de faire une analyse adimensionnelle sur les alternatives. Ensuite, nous montrons la définition :

$$P(A_K/A_L) = \prod_{j=1}^m (a_{Kj}/a_{Lj})^{w_j} \text{ pour } \\ K, L = 1, 2, 3, \dots, m.$$

Nous montrons sur la Fig. 5 les différents objectifs qui sont les différents étapes de pilotage.

6 Conclusion

La logique non-monotone est un outil important en Intelligence Artificielle car elle permet de capturer le raisonnement incertain. Nous avons fait une simulation d'un cas contradictoire (décollage) et nous avons constaté que la logique des défauts peut gérer les situations d'incertitudes et

des contradictions, de cette façon, nous avons obtenu 5 extensions ou solutions qui sont sous-systèmes cohérentes. Nous sommes en train d'améliorer les aspects des décisions, pour cela nous étudions l'optimum de Pareto. Le théorème de Pareto n'est pas sensible aux déséquilibres dans la répartition des ressources. Ce point est important pour résoudre le problème de gestion de l'énergie du système.

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.
- [8] K. Terrell, S. Zein-Sabatto, Intelligent reconfigurable control system for aircraft flight control, *Southeast-Con, 2017, IEEE*, 2017.