



AVERTISSEMENT

Ce document est le fruit d'un long travail approuvé par le jury de soutenance et mis à disposition de l'ensemble de la communauté universitaire élargie.

Il est soumis à la propriété intellectuelle de l'auteur. Ceci implique une obligation de citation et de référencement lors de l'utilisation de ce document.

D'autre part, toute contrefaçon, plagiat, reproduction illicite encourt une poursuite pénale.

Contact : ddoc-theses-contact@univ-lorraine.fr

LIENS

Code de la Propriété Intellectuelle. articles L 122. 4

Code de la Propriété Intellectuelle. articles L 335.2- L 335.10

http://www.cfcopies.com/V2/leg/leg_droi.php

<http://www.culture.gouv.fr/culture/infos-pratiques/droits/protection.htm>

Résumé

La demande croissante de systèmes dynamiques sûrs et fiables est devenue un sujet important. Les systèmes de contrôle/commande deviennent de plus en plus complexes et sophistiqués. Par conséquent, les questions de disponibilité, de rentabilité, de fiabilité, de sécurité de fonctionnement et de protection de l'environnement revêtent une importance majeure. La commande tolérante aux pannes est devenue un sujet important étudié d'un point de vue théorique et pratique pour aborder ces points essentiels.

Les conséquences des défaillances peuvent être extrêmement graves en termes de mortalité humaine, d'impact environnemental et de pertes économiques. C'est pourquoi il existe un besoin croissant de contrôle en ligne. Un défaut peut être défini comme un changement inattendu du fonctionnement d'un système, même s'il ne représente pas nécessairement une défaillance ou une panne physique. Pour éviter la détérioration de la production ou l'endommagement des machines, les défauts doivent être détectés afin d'arrêter leur propagation.

Un système de commande tolérant aux fautes est conçu pour maintenir une partie de son intégrité de commande dans le cas d'un ensemble spécifié de fautes possibles ou de changements importants dans les conditions de fonctionnement du système. Une fois qu'une défaillance a été détectée et isolée, la reconfiguration du système défaillant peut être effectuée si le système de commande a intégré un élément de reconfiguration automatique.

Une commande tolérante aux fautes peut ne pas offrir une performance optimale au sens strict pour le fonctionnement normal du système, mais généralement la commande tolérante aux fautes atténue les effets d'un composant défectueux du système sans compromettre complètement la mission ou mettre les utilisateurs en danger. Il est important de souligner que lorsqu'un défaut survient dans un système, que ce soit dans les capteurs ou les actionneurs, les caractéristiques du système entier peuvent subir des changements significatifs, c'est-à-dire une dégradation.

Les défauts peuvent être classés comme suit : défauts de l'installation qui modifient les propriétés dynamiques du système, défauts des capteurs où les dispositifs de mesure du système présentent des erreurs, et défauts des actionneurs qui influencent le régulateur de l'installation par des interruptions ou des modifications (voir la figure 1).

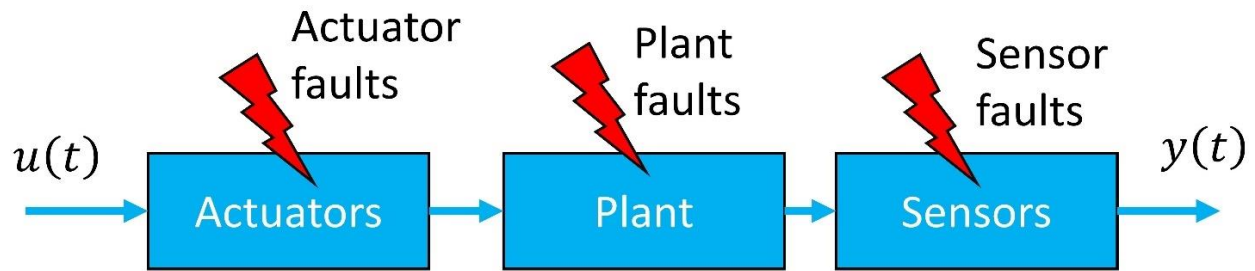


Figure 1 Description générale d'un système défectueux.

Cette thèse présente la conception de différentes méthodes de tolérance aux fautes basées sur le cadre théorique des systèmes multi-agents. Le problème du suivi du leader est considéré dans le but que tous les agents suivent la trajectoire d'un agent leader malgré les fautes.

Les systèmes multi-agents sont composés de systèmes dynamiques individuels communiquant leurs états à travers des réseaux. Un agent individuel possède des actionneurs, des capteurs et un calculateur pour se rapporter à son propre environnement. Chaque agent possède un protocole de communication qui permet l'échange d'informations avec les agents voisins. Ils peuvent échanger de cette manière les informations sur les états actuels. L'échange d'informations entre agents par le biais de réseaux est la principale caractéristique des systèmes multi-agents. Néanmoins, les limitations de bande passante, les retards ou les pertes de paquets sont des défis dans les applications techniques réelles.

L'objectif de la synchronisation des systèmes multi-agents coopératifs est souvent d'atteindre une valeur commune en fonction des états des agents. Comme décrit ci-dessus, un agent est un système dynamique capable de communiquer ses états avec d'autres agents. La plupart des systèmes multi-agents sont des réseaux intelligents, des drones et des réseaux de capteurs, entre autres. Les systèmes multi-agents peuvent être classés comme mobiles ou non-mobiles.

Les systèmes multi-agents du premier ou du second ordre sont donc utilisés pour illustrer les systèmes multi-agents mobiles dans le contexte du rendez-vous, du contrôle de la formation ou du flocking, qui sont les trois principales tâches à accomplir dans les applications coopératives. Ces types de systèmes multi-agents peuvent illustrer le comportement de robots mobiles autonomes, de véhicules aériens sans pilote (UAV), de véhicules sous-marins autonomes, entre autres, dans un espace euclidien.

Les objectifs de la commande dans les systèmes multi-agents sont le consensus et le consensus leader-suiveur. Dans ces deux objectifs de contrôle/commande, les agents doivent atteindre une valeur commune par rapport à leurs états. Le consensus est un accord par rapport aux états des autres agents et dans le consensus de suivi du leader, les agents doivent suivre les trajectoires d'un agent leader. Le consensus est un problème important dans le contexte des systèmes multi-agents où tous les agents parviennent à un accord par rapport aux états des autres agents. Un algorithme de consensus ou un protocole de consensus est une règle d'interaction qui spécifie l'échange d'informations entre un agent et ses voisins sur le réseau. D'autre part, le consensus leader-suivant est un cas particulier obtenu lorsque toutes les trajectoires des agents doivent converger vers la trajectoire d'un leader physique ou virtuel.

La gestion adéquate des ressources est un défi à prendre en compte lors de la mise en œuvre de systèmes de contrôle dans des applications réelles. La plupart des systèmes autonomes disposent de sources d'énergie indépendantes basées sur des batteries, qui, dans certains cas, sont insuffisantes pour effectuer de longues opérations dans le temps. Ces systèmes autonomes possèdent des microprocesseurs intégrés qui sont utilisés pour gérer des composants tels que des capteurs et des actionneurs. En raison du développement rapide des technologies informatiques, les systèmes à temps discret prennent une importance croissante. Une question cruciale est de savoir comment choisir un temps d'échantillonnage approprié.

Le microprocesseur échantillonne les capteurs à chaque instant discret et calcule la loi de commande afin de manipuler le système. La commande échantillonnée périodiquement à chaque période d'horloge "est appelée commande déclenchée par le temps. La commande déclenchée par événement est une méthodologie de commande souvent utilisée dans les systèmes multi-agents pour réduire la charge de communication d'un réseau numérique et le taux de mise à jour de la commande. Dans la commande déclenchée par événement, le contrôleur n'est mis à jour que si cela est nécessaire. Par conséquent, la consommation d'énergie de chaque agent est réduite, et la limitation de la bande passante est bien gérée. Il convient de noter que les stratégies de commande événementielle et la commande tolérante aux pannes pour les systèmes multi-agents soumis à des pannes d'actionneur, de capteur ou de communication ont été récemment explorées, comme dans l'article où deux stratégies de commande tolérante aux pannes sont utilisées pour les systèmes

multi-agents soumis à des pannes d'actionneur, de capteur et à des perturbations inconnues à l'aide de réseaux neuronaux adaptatifs et d'observateurs non linéaires.

Les principaux composants d'une boucle de contrôle/commande basée sur les événements sont le système lui-même, le contrôleur (à temps continu ou à temps discret) et un générateur d'événements. Les liaisons de communication ne sont utilisées qu'après l'apparition d'un événement au moment t_k et la sortie actuelle $y(t_k)$ (ou l'état $x(t_k)$) est transmise au contrôleur. Le contrôleur détermine la nouvelle entrée u_k , qui est conditionnée par le générateur d'entrée de commande pour déterminer l'entrée continue $u(t)$ dans l'intervalle de temps $[t_k, t_{k+1}]$ jusqu'à ce que le prochain événement se produise au moment t_{k+1} .

En raison de la complexité accrue et du nombre croissant de composants, les systèmes multi-agents sont particulièrement sensibles aux défauts et aux perturbations, qui peuvent se produire avec une probabilité plus élevée et entraîner une dégradation des performances ou une panne de tous les agents. Les performances du consensus sont souvent affectées par des perturbations externes. La commande tolérante aux fautes des systèmes multi-agents a attiré l'intérêt des chercheurs ces dernières années.

Les perturbations et les incertitudes du modèle modifient le comportement de l'installation comme des défauts. Du point de vue de la commande tolérante aux fautes, lorsque les fautes sont représentées comme des signaux externes supplémentaires, elles sont appelées fautes additives. Lorsque les défauts sont modélisés comme des déviations de paramètres, ils sont appelés défauts multiplicatifs car le paramètre du système dépendant de la taille du défaut est multiplié par le dysfonctionnement. Au départ, les perturbations et les incertitudes du modèle ont des effets similaires sur le système. En revanche, les fautes sont des éléments inconnus qui doivent être détectés et éliminés. Les perturbations et les incertitudes du modèle sont connues, et elles sont traitées par un filtrage approprié ou une conception de commande robuste.

Les principales contributions de cette thèse se concentrent sur les stratégies de contrôle pour différentes entrées externes inconnues considérées comme des défauts et/ou des perturbations :

- 1) La conception d'un mécanisme déclenché par des événements pour résoudre le problème de contrôle/commande leader-suiveur en réduisant l'échange d'informations entre les agents et le taux de mise à jour du contrôle.

2) La conception d'une commande tolérante aux pannes basée sur des actionneurs virtuels dans les systèmes multi-agents soumis à des pannes d'actionneurs.

3) La conception d'un contrôle/commande de formation déclenché par événement pour les systèmes multi-agents robuste aux défauts de communication.

4) La conception d'une commande leader-suiveur à limite quadratique dans les systèmes multi-agents soumis à des perturbations limitées.

La performance et l'efficacité des stratégies proposées sont démontrées par des exemples numériques et par leur mise en œuvre dans une flotte expérimentale de véhicules aériens sans pilote.

Mots-clés : Contrôle tolérant aux fautes, consensus leader-suiveur, système multi-agents, fautes de communication, consensus à limite quadratique, actionneurs virtuels, contrôle de formation, drones.

Abstract

Increasing demand for safe and reliable dynamic systems has been becoming an important subject. Modern control systems are becoming more complex and sophisticated, in consequence, the issues of availability, cost efficiency, reliability, operating safety, and environmental protection are of major importance. Fault-tolerant control has become an important subject in modern control theory and practice.

The consequences of faults can be extremely serious in terms of human mortality, environmental impact, and economic loss. Therefore, there is a growing need for on-line control. A fault can be defined as an unexpected change of system function although it may not represent physical failure or breakdown. To avoid production deterioration or machine damage, faults must be found in order to stop the propagation.

A fault-tolerant control system is designed to maintain some portion of its control integrity in the case of a specified set of possible faults or large changes in the system operating conditions. Once a fault has been detected and isolated, the reconfiguration of the faulty system can be done if the control system has built-in an element of automatic reconfiguration.

A fault-tolerant control may not offer an optimal performance in a strict sense for normal system operation, but generally the fault-tolerant control mitigates effects of faulty system component without completely compromising the mission or putting the users at risks. It is important to emphasize that when a fault occurs in a system, either in sensors or actuators, the characteristics of the entire system can submit to significant changes, i.e., degradation.

The faults can be classified as plant faults which change the dynamical properties of the system, sensor faults where the measuring devices of the system have errors, and actuator faults which influence the plant controller by interruptions or modifications (see Figure 1).

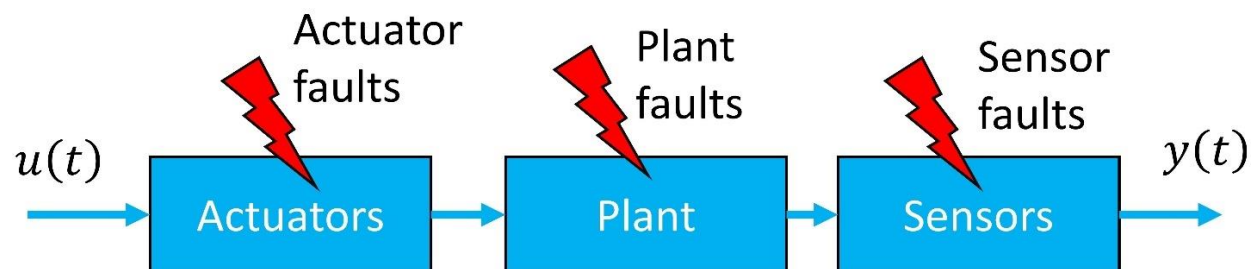


Figure 2 General description of a faulty system

This thesis presents the design of different fault-tolerant methods based on multi-agent system theoretical framework. The leader-following problem is considered with the aim that all agents follow the trajectory of a leader agent despite faults.

Multi-agent systems are composed by individual dynamic systems that communicate each other their states through networks. An individual agent has actuators, sensors, and a computer for relating to its own environment. Each agent has a communication protocol which allows the exchange of information with neighboring agents. They can exchange in this way the information of the current states. The exchange of information between agents through networks is the main characteristic of multi-agent systems. Nevertheless, bandwidth limitations, delays, or packet losses are challenges in real engineering applications.

The objective of synchronization of cooperative multi-agent systems is often to reach a common value in relation to the states of the agents. As described above, an agent is a dynamic system able to communicate its states with other agents. Most of the multi-agent systems are smart grids, drones, and sensor networks, among others. Multi-agent systems can be classified as mobile or non-mobile.

Therefore first-order or second-order multi-agent systems are used to illustrate mobile multi-agent systems in the context of rendezvous, formation control, or flocking which are the three main tasks to tackle in cooperative applications. These types of multi-agent systems can exemplify the behavior of autonomous mobile robots, unmanned aerial vehicles (UAVs), autonomous underwater vehicles, among others, in a Euclidean space.

The control objective in multi-agent systems are consensus and leader-following consensus. In both control objectives, the agents must achieve a common value in relation to their states. Consensus is an agreement in relation to the states of the other agents and in leader-following consensus, the agents must follow the trajectories of a leader agent. Consensus is an important problem in the context of multi-agent systems where all the agents reach an agreement in relation to the states of the other agents. A consensus algorithm or consensus protocol is an interaction rule that specifies the exchange of information between an agent and its neighbors on the network. On

the other hand, leader-following consensus is a particular case obtained when all the trajectories of the agents must converge to the trajectory of a physical or virtual leader.

Proper resource management is a challenge to be considered when implementing control systems in real applications. Most autonomous systems have independent battery-based power sources, which in some cases are insufficient to carry out long operations over time. These autonomous systems have embedded microprocessors that are used to manage components such as sensors and actuators. As a result of the rapid development of computer technologies, discrete-time systems are becoming increasingly important. One crucial issue is how to select an appropriate sample time.

The microprocessor samples every discrete time the sensors and compute the control law in order to manipulate the system. The control which is periodically sampled each time is called time-triggered control. Event-triggered control is a control methodology often used in multi-agent systems for reducing the communication load of a digital network and the control update rate. In event-triggered control, the controller is not updated unless it is required. Consequently, the energy consumption in each agent is reduced, and the bandwidth limitation is well managed. It should be noticed that event-triggered control strategies and fault-tolerant control for multi-agent systems under actuator, sensor, or communication faults have been recently explored such as in where two fault-tolerant control strategies are used for multi-agent systems under actuator, sensor, and unknown disturbances using adaptive neural networks and nonlinear observers.

The main components of an event-based control loop consist of a plant, the controller (continuous-time or discrete time), and an event generator. The communication links are only used after the occurrence of an event at time t_k and the current output $y(t_k)$ (or state $x(t_k)$) is transmitted to the controller. The controller determines the new input u_k , which is conditioned by the control input generator to determine the continuous input $u(t)$ in the time interval $[t_k, t_{k+1})$ until the next event occurs at time t_{k+1} .

Due to the higher complexity and the increasing number of components, multi-agent systems are particularly sensitive to faults and disturbances, which can happen with a higher probability and result in performance degradation or breakdown of all the agents. The performance of the consensus is often affected by external disturbances. Fault-tolerant control of multi-agent systems has attracted the researchers' interest in recent years.

Disturbances and model uncertainties change the plant behavior like faults. From the fault-tolerant control point of view, when faults are represented as additional external signals, they are called additive faults. When the faults are modeled as parameter deviations, they are called multiplicative faults because the system parameter depending on the fault size are multiplied with the malfunction. At the beginning, disturbances and model uncertainties have similar effects on the system. In contrast, faults are unknown elements which must be detected and removed. Disturbances and model uncertainties are known, and they are handled by appropriate filtering or robust control design.

The main contributions of this thesis are focused on control strategies for different external unknown inputs considered as faults and/or disturbances:

- 1) The design of an event-triggered mechanism to solve the leader-following control problem reducing the exchange of information between agents and the control update rate.
- 2) The design of a fault-tolerant control based on virtual actuators in multi-agent systems subject to actuators faults.
- 3) The design of an event-triggered formation control for multi-agent systems with communication faults.
- 4) The design of a quadratic boundedness leader-following control in multi-agent systems subject to bounded disturbances.

The performance and effectiveness of the proposed strategies are shown through numerical examples and implementation in an experimental fleet of unmanned aerial vehicles.

Keywords: Fault-tolerant control, multi agent system, leader-following consensus, communication faults, quadratic boundedness consensus, virtual actuators, formation control, UAVs.