Non-monotonie et Resilience: Application au Pilotage d'un Moto-planeur Autonome

José Luis Vilchis Medina[1] **Pierre Siegel**[1] **Andrei Doncescu**[2]
[1]Aix-Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France
[2] LAAS, CNRS, France
{joseluis.vilchismedina,pierre.siegel}@lis-lab.fr, andrei.doncescu@laas.fr

Résumé

Dans cet article, nous présentons un système résilient pour un moto-planeur, basé sur une logique non-monotone. Les systèmes écologiques, biologiques et autres sont résilients, par exemple, les catastrophes naturelles, les bancs de poissons et les nuées d'oiseaux. C'est une propriété qui permets d'absorber les perturbations et de surmonter les adversités. Le pilotage est également un système résilient car il peut avoir des situations conflictuelles et l'environnement est imprévisible. Le pilote doit alors changer de comportement. Quand un pilote fait face à ce genre de situations, il entre dans un raisonnement incertain, malgré le fait qu'il doive prendre des décisions pour différents objectifs. Nous introduisons un modèle résilient non-monotone pour piloter un moto-planeur autonome. Ce modèle n'inclut pas la notion de temps. La logique des défauts a été utilisée pour trouver des points fixes à partir d'informations ambiguës et conflictuelles. Le modèle proposé ici contient une connaissance du monde avec un ensemble de situations, d'objectifs et d'actions. Après le calcul des solutions plausibles, la prise de décision est basée sur une théorie non-probabiliste. Nous avons défini une notion de stabilité dans des situations de pilotage incertaines en utilisant la propriété de résilience.

Abstract

This article presents a resilient system for a motor-glider based on non-monotonic logic. Resilience is the property of a system allowing to absorb disturbances and overcoming adversities. Ecological, biological and many other systems are resilient, for instance, natural disasters, fish school and birds flock. Piloting is also a resilient system because it could have conflicting situations and environment is unpredictable, so behavior change. When a pilot faces such kind of situations, he enters into uncertain reasoning, despite the fact that he must take decisions for different objectives. We introduce a non-monotonic resilient model to pilot an autonomous motor-glider. This model does not include the notion of time to make decisions. Default logic is used to find fixed points from ambiguous and conflicting information. The resilient model proposed here contains a world knowledge with a set of situations, objectives and actions. After computation of plausible solutions, decision-making is based on a non-probabilistic theory. We define a notion of stability in uncertain situations of flight using the property of resilience.

1 Introduction

After many decades of research in the field of aeronautics today new directions open to use plane or any aerial vehicle for different applications in an autonomous way. Definition of autonomous aerial vehicle (UAV) is a programmed vehicle which receives directions from a source placed at a distance apart from it. It can be flown without a pilot by using a particular system on the ground. Using UAV is not only confined to the military fields. They are used in many areas such as agriculture, construction, entertainment, and so far. In our study, we are using a motor-glider. Motor-glider has many constraints when is flying, besides, the pilot has short time to make decisions. He considers certain information and thus be able to make actions, for instance, increase or decrease the engine power, turn the steering wheel to the right, pull, etc. A motor-glider is equipped with an engine motor, which allows to take-off and climb without assistance, in contrast with a normal glider that is non-motorized. In the 70's, Holling introduced the term of resilience to model the dynamics of natural disasters [5]. In other fields of science the concept of resilience is defined as the property of a system to absorb and anticipate perturbations [4]. In ecology, resilience aids to understand natural disasters behavior [5, 1]. In engineering, resilience ensures consistency, robustness and stability [5, 14], even in uncertain environments [16]. Piloting use non-monotonic reasoning when environment change, it should take decisions because perturbations appear. In this paper, we present a model based on non-monotonic logic and the property of resilience, both we will allow us to tackle the problem of uncertain reasoning with incomplete information and stability of an autonomous motor-glider. The sections are composed in the next order. First, the traffic pattern circuit for airplanes and states of flight are explained. Non-monotonic reasoning and default logic are presented in section 3. The complete model as well as situations, objectives and actions are described in section 4. The properties of resilience and stability are explained in section 5. The implementation of the model is described in section 6. Finally, conclusion is described in section 7.

2 Traffic Pattern Circuit

Every pilot knows the traffic pattern circuit. It is one of the basic maneuvers to take-off and land. However, it contains the necessary rules to carry out a long flight. Next, we explain the different states of flight for an airplane. Since most of these states are the same for motor-gliders. First of all, the pilot needs to know airplane states, so he uses the cockpit. The cockpit is a set of instruments on board that displays parameters such as ***airspeed*** (Miles/h), ***artificial horizon*** (pitch and roll) [1], ***variometer*** (Feet/s), ***altitude*** (Feet), ***compass***,...

Traffic pattern circuit, Fig. 1, has different flight stages. It starts at the point S_p where the airplane is in *Rest*. When the pilot is ready and he has the authorization, he increases all the engine power to get a right airspeed to take-off (point a). This is airplane should climb to a suitable height (point b). After that, the pilot should turn the yoke to the left making an orthogonal path to the runway (point c). At this point, he turns again the yoke to the right having constant airspeed, constant altitude and zero vertical speed. When he arrives to the point d he will prepare to land. Turning the yoke to the right, decreasing in altitude and having negative vertical speed, until arrive to the point e. Once again, he should turn the yoke to the right to continue decreasing in altitude and having a stable roll and negative pitch, until point f. After this point, airplane touches the ground. Final point F_p is where airplane state is again in *Rest*. In order to formalize the circuit, we represent knowledge using First-Order Logic (FOL). This is a formal language, which allows to represent almost everything in natural sense, it is expressiveness. We could say : "An airplane is landing". Using FOL, we have : $land(airplane)$. Another instance could be : "Pilot increase the engine power". In FOL we have : $engine(pilot, increase)$ and so on. The flight manual contains all necessary information to pilot an airplane, including technical descriptions, physical limitations, rules and emergency procedures. But all these information are generals that depending on the situation could have contradictory rules. For instance, there is a rule that says the minimum over flight height will never be less than 500 feet, in fact this altitude depends of the agglomeration. This rule could be expressed in FOL, considering that x = airplane, as follows :

1. ***Pitch*** is the angle formed by the airplane when has rotated around "y-axis". Similarly, ***roll*** is formed by the airplane when has rotated, but around "x-axis".

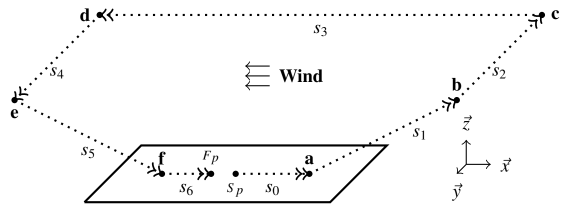

FIGURE 1 – Traffic Pattern.

$$altitude(x) \rightarrow (x > 500) \quad (1)$$

But when an airplane lands its altitude is less than 500 feet. This could be expressed as follows :

$$land(x) \rightarrow (x < 500) \quad (2)$$

General rules are described in the flight manual but pilot is the one who finally decides if he violates such rules. We can see that equation (1) and (2) are contradictory. This is a limitation of classical logic, because it is monotonous. Formally the property of monotony is : $A \vdash w$ then $A \cup B \vdash w$. In other words, adding new information to a model, the consequences are not reduced. This kind of problem, about contradictions and exceptions is well known in Artificial Intelligence. It has been studied from along time [10, 9]. We can see that it is a non-monotonic problem. In order to tackle it we have to move from this framework of classical logic. Because pilot use non-monotonic reasoning when he has new information and he can break the rules.

3 Non-monotonic Reasoning and Default Logic

Non-monotonic reasoning is a class of reasoning where we make assumptions about things jumping to the conclusions. Humans use kind of reasoning, this is the way we can do in situations with incomplete and contradictory information. Pilot does the same thing because environment change and he will have exceptions. In the 1970s, J. McCarthy, D. McDermott, Reiter and others started studies on non-monotonic inference, deriving in default reasoning, autoepistemic reasoning and more others. A robust formalization with exceptions is that Reiter proposed, default logic [12]. In default logic, a default theory is a pair $\Delta = (D, W)$, where D is a set of defaults and W is a set of formulas strictly in FOL. A default d is : $\frac{A(X):B(X)}{C(X)}$, where $A(X), B(X), C(X)$ are well-formed formulas. Where

$X = (x_1, x_2, x_3, \ldots, x_n)$ is a vector of free variables(non-quantified). $A(X)$ are the ***prerequisites***, $B(X)$ are the ***justifications*** and $C(X)$ are the ***consequences***. Intuitively a default means,"if $A(X)$ is true, and there is no evidence that $B(X)$ might be false, then $C(X)$ can be true". When defaults are used it means extensions are calculated. An extension of a default theory Δ is a set E of logical formulas [12] with the smallest set that must verify the following property : If d is a default of D, whose the prerequisite is in E, without the negation of its justification is not in E, then the consequent of d is in E.

Definition 1. E is an extension of Δ iif :
— $E = \bigcup_{i=0}^{\infty} E_i$ with :
— $E_0 = W$ and
— for $i > 0$, $E_{i+1} = Th(E_i) \cup \{C(X) \mid \frac{A(X):B(X)}{C(X)} \in D, A(X) \in E_i \wedge \neg B(X) \notin E\}$
where $Th(E_i)$ is the set of formulas derived from E_i.

The previous definition is difficult to compute in practice. Because $\neg B \notin E$ supposes that E is known, but E is not yet calculated. In the case of normal defaults, $B(X) = C(X)$, E is an extension of Δ iif : we replace $\neg B(X) \notin E$ by $\neg C(X) \notin E_i$. According to Reiter if all defaults are normal, it exists at least one extension. Extensions are defined such as fixed points.

Example 1. Using default logic, from predicates (1) and (2) we have 3 defaults, which contains general information about altitude, where $alt = altitude, x = airplane, std\_fgt = steady\_flight$:

$$d_1 = \frac{((alt(x) > 500) \wedge roll(x, stable)) : std\_fgt(x)}{std\_fgt(x)} \quad (3)$$

$$d_2 = \frac{((alt(x) \leq 500) \wedge roll(x, stable)) : land(x)}{land(x)} \quad (4)$$

$$d_3 = \frac{(land(x) \wedge obstacle) : climb(x)}{climb(x)} \quad (5)$$

In natural sense, d_1 describes if x has an altitude more than 500 feet with a stable roll, and it is possible that x is in a steady flight, then x is in a steady flight. Default d_2 describes that if x has an altitude less than 500 feet with a stable roll, and it is possible that x lands, then x lands. And default d_3 describes if x lands and there is an obstacle, and it is possible to climb, then x climbs. Now, we are going to use these three defaults assuming that we have the following information :

$$W = \{(alt(x) \leq 500), roll(x, stable), obstacle\} \quad (6)$$

From $\Delta = (D, W)$, we calculate the set of extensions. We find $E_1 = W \cup land(x)$, where x lands, by using the default d_2. On the other hand, we find $E_2 = W \cup climb(x)$, where x climbs, by using the default d_3. We have two coherent solutions. Solving the problem of contradictory information.

There are mandatory rules that cover flight physics, security and more. For instance, in case of engine failure, x lands. Or if there is an obstacle in the runway, x must not land. But if x has a fault, the pilot must land to not die, so the risk is huge. In this case, the extension calculated will have a high weight. When different solutions are computed we should take into account criteria such as emergency, security, regulation, energy, etc. to choose the better decision. Situations are constantly changing because environment change. Using probabilities to choose one of them it is not the idea, such as Weighted Product Model or Weighted Sum Model [15]. We propose another manner to make decisions from a different point of view. We are in uncertain framework, we consider a non-probabilistic model. We focus on the opportunist model [2]. This model creates an opportunistic loss (or regret) matrix [13]. Formally, the set of regrets is defined as :

$$\forall E, \exists\, m_r = min\,\{max\,(c_i) - c_j\} \quad (7)$$

Where m_r is the minimization of the difference between the maximum value of the criteria c_i and alternatives c_j, this is for all extensions.

Example 2. Let us consider extensions and criteria. Criteria are information about the system or environment. Having two extensions, E_0 and E_1. E_0 has a higher value of

TABLE 1 – Criteria Table.

EXTENSION	**CRITERIA** *Energy*	*Risk*
E_0	5	2
E_1	2	3

energy than E_1, that is, E_0 has a good status of battery or gas, for example. On the other hand, E_0 is less dangerous that E_1, in terms of risk (e.g. agglomeration). Regrets are calculated, $E_0 : \{0, 1\}$ and $E_1 : \{3, 0\}$. In order to obtain the better decision of these two extensions. E_0 is the better option which minimize the risk (m_r), which make sense, in real-life if an airplane has enough energy and pilot makes actions that are not dangerous, he will choose them.

Until now we present how to solve a problem with incomplete and contradictory information as well as the way to choose the better option when we have several extensions. In the next section we are going to introduce new concepts, taking into account the property of resilience and non-monotonic to reason.

4 Non-monotonic Model

When a pilot has a disturbance of any kind, he will naturally move away from the **objective** (O), this it could be land, take-off, climb...However, he must make actions to achieve the goal. Pilot is in constant revision of behavior, taking information from the cockpit, from the environment and even from the control tower. Additionally, pilot should respect air regulations and navigation laws. For a better understanding, we introduce the following concepts.

4.1 Situations, Objectives and Actions

Firstly, the set of *situations* (S) contains information about parameters of the airplane (altimeter, airspeed, variometer...), environment, etc. On the other side, the set of *actions* (A) are what the pilot does physically (increase or decrease the engine power, turn the yoke to the left or right, ...) to the airplane. In this context, the ***situations*** and ***actions*** are represented by positive literals. We consider that for a certain situation, the challenge is to calculate the extensions that contain actions which allow to approach the desired ***objective*** (O). For instance, when an airplane is placed at the start point (S_p), Fig. 1, assuming it has the authorization, and it is possible to take-off, then the plane take-off. This objective could be described by a default as follows :

$$\frac{(rest(x) \wedge authorization) : takeoff(x)}{takeoff(x)} \tag{8}$$

In the same way, we could describe when a plane (starts at some point a) wants to maintain an altitude greater than 1500 feet with a north direction, to reach to the point b. A default could be as follows :

$$\frac{((alt(x) > 1500) \wedge compass(x, north)) : point(x, b)}{point(x, b)} \tag{9}$$

These are just two defaults as examples, but we can include many others in O. We consider two kind of objectives, short and long-term. The ***short-terms*** occur when there are perturbations and airplane moves away from the long-term objective. Thus, pilot will find another short-goal to get closer and converge. For instance, when the airplane is climbing (from the point a to the point b, Fig. 1) to an altitude of 1500 feet and there are wind disturbances, equation (8) is considered a sub-goal. On the other hand, a ***long-term*** objective is, for instance, maintain a steady flight for 5 minutes with an altitude of 1500 feet, equation (9) is considered a long objective.

Definition 2. In the world $\mathbf{K}$, there is always a resilience trajectory R.

$$\forall S, \forall O, \forall A \subseteq K \; \exists \, R \tag{10}$$

Short-term objectives have very fast change in comparison with long-term. Nevertheless, short-terms will allow to achieve long-term. As the system evolves and disturbances appear, exploration is an important stage of the model. Because this part it is the main process to find different sub-goals that will allow to absorb the shock ζ. Sub-goals g are related to the extensions since they contain actions to converge to the final goal. It is so that the system can jump between sub-goals and have a resilient behavior. If

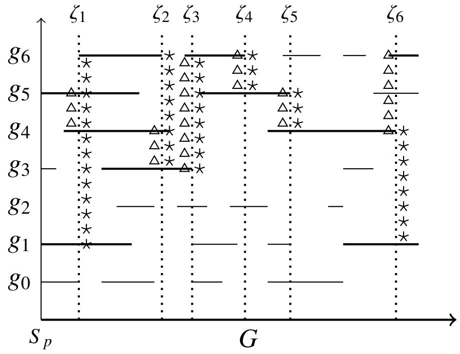

Figure 2 – Evolution of a goal G, switching sub-goals g when a disturb ζ occurs : $\triangle$ and $\star$ are the trajectories created.

we take a look at the discrete representation, Fig. 2. We have at the begging S_p the computation of four extensions : $\{g_0, g_1, g_3, g_5\}$, according to our decision-making model g_1 is chosen and then the system interacts with the environment. At some point, disturb ζ_1 occurs and extensions are computed one more time : $\{g_1, g_4, g_5, g_6\}$, the better solution is g_6 and then interaction happens again. This process occurs every time disturbs ζ appear. In this sense, computing and choosing extensions, trajectories ($\triangle$, $\star$) are created. For the first resilient trajectory $\triangle$, we have : $R_\triangle = \{g_5, \zeta_1, g_4, \zeta_2, g_3, \zeta_3, g_6, \zeta_4, g_5, \zeta_5, g_4, \zeta_6, g_6, \ldots\}$ and for the second trajectory ($\star$), we have : $R_\star = \{g_1, \zeta_1, g_6, \zeta_2, g_3, \zeta_3, g_6, \zeta_4, g_5, \zeta_5, g_4, \zeta_6, g_1, \ldots\}$

4.2 Model

We present a model that describes the evolution of pilot reasoning. In reality pilot makes two movements, he observes the horizon and next the cockpit, after that he does actions, he repeats this over and over again. This dynamic could be represented such as Fig. 3. The model has transitions but the notion of time is not considered. For example, if we have a situation s_i and if it is possible to go to the situation s_{i+1}, we should do actions. Firstly, we start with

$$\begin{tikzcd}
 & E = \{E_1, E_2, ..., E_n\} \arrow[r] & Choice \arrow[d] \\
\Delta = \{D, W, S\} \arrow[ur] & & E_n = \{A, S^+\} \arrow[d] \\
 & & \Delta^+ = \{D, W, S^+\} \arrow[dl] \\
 & \Delta \leftarrow \Delta^+ \arrow[uul] &
\end{tikzcd}$$

Figure 3 – Reasoning of a pilot based on default logic.

a default theory $\Delta = (D, W, S)$. Where D are the set of defaults, W are the set of FOL and S are the parameters of the airplane, environment, control tower, etc. We are considering that Δ is a default theory before a transition and Δ^+ is a default theory after a transition. Similarly, S is a situation observed before a transition and S^+ is a situation observed after a transition. From Δ, the set of extensions E is computed. Each extension contains actions. Once we have the solutions we must choice the better extension that brings us closer to the goal, then decision-making is based as before. After pilot applies actions he takes observations again (cockpit and environment) passing information from S to S^+. Then it goes back to Δ to compute extensions and choose the better one again. Sometimes for an airplane it is impossible to converge to the desired goal and alternative objectives must be found. In the set of objectives (O) the property of resilience is carried out. This is the property will be described in the next section.

5 Resilience

The concept of resilience is defined as the property of a system to absorb and anticipate perturbations [4]. So to apply this property to our model, let consider a knowledge world K, Fig. 4, which contains the set of *situations* (S), *objectives* (O) and *actions* (A). Inside the world K we will study the property of resilience. Since we know that the logic model has an evolution, we are interested to study its form and properties. We can define a trajectory as the satisfaction of 4 main properties : ***reorganization*** (α), ***exploration*** (β), ***release*** (γ) and ***conservation*** (δ), Fig. 4 [6, 14, 16]. We consider that *Non-monotonic reasoning* is exploration, *Choice* is reorganization and conservation, and finally interaction with environment is release. Trajectory has a form as closed loop that converge a stable equilibrium [5]. In control theory [8] stability is defined as follows :

Definition 3. A non-linear time-invariant system with $x' = f(x), f : \mathbb{R}^n \to \mathbb{R}^n$. It has a point $x_e \in \mathbb{R}^n$ is an equilibrium point of the system if $f(x_e) = 0$. It is global asymptotically stable, if for every trajectory $x(t)$, we have $x(t) \to x_e$ as $t \to \infty$. It is locally asymptotically stable near or at x_e if there is $R > 0$, s.t. $\|x(0) - x_e\| \leq R \Rightarrow x(t) = x_e$ as $t \to \infty$.

We consider Lyapunov's definition for our study. For every short and long-term objective $\mathbf{O}$ a neighborhood (ϵ) of the exact point of convergence is defined as follows :

$$0 < \|\mathbf{O}\| < \epsilon \tag{11}$$

In our model, stability will be when every objective $\mathbf{O}$ is inside ϵ. In this context, we define : $R \equiv O$, where R is the theoretical trajectory of resilience and O is the trajectory of objectives of our model [6]. If the equivalence is valid, then the system will have stability in term of resilience. Theoretical trajectory is defined as follow : $R : \{\ldots \alpha, \beta, \gamma, \delta, \alpha, \ldots\}$. An interesting point of the model is if we increase the number of defaults, we will increase the degrees of freedom. This is an important remark, because we consider degree of freedom a space in $\mathbf{O}$ where it could pass a trajectory.

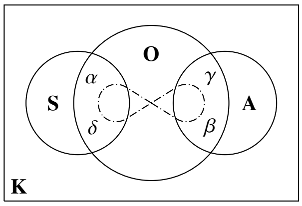

Figure 4 – Non-monotonic Resilience Stability.

6 Implementation

We are using a reduced-size model of motor-glider with a wingspan of 1366mm (53.75 in.), an overall length of 977mm (38.5 in.) and a HBM 2812-1100 Brushless Motor. On board a microcomputer based on Linux operating system is installed, it has the next characteristics : a cpu running at 1 GHz ARM11 (single core), 512 Mb of RAM and power consummation of 0.8 Watts. The microcomputer contains physical digital ports with serial communication protocols which allow to connect different devices. The inertial sensor provides the accelerations, angular velocities and measurements of the earth's magnetic field. These three information, it allows us to know the orientation of the motor-glider in space, for instance, if it is going up, down, turning. . .Altitude is provided by the GPS module but also it is calculated by an atmospheric pressure sensor. Pitot tube is an instrument that allows to measure the static and dynamic pressure, and thus to know the airspeed of the airplane, based on the Bernoulli's equation. For obstacle detection an ultrasonic sensor is used, with a max. detecting distance of 4-5m. The aileron control is done by servomotors through PWM signal. The circuit on board is supplied with 11.1 Volts and 1300 mAh LiPo battery. In the microcomputer, SWI-Prolog was installed. Until now, we have 80 defaults and the extensions are calculated in the order of milliseconds. However, if we increase the number of defaults, the calculation time does not increase much, since horn clauses are used [7, 3].

7 Conclusion

We introduced a resilient model for an autonomous airplane, using non-monotonic logic, in particular, default logic. We tackled contradictory and incomplete information to manage aviation rules and make decisions. We defined a non-probabilistic model to choose an extension considering criteria such as security, energy, emergency...We used the property of resilience to find alternative solutions, when disturbs occur, and converge to the objective or sub-objectives. We described stability using Lyapunov's definition. The implementation is currently in progress with good results. This is a motivation to have a resilient model able to find thermal and to be able to fly as long as possible autonomously. We are also interesting to study Minsky's model, Fig.5, which describes how the mind get goals by changing the set of axioms in use [10, 11]. From our model in the Fig.3, we could consider the "Now" such as the actual situation S and "Want" such as S^+, the long-term objectives. The differences will be the actions that we should do to converge to the main objective.

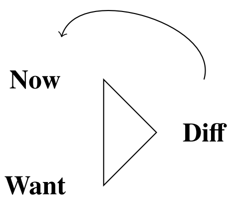

FIGURE 5 – Minsky's model.

Acknowledgements

We thanks to the Secretariat of Energy (SENER) through the Mexican National Council for Science and Technology (CONACYT)[grant number 581317/412566] for their support.

Références

[1] Chandra, Ashik: *Synergy between biology and systems resilience*. Mémoire de maîtrise, MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY, 2010.

[2] Chiles, Todd H et John F McMackin: *Integrating variable risk preferences, trust, and transaction cost economics*. Academy of management review, 21(1) :73–99, 1996.

[3] Doncescu, A. et P. Siegel: *DNA Double-Strand Break-Based Nonmonotonic Logic.* Computational Biology, Bioinformatics and Systems Biology. Elservier, 2015.

[4] Goerger, Simon R., Azad M. Madni et Owen J. Eslinger: *Engineered Resilient Systems : A DoD Perspective*. Dans *Procedia Computer Science*, tome 28, pages 865–872. ELSEVIER, 2014.

[5] Holling, Crawford S: *Resilience and stability of ecological systems*. Annual review of ecology and systematics, 4(1) :1–23, 1973.

[6] Holling, Crawford S: *Understanding the complexity of economic, ecological, and social systems*. Ecosystems, 4(5) :390–405, 2001.

[7] Le, T., A. Doncescu et P. Siegel: *Utilization of Default Logic for Analyzing a Metabolic System in Discrete Time.* ICCSA, 2013.

[8] Lyapunov, Aleksandr Mikhailovich: *The general problem of the stability of motion*. International journal of control, 55(3) :531–534, 1992.

[9] McCarthy, John: *Formalizing common sense*. Ablex, Norwood, New Jersey, 1990.

[10] Minsky, Marvin: *A framework for representing knowledge*. 1974.

[11] Minsky, Marvin: *The emotion machine*. New York : Pantheon, 56, 2006.

[12] Reiter, Raymond: *A logic for default reasoning*. Artificial intelligence, 13(1-2) :81–132, 1980.

[13] Sion, Maurice: *On general minimax theorems*. Pacific Journal of mathematics, 8(1) :171–176, 1958.

[14] Sitterle, Valerie B., Dane F. Freeman, Simon R. Goerger et Tommer R. Ender: *Systems Engineering Resiliency : Guiding Tradespace Exploration within an Engineered Resilient Systems Context*. Procedia Computer Science, 44 :649 – 658, 2015, ISSN 1877-0509. 2015 Conference on Systems Engineering Research.

[15] Triantaphyllou, Evangelos: *Multi-criteria decision making methods*. Dans *Multi-criteria decision making methods : A comparative study*, pages 5–21. Springer, 2000.

[16] Zhang, Xiaoge, Sankaran Mahadevan, Shankar Sankararaman et Kai Goebel: *Resilience-based network design under uncertainty*. Reliability Engineering & System Safety, 169 :364 – 379, 2018, ISSN 0951-8320.