Using adjustable autonomy and human–machine cooperation to make a human–machine system resilient – Application to a ground robotic system

Stéphane Zieba d,⇑, Philippe Polet a,b,c, Frédéric Vanderhaegen a,b,c

a Univ Lille Nord de France, F-59000 Lille, France
b UVHC, LAMIH, F-59313 Valenciennes, France
c CNRS, FRE 3304, F-59313 Valenciennes, France
d Laboratory for Cognitive Systems Science, Department of Risk Engineering, University of Tsukuba, Tsukuba, Japan

A R T I C L E   I N F O

Article history:
Received 11 December 2009
Received in revised form 15 September 2010
Accepted 24 September 2010

Keywords:
Human–machine systems
Adjustable autonomy
Resilience
Human–machine cooperation
Modelling
Algorithms

A B S T R A C T

This study concerns autonomous ground vehicles performing missions of observation or surveillance. These missions are accomplished under the supervision of human operators, who can also remotely control the unmanned vehicle. This kind of human–machine system is likely to face perturbations in a dynamic natural environment. However, human operators are not able to manage perturbations due to overload. The objective of this study is to provide such systems with ways to anticipate, react and recover from perturbations. In other words, these works aim at improving system resilience so that it can better manage perturbations. This paper presents a model of human–robot cooperative control that helps to improve the resilience of the human–machine system by making the level of autonomy adjustable. A formalism of agent autonomy is proposed according to the semantic aspects of autonomy and the agent's activity levels. This formalism is then used to describe the activity levels of the global human–machine system. Hierarchical decision-making methods and planning algorithms are also proposed to implement these levels of activity. Finally, an experimental illustration on a micro-world is presented in order to evaluate the feasibility and application of the proposed model.

1. Introduction

The context for this study is the TAROT project (TAROT is the French acronym for Essential Technologies for Ground Robot Autonomy), launched by French Defence Procurement Agency (DGA). This project is trying to increase the decisional autonomy of ground robotic systems. This study considers a ground robotic system that behaves autonomously to accomplish observation and/or surveillance missions. These autonomous behaviours allow the robot to detect relevant objects in the environment (e.g., trees, walls, fences) and to track them in order to accomplish a mobility task. Such behaviours try to better localize the unmanned robot in an unstructured environment [23,30].

Several questions have been raised concerning the definition of the autonomy modes in the TAROT project and their appropriate selection according to the context. The objective of this project is not to attain complete decisional autonomy for the robot, but to focus on the complementary capacities of the human operator and the robot. Using the complementary capacities of the human operator and the robot led to defining several autonomy modes [35].

⇑ Corresponding author.

E-mail addresses: zieba@css.risk.tsukuba.ac.jp (S. Zieba), philippe.polet@univ-valenciennes.fr (P. Polet), frederic.vanderhaegen@univ-valenciennes.fr (F. Vanderhaegen).

0020-0255/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2010.09.035
These modes are organised around the decision-making activity contained in the model proposed by Parasuraman et al. [22], as shown in Table 1. The decisions correspond to the transitions between the mission goals. The four modes of autonomy range from the teleoperation mode, in which the human operator performs all the actions (M0), to the fully autonomous mode, in which the robot performs its mission autonomously and the human operator is only informed of the mission progress (M3).

In a dynamic context, the human–machine system must be able to select the most appropriate autonomy mode. In TAROT, this selection depends on the context, especially the human operator’s availability. Indeed, this project aims at soliciting the human operator in a relevant way. Fig. 1 presents the different autonomy adjustments possible in TAROT. A horizontal adjustment modifies the autonomous function that is controlling the robot. For this kind of adjustment, the system stays at the same autonomy mode. A vertical adjustment modifies the task sharing between the machine and the human agents; this adjustment will thus modify the autonomy mode of the human–machine system, choosing one of the four possible modes. Logically, the lower the autonomy mode, the more the human operator is involved.

In TAROT, the different adjustments of autonomy are performed according to a cooperative mode of human–robot interaction (i.e., one in which the human operator and the robot are cooperating). Based on the questions raised about this project, this study is thus focused on the human–machine cooperation faced with perturbations. Undeniably, the human–machine system, composed of a human operator and an autonomous robot, is likely to have to deal with perturbations, which may be environmental disturbances, technical failures or human errors. The objective of this study is to provide this system with ways to anticipate, react and recover from perturbations. For the TAROT project, two ways to control resilience of the system have been identified: adjustable autonomy and human–machine cooperation.

The first part of this paper introduces the concept of resilience applied to a human–machine system. The notions of adjustable autonomy and human–machine cooperation are situated in this context. The second part of the paper presents our model of human–robot cooperative control with its formalism of agent autonomy. This formalism is then used to define the activity levels of the human–machine system. The last part of the paper reports the results of our experimental campaign to illustrate the proposed model. The TAROT project is taken as the context of this campaign: a mission of observation in a natural environment by an unmanned robot supervised by a human operator.

**2. Resilience of a human–machine system**

Resilience is a broad concept that can be found in several domains of application. First, ecological or material science approaches defined resilience as a quantifiable property for evaluating the amount of energy that a material can absorb without
breaking or as the speed of system recovery, for instance, after perturbations [9,20,24]. Another set of definitions is more interesting in the field of human–machine systems. These definitions come from organisational theory, economics or safety [12]. In this case, resilience is defined as a conceptual, non-quantifiable system property that permits a stable system state to be maintained and/or recovered, in this way allowing the system to continue operations after a major mishap or in the presence of continuous stress [42].

A resilient organisation is characterised by the following three steps relative to a perturbation, as shown in Fig. 2 [10]. First, the system must be able to monitor itself constantly using several metrics to anticipate a perturbation. These metrics may be state indicators that give information about the current functioning of the different system agents. Second, when a perturbation that cannot be avoided occurs, the system must be able to assess the situation and reorganise itself. Third, after the perturbation has occurred, the different alternatives must be assessed and used to increase the relevance of the metrics used in the first step. Increasing the relevance of the metrics means that the past system experiences will help to determine the metrics that are the most useful for managing perturbations.

Even though resilience is a non-quantifiable property, the resilience potential of a system has to be able to be measured. Several criteria are proposed (i.e., the metrics mentioned above) that help to design a system and measure what contributes most to its resilience. These criteria are based on a review of the literature related to the resilience assessment of complex systems.

1. **Efficiency**: Efficiency is the ability of the system to be effective in a given operational mode (i.e., performance optimisation according to given constraints [6]). Indeed, resilience is characterised by preserving an acceptable level of performance despite perturbations. Thus, efficiency must be a criterion for assessing system resilience.

2. **Adaptability**: Adaptability is the capacity of the system to elaborate context-dependant solutions [1,6,11] and corresponds to the availability of various possible solutions proposed by the system to manage a perturbation. Adaptability was introduced as a resilience criterion by Fiksel in [6] and flexibility for resilient systems is mentioned by Anders et al. [1]. The adaptability criterion proposed in this paper merges the different aspects of these criteria: the ability of the system to restructure itself by adjusting the system autonomy and adapt to new constraints by proposing new action plans or switching to related goals [21].

3. **Border-line functioning**: Border-line functioning is the ability of the system to manage border-line use conditions in a given operational mode [27]. These border-line conditions of use correspond to the phase of instability that follows the occurrence of a perturbation. Border-line functioning assesses the behaviour of the system close to these boundaries of safety [25].

4. **Interaction**: In the case of a complex system, interaction is the ability of the system to initiate and manage the information exchanges between system agents [1,33].

All these criteria can be used to manage the overall system operations. The different measures for controlling system resilience must also be taken into consideration. Two of them, related to the TAROT project, will be focused on this paper: adjustable autonomy and human–machine cooperation. These control measures manage the interaction between system agents.

### 3. Measures for controlling resilience

#### 3.1. Adjustable autonomy

A discussion about adjustable autonomy should begin with the definition of the word, autonomy. Two main directions can be obtained from the etymology of *autonomy*, which is derived from the combination of two Greek words: *autos*, which
means “oneself”, and nomos, which means “law”. A common definition of “autonomy” is the capacity of an individual or a group to take care of itself or not to depend on an external authority.

Two interesting definitions for autonomy are identified. Bradshaw et al. [3] proposed two dimensions to autonomy: the descriptive dimension, which refers to actions that an agent is able to perform, and the prescriptive dimension, which refers the actions that an agent is authorised to perform. The second definition of autonomy is related to the levels of activities that can be accomplished by the agent [43]. Three traditional levels can be identified:

– the agent’s ability to accomplish the prescribed objectives,
– the agent’s ability to adapt to environmental modifications, and
– the agent’s ability to develop its own objectives.

A parallel can be made between these traditional levels and the human operator models [17,28] and the hybrid control architectures in robotics, both of which have the three same levels [4]. As mentioned by Trivino et al. [36], such “architectures lack a view that help the operator to see him/herself as an integral part of the system”. This lack can be due to the different autonomy representations of virtual and human agents. This observation led to the proposal of a definition of autonomy that can be applied to both virtual and human agents [45].

A definition for agent autonomy was thus proposed in order to combine the two previous definitions of autonomy [45] (Fig. 3), including:

– the agent’s skills, capacities and prescriptions to accomplish a given task (i.e., operational level),
– the agent’s skills, capacities and prescriptions to decide how to accomplish a given task (i.e., decisional level), and
– the agent’s skills, capacities and prescriptions to define a plan of action to accomplish a given task (i.e., deliberative level).

This definition thus combines two axes for the autonomy of an agent: the semantic aspects (skills, capacities and prescriptions) and the levels of activity (operational, decisional and deliberative). In all three levels, a distinction is thus made between the agent’s skills, capacities and prescriptions. For example, for a virtual agent, “skills” refer to the existence of technology and algorithms necessary to accomplish a task. “Capacities” can depend on the context and the available resources and are thus likely to be modified, resulting in the agent not being able to accomplish a given task, although it has the skill to do it. Finally, “prescriptions” determine whether or not the agent is allowed to perform a given task or whether or not the agent is compelled to perform a given task.

This definition can be placed in the context of the modes M0 to M3 defined for the TAROT project. For example, in mode M3, the operational and decisional levels are allocated to the robot, while the deliberative level is allocated to both the human operator and the robot. The human operator is still concerned about the mission progress, and he/she can still make some strategic decisions.

The vertical and horizontal adjustments of autonomy defined in the project TAROT have to be situated in this definition of autonomy. Adjustable autonomy can thus be defined according to the two axes of autonomy (semantic aspects and levels of activity).
The first possible autonomy adjustment is to modify the operating space of the agent. This case is close to the definition proposed by Bradshaw et al. [3], with its descriptive and prescriptive dimensions. The adjustment then concerns the actions that the agent is able to accomplish and the actions that it is allowed to accomplish.

Autonomy can also be adjusted at the global level (i.e., the system level). This kind of adjustment consists of adjusting the system’s autonomy mode to one of other levels while the system is operating [5,8]. This adjustment then modifies the task allocation between the agents and the authority for decisional tasks, for instance or another level of activity. Trading authority allows defining dynamically which agent has authority over a given function, either the human operator or the virtual agent [13].

The agents should be able to know their own requirements in terms of delegation and trading of authority [16]. The different autonomy adjustments mentioned can be initiated by the human operator or the virtual agent (i.e., the robot). These two kinds of agents thus have to cooperate in order to decide whether or not an adjustment is required and, if so, what type of adjustment is necessary.

3.2. Human–machine cooperation

3.2.1. Human-robot collaborative control

Some research studies have applied the principles of human cooperation to dynamically share tasks between human operators and automated systems. These shared tasks are decisional tasks (e.g., conflict detection, problem-solving [38,39], diagnosis [40] or image analysing and searching [2,34]) and may concern several human–machine system organisational configurations [37]. Human–machine cooperation can also be described briefly by presenting a human–robot interaction mode: human–robot collaborative control [7].

In this mode of human–robot interaction, the human operator and the robot are placed at the same decisional level. These two agents cooperate to accomplish a common goal by initiating an interactive dialogue. In the approach presented in this paper, this dialogue can take different forms depending on the cooperative situation. This dialogue is accomplished using the various forms of cooperation.

3.2.2. Forms of cooperation

Schmidt [32] defined these forms as augmentative, integrative and debative. These forms of cooperation are adapted to the context of human–robot cooperation [44]. The first form of cooperation — augmentative — is applied when agents have similar skills but insufficient capacities to accomplish the global task alone. The agents share tasks between themselves, thus accomplishing similar tasks. The second form of cooperation — integrative — is applied when the agents have different and complementary skills. Breakdown the global task into specific subtasks, the agents can accomplish the global task by integrating the results of the other agents. The last form of cooperation — debative — is applied when agents have similar skills; they compare the partial or global results of the other agents to determine which agent is going to perform the task. In other words, they find the optimal solution.

3.3. Discussion

To conclude this first part of the paper, some links can be drawn between resilience, adjustable autonomy and human–machine cooperation. As shown in the Fig. 2, resilience to perturbations is defined by the three things: anticipation, reaction and recovery. Anticipating a perturbation is done by using the resilience criteria that were mentioned above (i.e., efficiency, adaptability, border-line functioning, interaction) to decide whether or not an adjustment is necessary and, if so, what type of adjustment. When a perturbation occurs, the system proposes certain solutions. These solutions can be carried out by making vertical and horizontal autonomy adjustments. In fact, for weak perturbations, it may be sufficient to adjust the autonomy at the level of the agent by modifying its capacities or its authorisations. The global system thus remains at the same autonomy mode. For greater perturbations, it may be required to adjust the autonomy mode of the global system by modifying the task allocation between the agents. In order to manage a perturbation, a solution may be to decrease the system’s autonomy mode by involving the human operator in the three activity levels that represent autonomy.

Managing these adjustments will be discussed in detail in the next section. The first step is to define the semantic aspects for the tasks that an agent will accomplish.

4. Model of human–robot cooperative control

4.1. Formalism of autonomy

4.1.1. Notations

This sub-section summarizes the notations used into define the proposed formalism. A short description is provided for the different elements.

- \( \text{Act} \): set of actions achievable by the human–robot system
- \( A \): finite sequence of actions
- \( \text{Pol} \): set of policies for an action \( x \): authorisation, obligation, prohibition, absence of obligation
- \( D \): set of available resources for the system
$U(x)$ set of resources usable by an agent $x$: the agent must have the capacities to use a given resource and the authorisation to access a given resource.

$R(x)$ set of resources required for accomplishing action $x$

$G(x, a)$ set of resources allocated to the agent $x$ for action $a$

$\Theta$ set of tasks achievable by the human–robot system

$T_{ij}$ task defined by a pair of initial and final states $(e_i, e_j)$

$\text{realisable}(A, T)$ predicate assessing the achievability of a task $T$ according to a sequence of actions $A$: the task $T$ can be accomplished by accomplishing consecutively the actions of the sequence $A$

$(T_p, A)$ decomposition of a primitive task: assess a way to accomplish the task $T_p$

$(T_c, d_c)$ decomposition of a compound task: assess the possible ways to accomplish $T_c$

$d_c$ list of the decompositions of a compound task: composed of pairs containing the preconditions for accomplishing the decomposition and the tasks corresponding to the decomposition

$(\text{pre}_k, \text{sub}_k)$ $k$th element of $d_c$ composed of a list of facts.

$\text{pre}_k$ preconditions for accomplishing the $k$th element of $d_c$: composed of a list of facts.

$\text{sub}_k$ list of tasks corresponding to the $k$th element of $d_c$: these tasks can be primitive and/or compound tasks.

$S(x)$ skills set of agent $x$: tasks that can be accomplished by the agent $x$

$C(x)$ set of capacities of agent $x$: tasks that can be accomplished by an agent $x$ at time $t$

$P(x)$ set of prescriptions of agent $x$: tasks with associated policy (i.e., authorisations, obligations, prohibition, absence of obligation)

$A(x)$ set defining the autonomy of the agent $x$: tasks for which the agent $x$ is autonomous

$s(x, T)$ predicate assessing the skills of the agent $x$ to accomplish a task $T$

$c(x, T)$ predicate assessing the capacities of the agent $x$ to accomplish a task $T$

$p(x, T)$ predicate assessing the authorisation of the agent $x$ to accomplish a task $T$

$a(x, T)$ predicate assessing the autonomy of an agent $x$ to accomplish a task $T$

$CA(E, T)$ predicate assessing the application conditions of an augmentative form of cooperation for accomplishing a task $T$ by a set of agents $E$

$CI(E, T)$ predicate assessing the application conditions of an integrative form of cooperation for accomplishing a task $T$ by a set of agents $E$

$CC(E, T)$ predicate assessing the application conditions of a debative form of cooperation for accomplishing a task $T$ by a set of agents $E$

### 4.1.2. Tasks and actions

The definition of the tasks that are to be accomplished by an agent is based on a hierarchical decomposition. Let

$\theta = \{T_1, T_2, \ldots, T_m\}$ be the set of achievable tasks by the human–robot system. A task $(T_{ij})$ is defined as a pair of states (i.e., initial $(e_i)$ and final $(e_j)$ states) such that

$$T_{ij} = (e_i, e_j).$$

(1)

Two types of tasks are distinguished: compound tasks and primitive tasks. A compound task corresponds to the highest level of decomposition and is related to the goals defined for the system. This type of task does not concern a single agent. Several agents may indeed be involved in accomplishing a compound task. A compound task is defined as a task that can be recursively decomposed into at least one sequence of subtasks; these subtasks can be either compound tasks and/or primitive tasks. Fig. 4 illustrates the definition of a compound task.

The accomplishment of a compound task is defined by the following pair, $(T_c, d_c)$, where $T_c$ is the compound task and $d_c = (\{\text{pre}_1, \text{sub}_1\}, \ldots, \{\text{pre}_m, \text{sub}_m\})$ is the list of the possible decompositions of $T_c$ needed to accomplish $T_c$. An element on the list $d_c$ is a pair, $(\text{pre}_k, \text{sub}_k)$, composed of preconditions $\text{pre}_k$ and the decomposition $\text{sub}_k$ composed of compound and/or primitive tasks. Preconditions correspond to a list of facts that are required to accomplish a given decomposition. When the preconditions $\text{pre}_k$ are verified, accomplishing the task $T_c$ using the decomposition $\text{sub}_k$ is possible.

![Fig. 4. Illustration of a compound task.](image-url)
A primitive task is defined as a task that can be decomposed into at least one finite sequence of actions. The most basic elements of the formalism are the actions that compose the action sequences of primitive tasks. Actions are usually defined by preconditions, effects and required resources. Preconditions and effects are defined as lists of facts.

Let \( \text{Act} = \{x_1, x_2, \ldots, x_n\} \) be the set of actions. An action \( x_k \) can be considered as an elementary function between two states, \( e_i \) and \( e_j \), such that

\[
 x_k(e_i) = e_j \quad \text{where} \quad x_k \in \text{Act}. \tag{2}
\]

Accomplishing of a primitive task using a sequence of actions is defined by the following pair, \( (T_p, A) \), where \( T_p \) is the primitive task and \( A \) is a finite sequence of actions such that

\[
 A = (x_1, \ldots, x_m). \tag{3}
\]

A task \( T_y \) is considered to be primitive and achievable if there is at least one decomposition into a finite sequence of actions \( A = (x_1, x_2, \ldots, x_m) \), such that

\[
 x_m(x_{m-1}(\ldots x_1(e_i))) = e_j. \tag{4}
\]

The predicate realisable \( (A, T) \) is defined from this sequence of actions. This predicate judges that a task \( T \) is achievable using the sequence of actions \( A \), such that

\[
 \text{realisable} (A, T_y) \leftrightarrow \exists A = (x_1, x_2, \ldots, x_m) \in \text{Act}^m, \quad x_m(x_{m-1}(\ldots x_1(e_i))) = e_j. \tag{5}
\]

Similarly to compound tasks, a primitive task can be decomposed into several different sequences of actions. In the case illustrated in Fig. 5, the primitive task can be accomplished by three different sequences of actions.

Each action is characterised by resources necessary to accomplish it. Several notions related to the resources have to be defined in order to take the dynamic aspect of the system into account. Let \( D = \{r_1, r_2, \ldots, r_m\} \) be the set of available resources. The system’s initial resources when no action is currently being accomplished are supposed to be known, limited and quantifiable. Let \( U(x) \subseteq D \) be the set of resources that can be used by an agent \( x \). An agent is not permitted to use a resource for one of two reasons:

- the resource is not included in its set of available resources,
- the agent does not have the skills, capacities or authorisations to use the resource; in other words, it cannot autonomously use the given resource.

Let \( R(x) \) be the set of required resources for accomplishing action \( x \) (i.e., to apply the effects of the action \( x \)). The set \( G(x, x) \) is defined as the set of resources allocated to the agent \( x \) for accomplishing action \( x \), such that

\[
 G(x, x) = U(x) \cap R(x). \tag{6}
\]

Using the definitions of tasks and actions, the semantic aspects of autonomy (i.e., the skills, capacities and prescriptions) included in the three activity levels for agent autonomy (Fig. 3) can now be defined.

### 4.2. Semantic aspects of autonomy

#### 4.2.1. Definition of skills

The first semantic aspect concerns an agent’s skills. Skills refer to the action sequences known by a given agent that are needed to accomplish a task. Skills are thus an application of the predicate realisable \( (A, T) \) that defines the potential for accomplishing primitive tasks. A task is included in the skills of an agent if the agent knows at least one action sequence needed to accomplish this task.

Let \( S(x) \) be the set of skills of an agent \( x \),

\[
 S(x) = \{(A, T_y) \in \text{Act}^n \times \emptyset | \text{realisable} (A, T_y) \}. \tag{7}
\]

The predicate \( s(x, T_y) \), which evaluates whether or not agent \( x \) has the skills to accomplish a task \( T_y \), is defined by

\[
 s(x, T_y) \leftrightarrow \exists A = (x_1, x_2, \ldots, x_m), \quad (A, T_y) \in S(x). \tag{8}
\]
Fig. 6 illustrates the skills of the agent \( x \). This agent knows the different sequences of actions needed to accomplish three different tasks. For two of these tasks \( (T_{08} \text{ and } T_{09}) \), the agent knows several sequences for accomplishing the same task.

### 4.2.2. Definition of capacities

The second semantic aspect concerns an agent’s capacities. Capacities refer to the tasks that an agent is able to accomplish in a given context. Two conditions have to be fulfilled for the agent’s capacities. First, the agent must be competent, which means that the agent must have the required skills to accomplish the task. Second, the agent must also have the resources required to accomplish every action in the sequence corresponding to the task.

The predicate \( c(x, \Delta) \), which evaluates whether or not agent \( x \) has the capacities and resources to accomplish a task \( \Delta \), is defined by

\[
c(x, \Delta) \leftrightarrow (\exists \Delta \in \text{Act}^m, \text{realisable}(\Delta, T_{ij}), \forall x_k \in \Delta. R(x_k) \subseteq G(x, x_k)).
\]  

Fig. 7 illustrates the capacities of agent \( x \) using the example introduced in Fig. 6. The dotted lines represent the sequences of actions for which the agent has the required resources. Since the agent has the resources to accomplish one of the sequences of actions that allow the task to be accomplished, the task can be included in this agent’s capacities list.

### 4.2.3. Definition of prescriptions

The third semantic aspect concerns an agent’s authorisations/prohibitions with respect to the actions. The prescriptive dimension of autonomy can be defined in relation to authorisations and obligations, distinguishing positive \( (A^+) \) and negative \( (A^-) \) authorisations and positive \( (O^+) \) and negative \( (O^-) \) obligations. Bradshaw et al. [3] defined the following set of policies: \( \text{Pol} = (A^+, A^-, O^+, O^-) \).
Let \( P(x) \) be the set of prescriptions of an agent \( x \),
\[
P(x) = \{(\alpha, \text{pol}) \in \text{Act} \times \text{Pol}\}.
\]
The predicate \( p(x, T_{ij}) \), which evaluates whether or not the agent \( x \) is allowed to accomplish the task \( T_{ij} \), is defined by
\[
p(x, T_{ij}) \iff (\exists A \in \text{Act}^m, \text{ realisable } (A, T_{ij}), \forall \alpha_k \in A, \exists (\alpha_k, \text{pol}) \in P(x), \text{ pol} \neq A^{-}).
\]
The policy associated to the action must not be a negative authorisation (i.e., a prohibition), so that the agent is authorised to accomplish a task. Fig. 8 illustrates the authorisations/prohibitions of an agent \( x \) using the example introduced in Fig. 6. The dotted lines represent the tasks that the agent is authorised to accomplish. The predicate illustrates the authorisations/prohibitions that justify that some sequences of actions are not valid. In fact, the policy associated to an action in the sequence is a prohibition, which makes the whole sequence impossible to accomplish. However, no prohibition is associated to the actions composing the two sequences indicated with the dotted lines.

4.2.4. Definition of agent autonomy

The autonomy of an agent to accomplish a given task \( T_{ij} \) is defined by the following predicate:
\[
a(x, T_{ij}) \iff s(x, T_{ij}) \land c(x, T_{ij}) \land p(x, T_{ij}).
\]

An agent is thus autonomous to accomplish a given task in any of the three activity levels defined in Fig. 3 if it has the skills and capacities to accomplish it and if it is authorised to accomplish it. Based on the examples given in Figs. 6–8, the agent is autonomous to accomplish the task \( T_{08} \). Indeed, it has the skills, the capacities and the authorisation to accomplish this task using a given sequence of actions.

Agent autonomy can also be defined for a given alternative, a given sequence of actions \( A' \), expressed by the following notation:
\[
a_{A'}(x, T_{ij}).
\]

In addition, agent autonomy can be defined as the set of tasks for which an agent is autonomous:
\[
A(x) = \{T_{ij} \in \theta | a(x, T_{ij})\}.
\]

All the predicates defined above can be placed at the three agent activity levels (see Fig. 3). The predicates that define the skills, the capacities and the authorisations are applied to the tasks at the three levels of activity. At each level, the tasks for which the agent is autonomous can be obtained.

The semantic aspects of autonomy will allow the way the agents cooperate to be defined according to their respective autonomies. To achieve this objective, the conditions of application for human–machine cooperation are defined using the forms of cooperation introduced in Section 3.2.2.

5. Conditions of application for human–machine cooperation

5.1. Conditions for augmentative cooperation

The application conditions for the augmentative form are linked to the capacities of the agents and their resources. The sum total of their resources must be included in the resource set required to accomplish a global task. The agent set has the capacities to accomplish the global task, whereas the single agent does not. The augmentative form is applied when agents have similar skills but must aggregate their capacities to accomplish a task.
Let \( CA(\{x, y\}, T_i) \) be the required conditions to use augmentative cooperation between two agents \( x \) and \( y \) in order to perform a task \( T_i \):

\[
\begin{align*}
(s(x, T_i) \land s(y, T_i)) & \land \\
(c(x, T_i) \land c(y, T_i)) & \land \\
(p(x, T_i) \land p(y, T_i)) & \rightarrow CA(\{x, y\}, T_i).
\end{align*}
\] (16)

This form of cooperation is illustrated in Fig. 9. Although cooperative activities are not explicitly mentioned in Fig. 9, for augmentative cooperation, it is a case of mutual control. Two agents \( x \) and \( y \) do not have the capacities to accomplish the global task, so they are not autonomous. Their resources are combined and the global task is broken down into similar subtasks that can be accomplished by each of the two agents.

### 5.2. Conditions for integrative cooperation

Integrative cooperation is one response to agent specialization. Agents have different skills, capacities and/or authorisations. The application conditions for this form of cooperation are linked to the autonomy that the agents have to accomplish the task in which they are specialised. The agent must have the skills, the capacities and the authorisation to accomplish the task.

Let \( CI(\{x, y\}, T_i) \) be the required conditions to use integrative cooperation between two agents \( x \) and \( y \) to perform a task \( T_i \):

\[
\begin{align*}
(\exists T_{i,k} \in 0, T_{i,j} \in 0, T_i = T_{i,k} \cup T_{i,j}) & \land \\
(s(x, T_{i,k}) & \land s(y, T_{i,j})) \land \\
(c(x, T_{i,k}) & \land c(y, T_{i,j})) \land \\
(p(x, T_{i,k}) & \land p(y, T_{i,j})) & \rightarrow CI(\{x, y\}, T_i).
\end{align*}
\] (17)

These conditions can be simplified by using the agent autonomy definition:

\[
(\exists T_{i,k} \in 0, T_{i,j} \in 0, T_i = T_{i,k} \cup T_{i,j}) \land \rightarrow CI(\{x, y\}, T_i).
\] (18)

This form of cooperation is illustrated in Fig. 10. There are two agents, \( x \) and \( y \). Agent \( x \) is autonomous to perform only a part of the global task; agent \( y \) is autonomous to perform the remaining parts of the task. The global task can be accomplished by combining the activities of both agents. Therefore, the conditions required to use integrative cooperation are satisfied.

### 5.3. Conditions for debative cooperation

Debative cooperation is used in a situation in which the agents have similar skills, and they “discuss” to determine which agent is going to perform the task. The application conditions are linked to the autonomy of the agents to accomplish a specific sequence of actions that allows the global task to be accomplished.

Let \( CC(\{x, y\}, T_i) \) be the required conditions to use debative cooperation between two agents \( x \) and \( y \) to perform a task \( T_i \):

\[
\begin{align*}
(\exists A' \in Act^m) \land \exists A'' \in Act^m) & \land \land \\
(\forall A' \neq A'', (A', T_i) \in S(x), (A'', T_i) \in S(y)) & \land \\
(s(x, T_i) & \land s(y, T_i)) \land \\
(c(x, T_i) & \land c(y, T_i)) \land \\
(p(x, T_i) & \land p(y, T_i)) & \rightarrow CC(\{x, y\}, T_i).
\end{align*}
\] (19)
Like with integrative cooperation, the conditions can be written as follows:

\[
\begin{align*}
\exists \Delta' \in \text{Act}^m, \text{realisable } (\Delta', T_{y}) \\
\exists \Delta'' \in \text{Act}^m, \text{realisable } (\Delta'', T_{y}) \\
\Delta' \neq \Delta'', (\Delta', T_{y}) \in S(x), (\Delta'', T_{y}) \in S(y), \\
(a(x, T_{y}) \land a(y, T_{y})) \rightarrow CC(\{x, y\}, T_{y}).
\end{align*}
\]  

(20)

Fig. 10. Illustration of the activity of two agents in an integrative cooperation.

Fig. 11. Illustration of the activity of two agents in a debative cooperation.

Fig. 12. Global view of the levels of activity of the system.
The debative form of cooperation is illustrated in Fig. 11. Two agents have two different alternative solutions and are autonomous to accomplish the global task with either. In this case, debative cooperation can be used and consists of comparing the results of the alternative solutions for certain subtasks or for the global task. The agents cooperate by negotiating the different plans of action.

Section 5 defined the ways that the agents cooperate. The next section is devoted to the activities that result from this cooperation on the level of the human–machine system.

```plaintext
PLANNING(goal, state_of_the_system, list_of_alternatives)
s = state_of_the_system, T = goal, alternative = ()
T₀ = \{t \in T | \text{predecessors}(t) = \emptyset\} // task without preceding task
repeat // no more tasks in the goal to achieve
    if T = \emptyset then add alternative to list_of_alternatives
    else
        choose a task t \in T₀
        if \text{primitive_task}(t) then // processing of a primitive task
            M = \{m | \text{realisable}(m,t), \text{precond}(m)\}
            if M = \emptyset then return failure
            else
                choose m \in M not included in list_of_alternatives
                // search for an autonomous agent for the global task
                E₉ = \{agt \in Agt | \text{a}(agt,t)\}
                if Card(E₉) ≥ 1 then // at least one agent is autonomous
                    choose agt \in E₉ not included in list_of_alternatives
                    for each action a in m
                        modify s according to \text{effets}(a)
                        add a to alternative
                    else // no agent is autonomous for the global task
                        // search for applicable forms of cooperation
                        if CA(F,t) then
                            // allocation of similar tasks to the agents of F
                            allocate t to the agents of F
                            for each action a in m
                                modify s according to \text{effets}(a)
                                add a to alternative
                        else if CI(F,t) then
                            // allocation of specific subtasks to the agents of F
                            decompose t in \text{card}(F) specific tasks t’
                            allocate t’ to the agents agt in F such as a(agt, t’)
                            for each agent agt in F
                                for each action a in m
                                    modify s according to \text{effets}(a)
                                    add a to alternative
                        else return failure
                        modify T by removing t
        T₀ = \{t \in T | \text{predecessors}(t) = \emptyset\}
    else
        if \text{compound_task}(t) then // processing of a compound task
            \[N = \{n | \text{realisable}(n,t), \text{precond}(n)\}\]
            if N = \emptyset then return failure
            else
                choose n \in N not included in list_of_alternatives
                modify T by replacing t by \text{sub}(n)
                if \text{sub}(n) ≠ \emptyset then T₀ = \{t \in \text{sub}(n) | \text{predecessors}(t) = \emptyset\}
                else T₀ = \{t \in T | \text{predecessors}(t) = \emptyset\}
```

Fig. 13. Planning algorithm.
6. Definition of the system's activity levels

The goal of this study is to extend the three agent activity levels (i.e., operational level, decisional level and deliberative level) to the global system. The agents can be the human operator and/or the robot. In the system, three processes that correspond to these activity levels are the planning process, the decision-making process and the goal satisfaction process. The functions of these levels are detailed in the following sub-sections.

6.1. Functioning of the activity levels

The global functions of these three activity levels are illustrated in Fig. 12. First, a planning algorithm generates a list of alternatives. These alternatives use the forms of cooperation to allocate the tasks between the system agents. One of the alternatives is selected through the decision-making process. A multi-criteria decision-making method uses the resilience criteria to select the most appropriate alternative. The selected alternative is likely to adjust the autonomy either at the level of the agent by modifying their capacities or prescriptions, or at the level of the system by changing the autonomy mode among the four possible modes defined in the TAROT project. These adjustments condition the forms of cooperation that can be used according to the application conditions defined in the previous section. The form of cooperation that corresponds to the selected alternative controls the satisfaction of the current goal by managing the cooperative activities between the agents. When the goal cannot be satisfied, due to the preconditions of the next action, a request is addressed to the planning algorithm in order to generate a new plan of action.

6.2. Description of the planning algorithm

The planning algorithm is based on the SHOP algorithm (see Fig. 13). This algorithm is an ordered planner [19], which means that the actions are planned in the order in which they will be accomplished. A task without preceding tasks (evaluated by the predicate \textit{predecesseurs(t)}) is selected from the goal to be accomplished, which is a compound task. If the selected task is also a compound task, the selection is repeated on the first task in the breakdown of the compound task, since the algorithm is an ordered planner. If the selected task is a primitive task, the predicates for autonomy and cooperation are evaluated. The tasks are then allocated to the agents according to their autonomy or according to the applicable form of cooperation. Each alternative generated by the algorithm is placed on a list of algorithm parameters so that the debative cooperation can be initiated when several plans of action involving the same agents are available. The alternatives on the list involving the same agents are then compared in order to determine the optimal solution between different possibilities.

6.3. Selection of an alternative

When the list of alternatives has been generated, the decision-making process must select one of the alternatives. This process uses the multi-criteria Analytic Hierarchy Process (AHP). (For an extensive description of this decision-making method, the interested reader can consult Saaty's work [31]). The criteria used in this study are the resilience criteria that were defined in Section 2: efficiency, adaptability, border-line functioning and interaction. These criteria are broken down into indicators that are used to evaluate the alternatives generated by the planning algorithm. Fig. 14 shows one decomposition of the problem using this method.

This method is interesting for human–machine systems because it can include human operator judgments. This judgment is used to determine the preferences granted to each criterion. These preferences are computed from binary comparisons between all the criteria.

The alternatives are compared according to the objective indicators (i.e., the criteria on the lowest hierarchical level). The values of the comparisons are then converted in a value on the scale used in the AHP method. Finally, a score is computed for each alternative according to their respect of the criteria and the indicators.

Our model of human–robot cooperative control was simulated on a micro-world. This simulation is presented in Section 7.

![Fig. 14. Decomposition of the problem through a multicriteria approach.](image)
7. A simulation of our cooperative control model

7.1. Presentation of the simulation

The micro-world used for the experimental campaign simulates the context of the TAROT project: observation missions by an unmanned ground robot in a 3D natural environment. The autonomous functions included in the simulation allow the robot to detect relevant objects in the environment and to track them in order to accomplish the task of autonomous mobility. This task is decomposed into several goals and subtasks, which correspond to different waypoints. These waypoints have to be reached by using specific modes of autonomy and/or autonomous behaviours (e.g., edge-following, object-following, teleoperation).

Missions are supervised by a human operator. The human operator also has to accomplish an observation task, which consists of detecting objects in the environment and locating them on a map. Fig. 15 shows the human–machine interface of the simulation.

7.2. Objectives of the experimental protocol

The main objective of the experimental protocol was to evaluate the feasibility of implementing the proposed predicates to manage the autonomy adjustments. This experimental simulation aims to illustrate human–machine cooperation in managing perturbations through the alternatives proposed by the planning algorithm.

These alternatives initiate the augmentative and integrative forms of cooperation and can lead to autonomy adjustments, either by modifying the autonomy mode or the agent's capacities and/or authorisations, especially the autonomous mobility functions. The predicates for evaluating agent autonomy and the application conditions for human–machine cooperation were implemented. Planning and decision-making processes were simulated in order to obtain similar scenarios for all the operators. The alternatives and their rankings were thus determined beforehand.

The experimental protocol includes a comparison between the autonomy modes used in the TAROT project and the proposed model of human–robot cooperative control. In this paper, the baseline is the approach used in the TAROT project [35]. The autonomy mode M2 defined in TAROT is similar to cooperative control as it is defined in terms of collaborative control by Fong et al. [7]. This autonomy mode M2 is considered as a tentative approach to cooperation between unmanned system and experienced operators.

An experimental campaign was carried out during the TAROT project with experienced operators. These operators were used to controlling ground vehicles remotely but had little experience with systems designed for human–machine
cooperation and adjustable autonomy. The preliminary conclusions of this experimental campaign highlighted the importance of the information required to understand the reasoning of the system [35].

The algorithm proposed in this paper is intended to share and delegate planning and decision-making activities to the unmanned system, especially in cases of perturbations. Some experimental results are presented below (Section 7.4) and discussed in order to show the relevance of the resilience criteria defined in Section 2.

7.3. Description of the experimental protocol

Fifteen subjects participated in this experimental campaign. They were students from the Université de Valenciennes and thus constitute a homogeneous population.

The experimental protocol was divided into three main phases. The first phase was a familiarisation phase, with a presentation of the context and the interface of the micro-world. In this first phase, the subjects also performed a training mission, both with and without cooperative control. The second phase was the experimental phase, composed of two missions, with a time limit of 8 min for each mission. Each mission was performed without and with cooperative control. The order of these two types of missions was changed between the operators to mitigate the learning effect. The last phase of the protocol was the feedback phase, in which the operators filled out questionnaires about their perceptions and acceptance of the cooperative control.

As mentioned above, the operators performed two missions. As shown in Fig. 16, the first mission included 3 different perturbations, which concerned the various autonomous functions: a failure to detect an object, a failure to follow an edge and a failure to track a tree. The second mission included 2 perturbations for which the operator didn’t have any warning. First, the wrong object was followed by the autonomous function in the autonomous mode M3. Second, the robot made a localisation error in the teleoperation mode M0. The second perturbation was a total failure of an autonomous function, which made it impossible to accomplish the current goal.

7.4. Experimental results and discussion

This section presents our results, and our comments, for each resilience criterion.

7.4.1. Efficiency

The first criterion is system efficiency. The efficiency assessment is presented in Table 2. Efficiency is evaluated through the mission time. First, a decrease of the failure rate with respect to the temporal constraints was observed. In fact, 36% of the 30 trials (2 missions for each of the 15 subjects) failed to respect the temporal constraint without cooperative control,
whereas 13% of the 30 trials failed with cooperative control. The inter-operator variability was also reduced by the missions performed with cooperative control. Both the mean time and the standard deviation were reduced in the missions performed with cooperative control. The variation coefficients computed for the different missions show a decrease in the dispersion of mission time for the missions performed with cooperative control.

Student tests for paired samples were carried out on mission times and number of objects detected. The following results were obtained. On the first mission, there is a significant difference between the missions performed with and without cooperative control: \( t(14) = 2.378; p < 0.05 \). On the second mission, there is no significant difference: \( t(14) = 0.464; ns \).

Efficiency can also be evaluated for the observation task using the number of objects detected by the human operator. In this case, it appears that every task in the mission did not experience improved efficiency. According to the Student test for paired samples, the number of detected objects is not significantly different between the missions performed with and without cooperative control: \( t(14) = 1.705; ns \) for the first mission and \( t(14) = 0; ns \) for the second mission.

7.4.2. Adaptability

The analysis of adaptability must be taken into account the fact that the system propositions for managing perturbations were simulated so that all the operators would face similar missions. However, some tendencies can be drawn from the operator reactions, faced with the different solutions for managing the perturbations.

First, the automated system was accepted by the human operators. In reality, 79% of the solutions proposed to cope with a perturbation were accepted by the operators. This result explains the variation coefficients computed in Section 7.4.1. These variation coefficients decreased in the missions performed with cooperative control, which can be explained by most of the operators selecting the default alternative.

Even if inter-operator variability is reduced with cooperative control, as illustrated by the efficiency criterion for the mission time, the system is still able to adapt to the operator modifications. When the operators select another alternative than the default alternative, 77% of them accomplish the mission before the temporal constraint.

7.4.3. Border-line functioning

Border-line functioning is evaluated through the time taken to resolve a perturbation, computed from the time when the perturbation occurred to the time when the operator returned to the initial mission plan. The border-line functioning assessment is presented in Table 3. In the two missions, the mean time and the standard deviation were reduced for each perturbation. The recovery phase corresponds to a phase of system instability that follows each perturbation.

Variation coefficients were computed for the time taken to resolve the perturbations in the two missions. The data dispersion is slightly improved by the cooperative control for the two missions due to the operators selecting the alternatives proposed by the system, as shown in Section 7.4.2.

Student tests for paired samples were carried out on the time taken to resolve the different perturbations. For the first mission, there is no significant difference for the first perturbation: \( t(14) = 1.013; ns \). However, there is significant difference

### Table 2
Assessment of efficiency by the mission time.

<table>
<thead>
<tr>
<th>Mission</th>
<th>Without cooperative control</th>
<th>With cooperative control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean time</td>
<td>00:07:41</td>
<td>00:06:32</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>00:01:28</td>
<td>00:01:02</td>
</tr>
<tr>
<td>Coefficient of variation</td>
<td>0.191</td>
<td>0.158</td>
</tr>
</tbody>
</table>

### Table 3
Assessment of border-line functioning by the time taken to solve perturbations.

<table>
<thead>
<tr>
<th>Mission</th>
<th>Without cooperative control</th>
<th>With cooperative control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mean time</td>
<td>00:01:37</td>
<td>00:01:31</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>00:01:11</td>
<td>00:00:45</td>
</tr>
<tr>
<td>Coefficient of variation</td>
<td>0.732</td>
<td>0.494</td>
</tr>
</tbody>
</table>
for the second perturbation ($t(14) = 2.611; p < 0.05$) and for the third perturbation ($t(14) = 2.422; p < 0.05$). For the second mission, there is no significant difference for the proposed perturbation: $t(14) = 0.314; ns$.

The subjective assessment of the border-line functioning is based on the ability of the operator to manage the observation task. For both missions, the operators perceived the observation task to be more achievable with the cooperative control. In this case, efficiency has to be taken into consideration: for the observation task, efficiency did not show an improvement in the performance of the observation task.

7.4.4. Interaction

Interaction between the agents (i.e., between the operator and the robot) were evaluated through the reaction time after the robot made a request. The interaction assessment is presented in Table 4. The mean reaction time was reduced by the cooperative control in a situation in which the operator is required for the recovery process. This is the case for one of the three perturbations in the first mission. However, the variation coefficient computed for the operator reaction time shows that the dispersion of data is not reduced by cooperative control. Student tests for paired samples were carried out on the reaction time of the operator. There is no significant difference with and without cooperative control: $t(14) = 1.86; ns$.

These results and the lack of significant differences show that information about the recovery process remains necessary. With cooperative control, even though the human operators are now provided with simple information about the perturbation and the proposed solutions, the amount and nature of the information have to be taken into consideration to make the resolution more efficient.

7.4.5. Ranking of the criteria

The operators were asked to fill out questionnaires in order to evaluate their acceptance of cooperative control and their perception of the resilience criteria. On this questionnaire, several questions aimed at obtaining the importance that the operators granted to the different criteria. The following mean ranking was obtained, from the most important to the least important: border-line functioning, interaction, adaptability and efficiency. According to these results, border-line functioning was considered as the most important criterion in cases of perturbations, which corresponds to the importance of the phase of recovery from a perturbation. Border-line functioning was thus considered as the most important resilience criteria in the context of this experimental campaign.

From an objective point of view, the indicator that the operators tried to optimise when a perturbation occurred was the recovery time. A correspondence was thus noticed between the criterion’s perceived importance and the actions of the operators. The alternatives proposed by the system tried to optimise border-line functioning and efficiency of the plans of action, in terms of mission time. Border-line functioning appears to be a relevant criterion in resilient system operations, as it is confirmed both by subjective ranking and objective values.

8. Conclusion and future works

A lack of integration of the human operator was observed in the management of perturbations with respect to the autonomy in human-robot systems. This study is about human-robot systems (i.e., unmanned ground vehicles that can be also remotely controlled by human operators) performing missions of observation or surveillance in a dynamic natural environment. In order to cope with the potential for human errors, technical failures or environmental perturbations, the human-robot system must be able to anticipate, react and recover from these unwanted situations. This study focused on improving system resilience using human-machine cooperation and adjustable autonomy in order to dynamically adapt the system to the different environmental constraints.

Our first contribution was the proposition of a formalism for agent autonomy and its extension to the system’s activity levels. Agent autonomy was defined in terms of three semantic aspects: skills, capacities and authorisations. Three activity levels were defined and each of these levels includes the semantic aspects. Planning and decision-making algorithms were also proposed to implement the decisional and deliberative levels of activity. These algorithms are based on hierarchical methods that take into account the criteria that were defined to characterise the resilience of the human-machine system: efficiency, adaptability, border-line functioning and interaction.

A simulation was run on a micro-world to test the proposed model. The experimental protocol concentrated on missions of observation in a natural environment. Human operators were asked to perform missions both with and without

<table>
<thead>
<tr>
<th>Mission 1</th>
<th>Without cooperative control</th>
<th>With cooperative control</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perturbation 1</td>
<td>Mean time 00:01:05</td>
<td>00:00:36</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>00:00:48</td>
<td>00:00:28</td>
</tr>
<tr>
<td>Coefficient of variation</td>
<td>0.738</td>
<td>0.778</td>
</tr>
</tbody>
</table>
cooperative control. This experimental protocol allowed the model to be implemented in the micro-world and its operation to be evaluated by human operators.

The human operators accepted the cooperative control for managing the perturbations. Their results were analysed using the proposed resilience criteria: efficiency, adaptability, border-line functioning and interaction. These criteria, which were correctly perceived by the operators, allowed the behaviour of the human–robot system to be analysed.

The objective data showed that the effect of cooperative control depends on the nature of the perturbations. Perturbations that lead to a modification of the mission goals are more likely to show no significant difference with and without cooperative control, whereas perturbations that were related to an identified autonomous behaviour show significant differences in resolution time. Perturbations that require the intervention of the human operator to assist the system are also more difficult for the operators to manage. This result may be due to a lack of experience of the operators with the system.

Further research works are now focused on the learning aspect of a resilient process. Learning from errors, both technical and human [29], by introducing uncertainty into the consequences of the plans of action [41] or learning from cooperation are some of the possible ways of capitalising on past experiences. Benefit–Cost-Deficit models [26,41] and cost-sensitive models [18] can be compared in decision-making and learning processes in order to increase relevance of the resilience criteria that are used to characterise the plans of action.

Learning by using genetic algorithms and neural networks [15] can lead to emergent behaviours faced with perturbations. These new behaviours reinforce the creative aspect needed in resilient systems. The defined autonomy modes introduce a form of learning in which both virtual and human agents learn not only how the system functions but also how the other agents reason. Indeed, the autonomy mode M2 defined in the TAROT project allows the human operator to learn how the unmanned system operates by analysing the proposed solutions. Perspectives for future research include the possibility to make the unmanned system learn from the solutions proposed by the human operator by analysing the situation and defining the reason for refusing the solution initially proposed by the system. By imitating the process of reasoning of the human operator, the unmanned system may be able to propose more relevant solutions to the human operator [14].

References

This research was performed by the Human–Machine Systems research team in the LAMIH laboratory at the Université de Valenciennes, in collaboration with the THALES Company. This research was financed by the French Defence Procurement Agency (DGA). This research was also partially financed by the International Campus on Safety and Intermodality in Transportation (CISIT), the European Community, the French Regional Delegation for Research and Technology, the French Ministry of Education, Research and Technology, the Nord/Pas-de-Calais region and the National Center for Scientific Research, the Scientific Research Group on the Supervision, Safety and Security of Complex Systems (GIS 3SGS), and the European Research Group on Human–Machine Systems in Transportation (GDR E HAMASYT). The authors gratefully acknowledge the support of these institutions.

Acknowledgments

This research was performed by the Human–Machine Systems research team in the LAMIH laboratory at the Université de Valenciennes, in collaboration with the THALES Company. This research was financed by the French Defence Procurement Agency (DGA). This research was also partially financed by the International Campus on Safety and Intermodality in Transportation (CISIT), the European Community, the French Regional Delegation for Research and Technology, the French Ministry of Education, Research and Technology, the Nord/Pas-de-Calais region and the National Center for Scientific Research, the Scientific Research Group on the Supervision, Safety and Security of Complex Systems (GIS 3SGS), and the European Research Group on Human–Machine Systems in Transportation (GDR E HAMASYT). The authors gratefully acknowledge the support of these institutions.

References


Stéphane Zieba received a PhD in automation from the Université de Valenciennes in 2009. He now works for the Laboratory for Cognitive Systems Science at the University of Tsukuba. His research concerns the resilience of cooperative human–machine systems in terms of managing perturbations.

Philippe Polet received his MSc and PhD in automation from the Université de Valenciennes in 1998 and 2002, respectively. He works with the Human–Machine Systems (HMS) team at the laboratory LAMIH (Laboratoire d’Automatique, de Mécanique et d’Informatique industrielles et Humaines). His research concerns risk analysis and cooperation in Human–Machine Systems.

Frédéric Vanderhaegen is a professor at the Université de Valenciennes and works with LAMIH. He is the head of the HMS research team. His research concerns human-automation interaction, specifically human error analysis and human reliability.