Journal of Intelligent & Robotic Systems (2018) 91:59–83
https://doi.org/10.1007/s10846-018-0881-x

Resilient Design and Operation of Cyber Physical Systems with Emphasis on Unmanned Autonomous Systems

George Vachtsevanos[1] · Benjamin Lee[1] · Sehwan Oh[1] · Michael Balchanos[1]

Received: 7 March 2018 / Accepted: 23 May 2018 / Published online: 14 June 2018

Abstract

Autonomy and autonomous systems are occupying central stage in the research community, as autonomous vehicles are proliferating and their utility in all aspects of the military and civilian domains are increasing exponentially from one year to the next. The development and application of resiliency and safety technologies to autonomous systems is, unfortunately, not keeping pace with their growth rate. Several factors impede the deployment and adoption of autonomous systems. Among them is the absence of an adequately high level of autonomy that can be relied upon, significant challenges in the area of human-machine interface requiring significant human intervention to operate and interpret sensor data, the need for emerging machine learning technologies and, most importantly, the resilient design and operation of complex systems to assure their safety, reliability and availability when executing missions in unstructured and cluttered environments. Recent advances in resiliency and safety of complex engineered systems have focused on methods/tools to tradeoff system performance for increased time to failure aiming at mission completion or trial and error methods to arrive at a suboptimal policy for system self-organization in the presence of a failure mode. This paper introduces a novel framework for the resilient design and operation of such complex systems via self-organization and control reconfiguration strategies that avoid empirical trial and error techniques and may be implemented and perform in real time on-platform. The main theme is summarized as: "a healthy and resilient system is a safe system". To accomplish this objective, we introduce an integrated and rigorous approach to resilient design while safety considerations ascertain that the targeted system is contained within a safe envelope. A resilient system is robustly and flexibly monitoring its internal and external environment, it can detect and anticipate disturbances that may affect its operational integrity and take appropriate action to compensate for the disturbance. Resilience enhances safety while improving risk factors and assures that vehicles subjected to extreme disturbances remain within their safe envelope. The enabling technologies begin with graph spectral and epidemic spreading modeling tools to represent the system behaviors under normal and faulty conditions; a Markov Decision Process is the basic self-organization module. We are introducing a novel approach to fault-tolerance by considering the impacts of severe fault modes on system performance as inputs to a Reinforcement Learning (RL) strategy that trades off system performance with control activity in order to extend the Remaining Useful Life (RUL) of the unmanned system. Performance metrics are defined and assist in the algorithmic developments and their validation. We pursue an integrated and verifiable methodology to safety assurance that enables the evaluation of the effectiveness of risk management strategies. Several unmanned autonomous systems are used for demonstration purposes.

Keywords Resilience · Self-organization · Reconfigurable control · Complex adaptive systems

1 Introduction/Motivation

A cyber-physical system (CPS) is an integration of computational with physical processes. Modern CPSs are increasingly complex, combining hardware/software with human intervention, interfacing commands and assessment strategies with decision-making. The CPS concept map

✉ George Vachtsevanos
gjv@ece.gatech.edu

Sehwan Oh
soh48@gatech.edu

[1] The Georgia Institute of Technology, Atlanta, GA 30332, USA

includes a large spectrum of physical/engineering systems from a "smart" campus to unmanned autonomous systems, swarms of autonomous systems, manufacturing processes, among others. Unmanned robotic platforms (UAVs, UUVs, UGVs) operating as a single vehicle or in swarm formation, are complex CPSs. UAVs place significant demands to human (sensor, pilot) operators who are required to make informed decisions in (almost) hard real-time, and require significant computational resources for data/information processing. It was suggested by an Autonomous Vehicle Operator (AVO) that, at times, "he's been more overcome by the torrent of information pouring in during a drone flight than he was in the cockpit". Currently, limitations in autonomy lead to operator work time exceeding the time of unmanned system deployment and gains in the field of autonomy are required to reverse the current trend. The final report of the Defense Science Board Summer Study on Autonomy, June 2016, provides recommendations for "accelerating DoD's adoption of autonomous capabilities" [1].

Cyber physical systems are designed to perform tasks/missions under nominal conditions – absence of internal/external severe disturbances. It is documented though that a significant percentage of Class A air mishaps are attributed to Unmanned Aerial Vehicles (UAVs). There is a need to design and operate unmanned systems capable of withstanding severe disturbances that may endanger the integrity of the vehicle. The proposed framework is founded on rigorous and verifiable technologies for endowing UAVs (and other unmanned systems/cyber physical systems) with capabilities that go beyond the "normal" operating regime and posses the ability to perform missions in the presense of extreme hazardous environments.

Achieving these gains will require developing new and innovative methods and tools to endow complex unmanned systems with attributes of resilience and safety assurance, risk assessment and management enabling them to operate across a range of functional capabilities. Resilient design and operation of UAS is the first step towards extending the system's operational envelope in the presence of extreme disturbances and assuring that safety margins are adhered to. The link between resilience and safety is a natural one with the former contributing to the latter's capability to maintain stability requirements. Learning strategies are contributing to a dynamic updating of design for resilience and safety of UAS algorithms ascertaining that "smart" knowledge bases are kept current, data are interpreted correctly and accurate decisions are made to support the operator. Multiple learning tools/methods are called upon depending on the case at hand.

This paper proposes the development and evaluation of a holistic, rigorous and verifiable framework for the resilient design and operation of high-confidence engineering systems. The proposed formalism aims to provide an understanding of complex system behaviors and the means for robust design and operation of such complex systems. The emphasis is in capitalizing on fundamentals of Complexity Theory for the resilient design of cyber physical systems (CPSs), with a focus on autonomous unmanned systems as the application domain. The foundations of the proposed design for resilience build upon lessons learned from early successes/failures of the interplay between life sciences and complex engineered systems, and rely upon characteristic attributes of the biological world such as *immunity* and *self-healing* to withstand and absorb severe disturbances. It builds upon studies conducted by an international group of researchers - led by two noted ecologists, Lance Gunderson and C. S. Holling - who investigated extensively the concept of resilience and its applicability to ecological, social, and management systems [2, 3].

The paper is organized as follows: Section 2 introduces the fundamental concepts of resilience and its application to engineering problems. The application of resilience is formulated as a novel approach to safety assurance in autonomy. Section 3 states the overall framework for resilience and safety of UAS, along with the state of the art enabling technologies. Section 4 proposes the resilient design and operation of autonomous systems with proper evaluation criteria for performance verification. The experimental platforms and results are also presented afterwards. Section 5 presents the safety assurance techniques in hostile environments, including risk assessment, evaluation, and control. A Conclusion section then finalizes the discussion of this paper (Fig. 1).

2 Resilient Systems and Resilience Engineering – Fundamental Concepts

Resilience engineering is an emerging discipline, which can be viewed as an evolution to traditional safety and survivability engineering practices. It brings a new perspective in understanding and analyzing system uncertainty, risk,

Fig. 1 MQ-1 predator control station (pilot and sensor) courtesy of general atomics

and furthermore assessing safety and survivability. It has been established under a set of premises, which stem from limitations in understanding of how risk and uncertainty affect safety, or how system complexity may lead to accidents. Additional issues about complex system interactions in large scale operating environments contribute to overall uncertainty and nonlinear, dynamic system behavior. Hollnagel has summarized the basic premises of resilience engineering, in the following statements [4]:

1. Performance conditions are always underspecified.
2. Adverse events can be attributed to an unexpected combination of normal performance variability.
3. Safety management cannot be based on hindsight nor solely rely on error tabulation and failure probability calculations

These statements reflect limitations in current safety engineering practice. They are the fundamental directions, which resilience engineering has been addressing as continuously evolving, and emerging discipline.

2.1 Overview of System Resilience

A *resilient system* can adjust its functioning prior to or following changes and disturbances so that it can go on working even after a major mishap or in the presence of continuous stress, mainly by being able to be proactive on safety [4]. The ability to be proactive is emphasizing the reduction in system susceptibility, by either preventing unwanted events and outcomes or eliminating hazards in the operating environment. Hollnagel has provided a template of the fundamental functionality that resilient systems must possess, shown in Fig. 2 [34]:

Anticipate disturbances. That includes potential threats, various disruptions and any other destabilizing conditions. Implementing this function relies on what model is selected to predict the future and under what tolerance for risk.

Monitor performance. Except for mission performance and system health monitoring, a resilient system must also be able to monitor risks and threats and continuously revise its own model of risk identification. This will allow for revealing of non-profound transient effects, that even though are not permanent, they can still contribute to failures and accidents.

Respond to threats. This function implies an intrinsic readiness, along with an inherent flexibility and adaptability in response to regular, irregular or unexpected, and unexampled events.

We pursue in this paper an integrated framework for resilient design and operation of complex unmanned systems that begins with monitoring, modeling and understanding the potential impact of threats/disturbances on the operational integrity of such critical assets followed

Fig. 2 The three basic functions of a resilient system

by a proactive approach to threat response via appropriate self-organization and control reconfiguration strategies, in difference to actions intrinsically generated.

2.2 Autonomy, Assurance and Risk: Challenges and Enabling Technologies

Several factors impede the deployment and adoption of autonomous systems:

1. In the absence of an adequately high level of autonomy that can be relied upon, substantial operator involvement is required, creating significant new challenges in the areas of human-machine interface and mixed initiative control.
2. Achieving higher levels of autonomy in uncertain, unstructured, hazardous and dynamic environments involves data-driven machine learning techniques with many open systems science and systems engineering challenges.
3. Machine learning techniques widely used today are inherently unpredictable and lack the necessary mathematical framework to provide guarantees on correctness, while DoD and industrial applications that depend on safe and correct operation for mission success require predictable behavior and strong assurance.
4. Unmanned systems operate now in uncertain and noisy environments subjected to hazards/dangers that endanger their operational integrity. New methods must be developed and implemented to impart attributes of resilience and safety to these critical asserts.

A few decades ago when academic institutions and government agencies began developing unmanned system technologies and testing UAVs, it was customary to perform field tests under almost "perfect"weather conditions. The rapid proliferation of UAVs and the ever expanding application domain for military and civilian missions are requiring that such autonomous systems must fly under "all weather" conditions. Achieving these objectives, while avoiding class A mishaps, requires the development of new technologies imparting on these complex systems properties of resilience to extreme hazards/disturbances (internal and external), means to assure that UAVs are maintaining their operational integrity even when subjected to fault/failure modes, and remain within a safe envelope meeting stability bounds.

2.3 The Concept of Autonomy

Autonomy, in the context of an unmanned autonomous system, is the capability of its components/systems to operate independently from external control. For military and civilian missions there is a spectrum of autonomy in a system from basic automation (mechanistic execution of action or response to stimuli) through partial autonomy, flexible autonomy and fully autonomous systems able to act independently in dynamic and uncertain environments.

An autonomous system must be capable of:

- Monitoring its internal and external environment and its own performance; establishing a harmonious interface with human operators
- Detecting, isolating and identifying incipient failure modes and extreme external disturbances
- Predicting the remaining useful life of failing components
- Taking appropriate corrective action to safeguard the integrity of the vehicle for the duration of the contingency

The integration of different hardware/software resources to provide a consistent management function under internal and external stresses have not yet fully been accomplished. We propose to address these challenges through the introduction of robust, rigorous and verifiable resiliency technologies. Our specific objective is to increase the level of autonomy in adapting the control of critical systems to respond to online estimates of current fault states and predictions of future fault growth. We apply a rigorous and analytical approach to developing, testing and evaluating novel enabling technologies to improve / enhance automated fault/failure adaptation methods. Learning strategies are contributing to hazard/threat detection, identification and prediction of their potential impact on UAS safety and resiliency. Reinforcement learning strategies are called upon to enhance and update the cases in the DCBR case library. Deep reinforcement learning is a valuable tool for feature extraction and selection as well as for "best" control actions.

3 The Framework for Resilience and Safety of UAS/State of the Art

Resilience is a key driver in the design of systems that are subjected to severe fault/failure modes or external disturbances (wind gusts for an unmanned aerial vehicle). Resilience is a measure of the persistence of systems and of their ability to absorb change and disturbance and still maintain the same relationships between state variables [2]. A resilient UAS monitors its own performance, it can anticipate pressures/disruptions and responds to irregular or regular threats/hazards. Safety management for engineering systems is proactive not reactive. Resilience enhances the system's robustness, monitors, revises risk models, and uses resources (software based) proactively in the face of

disruptions/hazards. Resilience is a foundational technology for safety. Safety concerns for UAS executing extreme maneuvers in unstructured environments raises issues of assurance and risk assessment/management. Risk reduction via resilient design and operation, involves protection against unwanted outcomes and reduces detrimental consequences.

We introduce a thorough and proactive methodology to resilient design and operation of autonomous systems for improved safety and risk assurance/management that entails the following major steps:

1. Threat/hazard characterization - detection, identification, prediction of hazard evolution; performance metrics are specified and used in the development and validation process.
2. Self-organization strategy for systems subjected to severe disturbances (internal or external) with performance metrics/stability conditions
3. Reconfigurable control strategy for systems subjected to severe fault modes
4. Safety assurance and risk assessment/management methodology

Safety Risk Management provides a workflow for a formal process to describe the system, identify hazards, assess risk, and control/minimize risk. We define possible hazard scenarios, quantify their frequency of occurrence and estimate/predict their consequences. Once the hazard analysis tasks are set in probabilistic terms, the design for risk assessment takes over and addresses quantitative and qualitative risk factors. Risk assessment requires an adequate representation/model of the future events. A prognostic method is pursued to accomplish this task while representing and managing the inherent uncertainty in prognosis. Safety Assurance enables the evaluation of the effectiveness of risk management strategies and ensures compliance with oversight entities. Risk reduction via resilient design and operation, involves protection against unwanted outcomes and reduces detrimental consequences.

Design for resilience builds on concepts of self-organization and control reconfiguration for cyber physical systems. We introduce a self-organizing strategy in the form of a Markov Decision Process (MDP) with dynamic programming for optimal performance. A system is considered *organized* if it has certain structure and functionality, and self-organization implies that the organization of the system occurs internally, without any external or centralized control unit [6]. In the simplest case, a self-organization strategy consists of two components: response and adaptation, responding to the system's functionality. Along with a reduced computational burden due to the targeted operation, a self-organization method provides the benefit of random noise adaptation, since the process is spontaneous with intrinsic update rules [7]. an optimal control approach with Reinforcement Learning (RL), Differential Dynamic Programming (DDP) and Model Predictive Control (MPC) is considered as a means for control authority redistribution and reconfiguration when the targeted system is subjected to severe threats/hazards. Prognostic knowledge is incorporated in a quadratic cost function of the optimal control problem as a soft constraint, thus providing a link between health management and reconfigurable control. Success criteria are founded on Lyapunov stability conditions setting the stage for a rigorous approach to verifying the efficacy of the proposed methodology and allowing for comparisons with robust and other classical control methods.

There is a rich literature describing proper adjustments to control actions that assure resilient behaviors. Fault Tolerant Control System (FTCS), motivated by commercial aircraft accidents [8], has been researched extensively. Clements developed a hierarchical control architecture showing the interconnections among fault detection & identification, set-point controller, control redistribution, control gain adaptation, and component restructuring [9]. Ge, Kacprzynski, Roemer, and Vachtsevanos introduced a higher-level adaptive system framework using an Automated Contingency Management (ACM) concept [10]. Drozeski, Saha, and Vachtsevanos proposed a three-tier hierarchical control scheme as Active FTCS [11]. Tang, Kacprzynski, Goebel, Saxena, Saha, and Vachtsevanos extended the ACM framework by integrating it with a prognostics module [12]. Brown, Georgoulas, Bole, Pei, Orchard, Tang, Saha, Saxena, Goebel, and Vachtsevanos proposed prognostics enhanced low-level reconfigurable control for an avionics component [13]. Bole, Tang, Goebel, and Vachtsevanos described a fault adaptive control architecture, and Bole addressed uncertainties in prognostics and reconfigurable control allocation strategies [14, 15]. We introduce in this paper a novel framework for resilient design and operation of UAS realized in terms of two complementary strategies: Self-organization and control reconfiguration. Resilience contributes to system safety properties by assuring that the system is capable of mitigating large-scale disturbances. Learning strategies are an essential component of the proposed architecture borrowing from the cognitive engineering domain. A reasoning paradigm, called Dynamic Case Based Reasoning, entails attributes of learning and adaptation and constitutes the "smart" knowledge base for decision support. The human is always in the loop receiving

Fig. 3 Media photos of UAV incidents

the actionable information gathered by the resilience/safety modules to arrive at "best" decisions.

3.1 UAV Failure Modes

"Drone lands on the White House lawn: January 2015—this one made national news. A Defense Department employee in Washington, D.C., crash-landed his personal drone on the White House lawn. He was, he admitted, intoxicated" [16].

The UAV numbers and categories have been increasing exponentially worldwide over the recent past but their reliability, resilience and sustainability have not kept pace with their growth motivating research and development efforts to improve their operational integrity and safety. It has been reported that more than 40% of all Class A air mishaps are attributed to UAVs (Fig. 3).

3.2 Disturbance/Hazard Analysis

Disturbance/hazard analysis considers the impact of varied operational and environmental disturbance factors on system safety. A degrading system affected by hazards can be expressed as

$$L(t+1) = f(L(t), U,) + \omega_1(t+1) \tag{1}$$

$$U = g(\sigma, t) + \omega_2 \tag{2}$$

$$y(t+1) = h(L(t)) + \omega_3(t+1), \tag{3}$$

where $L(t)$ denotes a time-varying life state, for example, a life degradation or loss condition; $\sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_n$ is a series of environmental and operational disturbance/hazard factors that affect the time evolution of system degradation, such as fault/failure modes; $U$is the disturbance factor severity function; y represents system output; $\omega_{1,2,3}$ are noises. In real applications, the degradation model f(·) usually assumes a nonlinear function, has non-monotonic attributes (for example, recovery effects), corrupted by non-Gaussian noise, and is appropriately represented by pdfs to indicate the underlying uncertainty. We use a particle filtering formulation for the solution of the stochastic equations listed above, as detailed in the sequel.

3.3 Hazard/Threat/Dissturbance Characterization, Detection and Prediction

Hardware, software, the environment, and human factors are major sources of hazards. For the CPS under consideration, we seek historical hazard data and categorize them as to their severity, frequency of occurrence, and testability. It is, of course, true that "you can only**manage** what you can**measure"** and data/information regarding hazards and their potential impact on system safety are absolute requirements to modeling, representation and control of hazards and safety margins, as detailed in the sequel.

3.4 Prognostics and Health Management Technologies

The foundation for the development and application of PHM technologies is a thorough understanding of the physics of failure mechanisms, as critical systems are subjected to disturbances/stress/usage patterns. From the physical components/systems themselves to a good understanding of how such systems fail and under what conditions leads to optimum Condition Indicator (CI) extraction and selection and, eventually, to accurate diagnostics and prognostics.

We introduced and will take advantage of an integrated and rigorous architecture for health management of critical assets [17]. The on-line modules of the architecture begin with pre-processing of raw data in order to reduce the data dimensionality and improve the Signal to Noise Ratio (SNR). Typical pre-processing routines include data compression and filtering, de-noising, FFTs, among others [18]. Data mining is a crucial step in the PHM process. *Deep Learned Features (DLF)* and *Deep Reinforcement Learning* produce optimum results for Condition Indicator (CI) extraction and selection. We pursue two fundamental

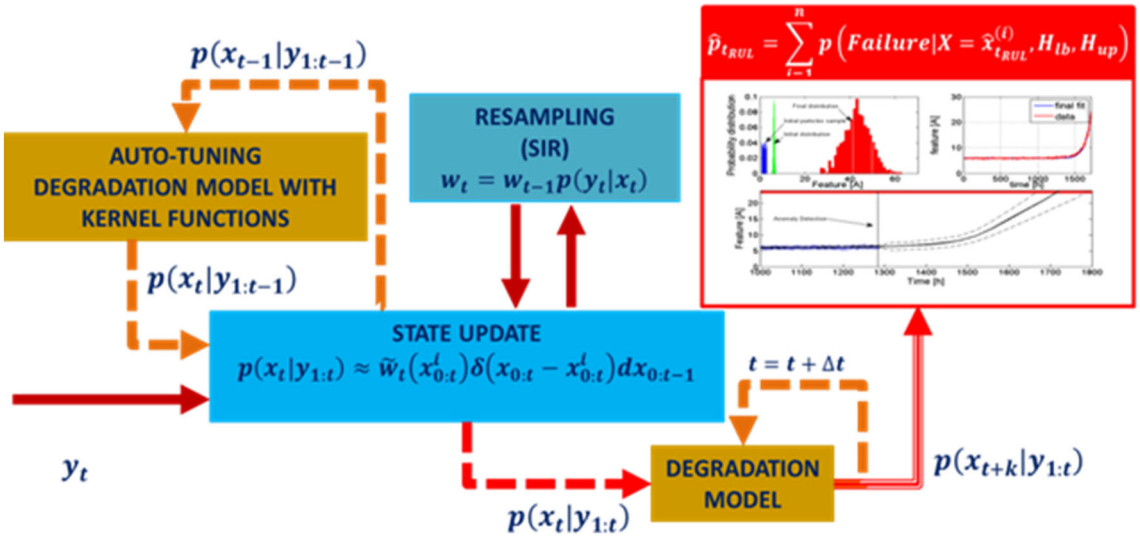


Fig. 4 The prognostic architecture using particle filtering

approaches to the data mining problem: A data mining formalism in order to determine the ”best” features that are descriptive of the faulty behavior of critical UAV components/subsystems using Kolmogorov Complexity to process large volumes of data, i.e. perform such functions as data compression, clustering, classification, anomaly detection, forecasting. We pursue a novel approach for feature extraction that builds upon concepts of Deep Learning [19–21]. Our goal is a Deep-Learned Features (DLF) framework that automatically learns neural features from data, to enhance the engineered feature library. The methodology to derive an optimum feature vector is *hierarchically* represented and *automatically* learned, as happens in natural biological vision systems (LeCun cat's visual cortex). Additionally, DP methods are used for enhanced situational awareness and decision support. Advantages of this approach include: (1) ability to learn from data even if a large fraction is *unlabeled* (no ground-truth labels/classes of corrosion available); (2) *scalability* to very large problems (e.g., 1000-class recognition, using GPUs if needed); (3) hierarchical representations in which some neurons may respond selectively to particular localized problems, as well as more global, proto-detector types of features to include in our library. We begin with rigorous and verifiable methods to detect, identify and isolate fault/failure modes. We then predict their time evolution. The enabling technologies borrow from physics of failure mechanisms for critical component/subsystem modeling, feature or condition indicator selection based on data mining, incipient failure diagnosis and prognosis algorithms, and fault-tolerance methods using real-time prognostic estimates and Model Predictive Control (MPC) routines to assure system stability and survivability. Prediction of Remaining Useful Life is a major component of the framework for hazard analysis and accurate prediction of a failing component's remaining useful life.. Risk assessment is based and quantified on prognostic information. These concepts are detailed and quantifiable metrics suggested in the sequel. Root causes are determined by tracing back from the (failing) component to the level using a fault tree analysis technique. Threats/hazards in engineering systems (unmanned autonomous vehicles) are characterized by their probability of failure. The analysis/design methods and variables/parameters are all probabilistic notions expressed as PDFs.

Failure Prognosis [22–24] Input from the diagnostic block is combined with stress profiles and feeds into the fault growth model. An estimation method (in this case particle filtering) is called upon to propagate the fault model initially one step at a time while model parameters are updated on-line in real-time as new sensor data become available. Eventually, the model is allowed to perform long-term prognosis with confidence bounds. The fault model PDF is convolved with the hazard zone PDF when the former reaches the threshold bounds and the resultant PDF is projected along the time axis (which is usually measured in “cycles” of operation) depicting the system's remaining life statistics. Prognosis is achieved by performing two sequential steps, prediction and filtering or update. Figure 4 depicts the particle filtering module including the essential steps of degradation modeling, resampling, auto-tuning and state update.

Figure 5 illustrates the predicted fault growth of a system where a fault is detected at time t_{detect} and a prediction of the RUL is made at time $t_{prognosis}$. When

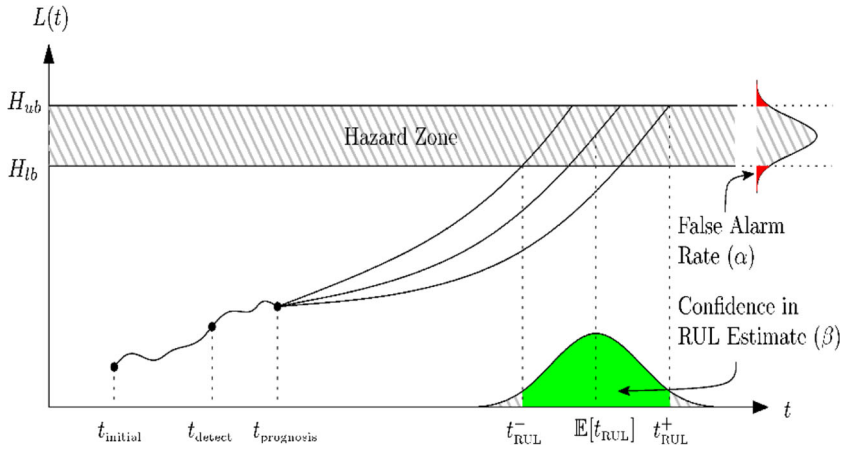


Fig. 5 Illustration of long-term prediction

the hazard is detected and its time evolution predicted, control reconfiguration and self-organization strategies are called upon to ascertain that the targeted unmanned system maintains its safe operation.

Small UAVs in swarm formation are tasked to execute surveillance and reconnaissance missions in a cluttered and uncertain environment. The UAVs are optimally located for maximum area coverage and a small number of them in the perimeter of the swarm are equipped with instruments to detect approaching wind gusts. A vertical wind gust is modeled as a power spectral density, in Dryden form [25, 26]. The formation control problem is addressed via cooperative game concepts. The swarm self-organizes as it moves from one designated area to the next. We take advantage of a *Markov Decision Process* (MDP) to enable the computation of the optimal value function, optimal policy, number of iterations taken, etc. Intra-UAV interactions are modeled as a strongly connected graph consisting of a number of nodes (UAVs) and edges that represent inter-agent information exchange links. Success criteria are formulated as Lyapunov functions. The compensation mechanism is implemented as a behavioral/reactive control problem realized as a neuro-fuzzy paradigm, thus accounting for uncertainty conditions. Failure conditions are compensated via re-allocation of tasks between healthy UAVs and spatially redistributing them to achieve set objectives.

3.5 Integrity Management Metrics – Risk, Confidence and Trust Consensus

We introduce quantifiable metrics for such integrity management attributes as risk, confidence and trust consensus, providing assurance that autonomy attributes are achieved. In summary, we:

Estimate trust consensus (an attribute of integrity management) from contributing sources of data/information (sensors, controllers, classifiers, etc.) [27]. For guaranteed performance, and in the presence of adverse conditions (fault modes, large-grain uncertainty, etc.), a trust consensus must be reached by the decision support system that determines which sources of data/information (sensors) to trust, which to disregard, and which to avoid. We propose to develop a framework for trust propagation and maintenance that yields global consensus of trust under rich enough sensor structure graphs. We leverage pioneering work conducted in the area of supervisory control of mobile sensor networks, material handling processes and communication networks [28–30].

Determine confidence (another integrity management attribute) in decision outcomes, control output, prognostics, etc.

Evaluate risk (an integrity management attribute) in decision making to safeguard the integrity of the asset, i.e., taking action (control reconfiguration, for example) based on fault/failure evidence, prognostic information, etc. We exploit the same concept in safety analysis.

3.6 Uncertainty Representation, Propagation and Management

Uncertainty in Prognosis is Probably the Most Significant Challenge Facing the PHM System Designer Uncertainty management tools seek to improve the fault signal to noise (uncertainty) ratio. They begin by determining the uncertainty sources in terms of an uncertainty tree and then exploit filtering or kernel-based methods for uncertainty management [31, 32]. We use particle filtering, as a nonlinear filtering method employing noisy observation data to estimate at least the first two moments of a state vector governed by a dynamic nonlinear, non-Gaussian state-space model.

Within the particle filtering framework, the Epanechnikov kernel is well suited for uncertainty representation in long-term prediction. Given initial conditions, it is possible to represent the uncertainty inherent to the predicted state

pdf by performing an inverse transform resampling procedure for the particle population [33]. This method obtains a fixed number of samples for each future time instant, avoiding problems of excessive computational effort. Furthermore, if only Epanechnikov kernels are used, it is ensured that the representation of the uncertainty will be bounded. These bounds intrinsically incorporate, measure, and represent model uncertainty (through the estimation of unknown parameters) and measurement noise (since the initial condition for long-term predictions corresponds to the output of the particle filtering procedure).

The issue of uncertainty management is related to a set of techniques aimed to improve the estimate at the current time instant, since both the expectation of the predicted trajectories for particles and bandwidth of Epanechnikov kernels depend on that pdf estimate.

In this sense, it is important to distinguish between two main types of adjustments that may be implemented to improve the current representation of uncertainty for future time instants:

- Adjustments in unknown parameters in the state equation.
- Adjustments in the parameters that define the noise PDFs embedded in the state equation, known as "hyperparameters".

Outer correction loops may be also implemented using neural networks, fuzzy expert systems, PID controllers, among others. Additional correction loops include the modification of the number of particles used for 1-step or long-term prediction purposes.

4 Resilient Design and Operation of Autonomous Systems

The Modeling Framework We take advantage of a rich array of modeling tools and methods representing the physical connections and dependencies of complex unmanned systems. We pursue in parallel Markov modeling-a probabilistic approach to represent complex systems, their states and state transitions. We formulate an Epidemic Spreading Model to estimate a probabilistic measure of system immunity and recovery time (i.e. self-healing). In the epidemic spreading model, disturbances are cascaded within the system model, and system components take on one of three states: susceptible, failed, or fixed (SFF model). Susceptible components are those that can be infected by a failed component, whereas fixed components are those that have healed. The densities of susceptible, failed, and fixed components change over time based on the system dynamics. The model is probabilistic due to the uncertainties (e.g. model uncertainty, state transitions), providing probabilistic measures of system immunity as well as recovery time (i.e. self-healing). The modeling toolset includes also structural and functional representations and dynamical system models that integrate disturbance factors into their structure.

In the context of a cyber-physical system's life degradation, we introduce a generalized heuristic modeling approach with consideration of critical disturbance/stress factors. Disturbance factor analysis considers the impact of varied operational and environmental disturbance factors on system end-of-life (EOL).

The Resilience Framework Resilience has been defined as the intrinsic ability of a system to adjust its functioning prior to, during, or following changes and disturbances, and thus, to sustain required operations even after a major mishap or in the presence of continuous stress [5]. Balchanos thoroughly reviewed various resilience-related research findings and addressed an assessment method of complex dynamic system resilience, which embraces system capability [34]. Tran also suggested a resilience assessment method based on time-dependent system reliability employing a probabilistic measure [35]. Situational awareness, prediction, planning, and action are required capabilities for a resilient system. A fault diagnosis and failure prognosis framework is assumed that has been implemented to provide situational awareness, and assess potential threats/hazards and their impact on system integrity [17].

The theoretical underpinnings for resilience evaluation borrow from Complexity Theory ("Critically interacting components self-organize to form potentially evolving structures exhibiting a hierarchy of emergent system behaviors") to set the stage for a rigorous understanding of complex system behaviors ("system-wide "emergent" behaviors which are difficult to predict from the behavior of any one element) and establish a basis for modeling of such platforms leading to the discovery of pertinent knowledge [36]. We use Kolmogorov Complexity (KC) as a metric of complexity. The full, or ultimate, exploitation of emergence is self-organization; a system aligns itself to a problem (internal or external disturbances) and is self-sustaining, even when the environment changes. We view self-organization as the basic principle for self-healing and immunity-the foundational elements in design of resilient systems.

4.1 Design for Resilience: the Enabling Technologies

Self-Organization The field of self-organization seeks general rules about the growth and evolution of systemic structure, the forms it might take, and finally methods that predict the future organization that will result from changes made to the underlying components under the influence of severe disturbances. Self-organization implies

that a system aligns itself to a problem and is self-sustaining, even when the environment changes under severe disturbance conditions. The enabling technologies begin with graph spectral and epidemic spreading modeling tools to represent the system behaviors under normal and faulty conditions; a Markov Decision Process is the basic self-organization module; decision-making is based on the current state only. The connectivity of a graph is an important measure of its resilience, as it indicates how much more node/edge disconnections can be tolerated until the graph is disconnected overall. Hence, the goal for a robust, reliable and resilient system under fault impacts would be to reorganize the graph and maximize the connectivity while observing the system constraints. A graph can be mathematically represented with the Laplacian matrix, which is defined as L = D − A, where L is the Laplacian matrix, D is the degree matrix (diagonal matrix showing the number of edges at each node), and A is the adjacency matrix (square matrix indicating the connection between nodes with 1s and 0s). The connectivity of a graph can be algebraically obtained by taking the second-smallest eigenvalue of the Laplacian matrix. The occurrence of severe disturbances can be represented with disconnections of corresponding nodes/edges, resulting in greatly reduced algebraic connectivity of the graph. To begin looking into the behavior of how a fault epidemic affects and spreads through a system, the transition matrix is defined as a square matrix with elements indicating the probability of traveling from node i to node j. The probabilities can be obtained through the derivation of a random walk normalized Laplacian, which can be written as $D^{-1}A$. This matrix serves as the transition matrix of a random walker on the graph, containing the likelihood of the epidemic spreading direction. Gathering the various matrices defined so far, a Markov Decision Process (MDP) can be constructed to observe the overall system behavior. MDP is a tuple consisting of {S, A, T, R, γ}, where:

- S = set of system states
- A = set of state-transitioning actions
- T = state transition matrix
- R(s,a) = reward for taking action a at state s
- γ = discount factor (to be further explained below)

In the MDP construction above, the reward function gives scalar values for each state transition, with greater value awarded to state transitions that result in moving toward the ideal behavior. A good definition of the reward function is the key to designing a resilient system, as the reward values can be constantly updated to optimize system resilience. The solution of the MDP will be in a form of a "policy," denoted by π and is the mapping from S to A, such that the system operation will proceed by repeating two steps: determine current state and execute action$\pi\ (s) = a$. Note here that the action is determined by only the current state and not the history of previous states, since the process is Markovian. To determine the policy, define a value function V(s) that accumulates the immediate rewards from each state along a series of state transitions. The optimal policy for a resilient autonomous system operation can be obtained by a dynamic programming method, namely the Bellman equation, written as

$$V(s) = \max\left(R(s,a) + \gamma \sum_{s'} T\left(s'|s,a\right) V(s')\right) \tag{4}$$

Here, the discount factor $\gamma \in [0, 1]$ suppresses the effect of future iterations to ensure convergence to a solution. The resulting system behavior following the obtained policy can be evaluated with proper resilience metrics. Due to limited resources (time, energy, etc.) or functional capabilities (joint angles, motor speed, etc.), tradeoff among the resources is inevitable in establishing system resilience. Some conditions for resilience metrics can be considered so that the metrics must be useful for decision making; should result in values so that the performance can be quantitatively assessed and compared; reflect uncertainty of the result; and consider failure recovery time. Overall, the optimal policy from MDP and the resilience metrics can be combined to ensure the system maintains its mission profile under severe disturbances.

Fault Tolerance Reconfigurable design of systems centers on incorporating autonomy and resilience, sustainment and reliability under changing operational requirements, severe disturbances (internal and external) and uncertain/dynamic environments or mission profiles, without major changes to the system's initial design. We address such challenging questions as i.) How does reconfiguration of one component affect the operation of other, neighboring, components? ii.) What is an appropriate strategy to maintain desired system behavior? We present a methodology for reconfigurable design and performance evaluation of complex systems paving the way for the design and construction of resilient, high-confidence autonomous systems. We are introducing a novel approach to fault-tolerance by considering the impacts of severe fault modes on system performances as inputs to Reinforcement Learning (RL) strategy that trades off system performances with control activity in order to extend the Remaining Useful Life (RUL) of the unmanned system so that a detrimental event does not occur in the presence of severe fault modes. The proposed approach employs a software solution and does not require a hardware complement to be integrated in the system design. The proposed architecture performs one of three actions, low-level control reconfiguration at the component level, mid-level control redistribution at the sub-system level and high-level mission adaptation

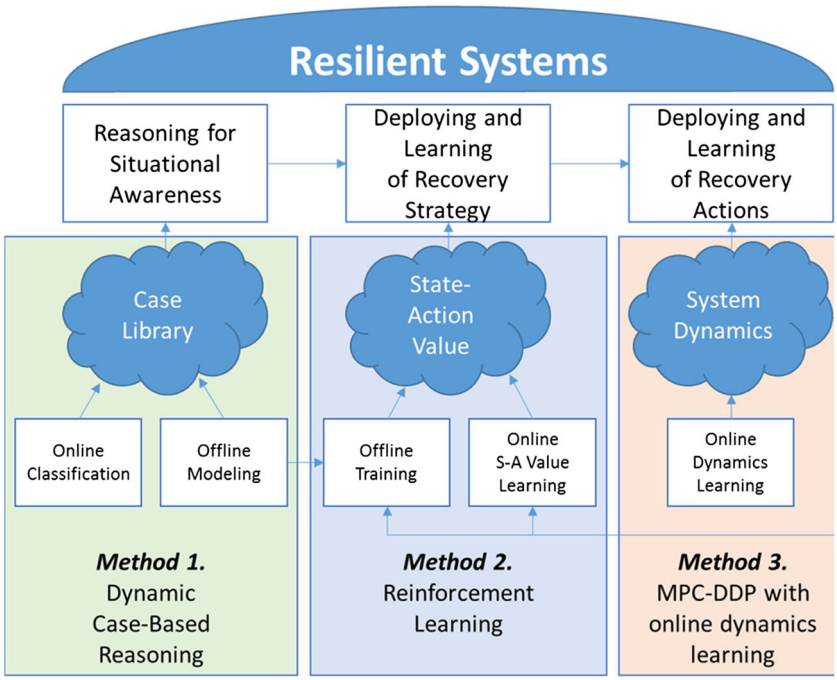


Fig. 6 Reconfiguration framework for resilient systems

at the highest echelon of the architecture. The fault-tolerant control begins with reconfiguration at the low-level since the impact of reconfiguration is localized to the individual component. If component reconfiguration is not sufficient to meet the mission objectives, control redistribution is performed at the middle level. The impact of control redistribution affects all components within the subsystem. This action provides more flexibility over component reconfiguration at the expense of increased computational complexity. Finally, if the previous actions are insufficient in achieving the desired objectives, mission adaptation is performed. During this action, lower-priority mission objectives are compromised or traded-off to achieve higher priority objectives. To handle sever disturbances, the middle-level reconfiguration plays critical roles for resilience in that trade-offs of system performances are actively considered. The theoretical underpinnings for the proposed resilience-based reconfiguration rely on concepts of DCBR, RL, and MPC-DDP as depicted in Fig. 6. The integrated framework of the three modules described above is a decision-making process, which optimizes control actions and system behaviors in order to extend RUL under severe fault modes. What distinguishes this framework from traditional reconfigurable control methods is to consider not only the current degraded states, but also consequences of them after all by RL.

RL is a supervised learning algorithm, seeking actions in environments (or mappings from states to actions) to maximize given rewards [37]. In RL, an agent takes an action in environments and observe changes of states accordingly. The observed states are translated into rewards, then an agent takes another action to collect maximum rewards over a given mission. What distinguishes it from typical optimal control methods is that the environment is not known to an agent; thus, the agent learns dynamics of environments by interacting with environments, so that it can choose optimal actions, which can produce the maximum rewards after all. In this sense, RL explicitly considers the whole picture of a mission.

RL utilizes the concepts of DP and Bellman's principle based on the MDP formulation of an environment, without the knowledge of system dynamics, but measurement data coming from interactions with environments. Since the environment is unknown, learning is realized by exploiting

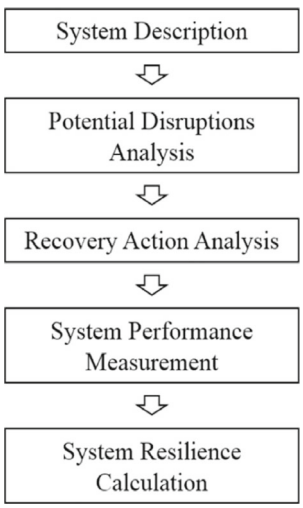


Fig. 7 Steps for resilience calculations and assessment

learnt policies and exploring an unknown state-action space [38]. There are two approaches: off-policy and on-policy methods. Off-policy methods, also called "an estimation policy", use a greedy search to determine control actions. A behavior policy makes decisions about control actions among all possible actions having a finite probability of being selected. On-policy methods, on the other hand, evaluate and improve control policies at the same time with a ε-greedy method, which chooses control actions by the probability, ε, to determine whether it takes a greedy action or random move. Both approaches include random moves, and it may cause unstable system behaviors during the exploring phase. To address this issue, RL learns the level of adaptation by adjusting an adaptation parameter of a cost function in MPC-DDP, in the proposed reconfigurable control framework. A general formulation of MPC in this framework addresses the solution of:

$$V(\mathbf{x}(t_0), t_0) = \min_{u} [\int_{t_0}^{t_f} l(\mathbf{x}(\tau), \mathbf{u}(\tau), \tau) d\tau + \Phi(\mathbf{x}(t_f), t_f)] \tag{5}$$

subject to:

$$\frac{d\mathbf{x}}{dt} = F(\mathbf{x}(t), \mathbf{u}(t)) \tag{6}$$

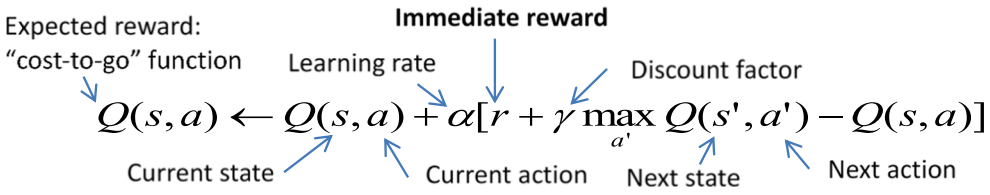
$$g(\mathbf{x}(t), \mathbf{u}(t)) \leq 0 \tag{7}$$

where t_0 is an initial time and t_f is a terminal time. $l(\cdot)$ is a scalar running cost, $\Phi(\cdot)$ is a scalar terminal cost, $F(\mathbf{x}(t),\mathbf{u}(t))$ is a generic nonlinear system dynamics as an equality constraint of the optimization problem, and $g(\mathbf{x}(t),\mathbf{u}(t))$ is a general function for inequality constraints.

The running cost function, $l(\mathbf{x}(\tau),\mathbf{u}(\tau),\tau)$, typically formulated as a quadratic function as shown in Equation 3. It is a linear combination of two terms: system performances and control efforts.

$$l(\mathbf{x}(t), \mathbf{u}(t), t) = \frac{1}{2}(\mathbf{x}(t) - \mathbf{r}(t))^T K(\mathbf{x}(t) - \mathbf{r}(t)) + \alpha \cdot \frac{1}{2}\mathbf{u}(t)^T R\mathbf{u}(t) \tag{8}$$

where $\mathbf{r}(t)$ is a reference, K and R are coefficient matrices, and α is an adaptation parameter. The right hand-side of Eq. 3 refers to the energy of the states and control inputs at each time instance. The adaptation parameter determines the trade-offs between system performances and control efforts. In this way, model-free and model-based optimal control techniques complement each other; MPC-DDP utilizes system dynamics to produce control actions stabilizing system behaviors for a finite time, and RL finds optimal weightings for the entire mission while it does not incur unstable behaviors during the exploration. Among RL methods, Q-learning is one of the most popular methods using TD learning. A general Q-learning formulation is:



(9)

The action a to be learned in the Q-learning process is α in the MPC formulation.

4.2 Resilience Metrics / Evaluation Criteria

The proposed resilience evaluation is composed of five steps, as illustrated in Fig. 7. The framework enables quantitative comparisons of potential system designs, with respect to their resilience to a set of disruptions (e.g. resource, hazards/threats, with multiple recovery actions considered). Each factor quantifies a particular aspect of system resilience. The approach uses system performance data to calculate a set of resilience factors and is independent of system complexity or heterogeneity. System performance is defined as a time-varying measure of how well a system is achieving a desired capability at a given time t. When assessing the resilience of a system, what the system is resilient to must be considered, since resilience of a system can be measured only in terms of the specific threat, requiring identification of potential threats, or disruptions. The framework also requires identification of potential recovery actions. These recovery actions enable a system to respond to disruptions and recover lost capabilities. In our case, the recovery actions refer to self-organization and control reconfiguration strategies, implemented after the disruption is identified. The notional data in Fig. 8 shows clear trends and smooth transitions in system performance. However, actual measured or simulated data is often volatile or noisy, due to the stochastic nature of many real or simulated processes/systems (UAVs), hence a Savitzky-Golay (S-G) filter method is used to smooth the performance data.

We introduce a resilience metric, R, in order to quantify a system's resilience to one disruption event. The metric R is formulated by identifying important characteristics of a resilient system, which are quantified by a set of resilience factors that contribute to system resilience, R.

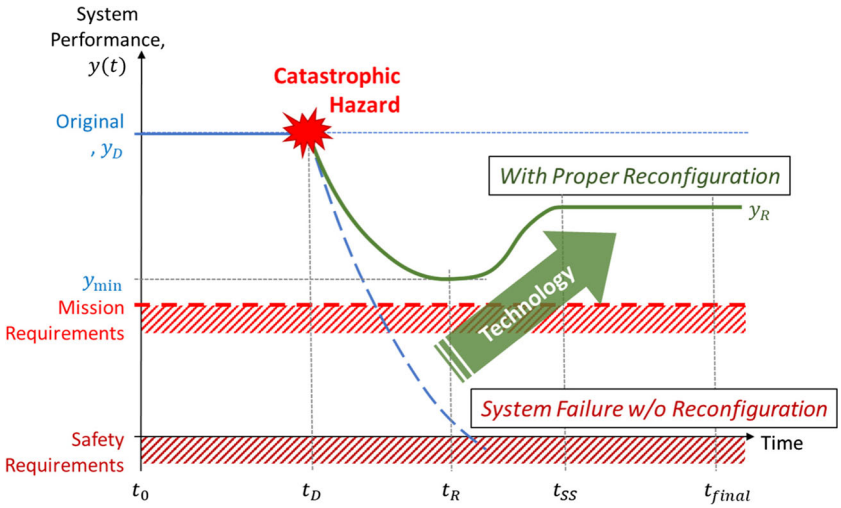


Fig. 8 Notional plots of system performance data with and without reconfiguration

The total performance factor, σ, accounts for the performance maintained by a system throughout the time period of interest. This factor is calculated as

$$\sigma = \frac{\sum_{t=t_0}^{t_{final}} y(t)}{y_D(t_{final} - t_0)} \tag{10}$$

where $0 \leq \sigma \leq \infty$, and has a value of one in a normal operating scenario. It is a function of the recovery factor, accounting for the end state of the system, the absorption factor accounting for the ability of the system to absorb the effects of a disruption.

Spectral matching is used to check if the self-organization method derived is proper at all. The spectral decomposition of the graph Laplacian associated with complex shapes provides eigenfunctions (modes) which are invariant to isometries. Each vertex on the shape could be uniquely represented with combinations of the eigenmodal values at each point, sometimes called spectral coordinates. Spectral matching consists of establishing the point correspondences by pairing vertices on different shapes that have the most similar spectral coordinates. Measuring consensus via system entropy allows evaluating the evolution of the distribution of nodal values, rather than the specific nodal values themselves. In addition, it will enable us to understand the structure of the underlying network in terms of its efficiency in attaining a consensus.

Table 1 Effect of failure mode on algebraic connectivity

# of failure mode	λ_2
0	0.3384
1	0.04874
2	$-3.0199e-16 \approx 0$

4.3 The Experimental Platforms

We have used a number of laboratory UAS platforms in the development and validation of the resilience framework. The first is a model hovercraft designed and built under NASA sponsorship and used for control reconfiguration purposes while the second is a typical ground UAS used to develop and demonstrate the self-organization strategy. The proposed self-organization method can combine the Markov Decision Process with Lyapunov stability conditions for a complex system to maintain stability under a severe failure mode. Test results indicate the robustness, stability and resilience characteristics while the modules of the framework performed in real time minimizing the computational burden. To illustrate the proposed self-organization method, a hexapod robot is selected as the test system. The mission profile is set for the hexapod to travel from a current point A to a goal point B in a straight-line path. Cully et al. suggest an improved trial and error method to determine the optimal action for a walking hexapod with a broken leg [39], but the large original search space and several minutes of lengthy adaptation time to the next step hamper their development. Instead, a self-organization method that spontaneously generates the optimal action can provide an alternative to decrease significantly the computational burden. An example of a possible failure mode in a hexapod is the locked joint failure, where a joint angle is fixed at a certain state and cannot be controlled. In the work of Yang, locked joint failures of different joints are shown to result in different effects on the leg workspace [40]. Since the Coxa joint is in charge of the horizontal swinging movement of the leg, locked joint failure at the Coxa joint completely disables the leg's swinging motion and the leg can only lift and plant itself vertically. On the other hand, locked joint failure at the vertically operating Femur and Tibia joints will have no effect on the swinging

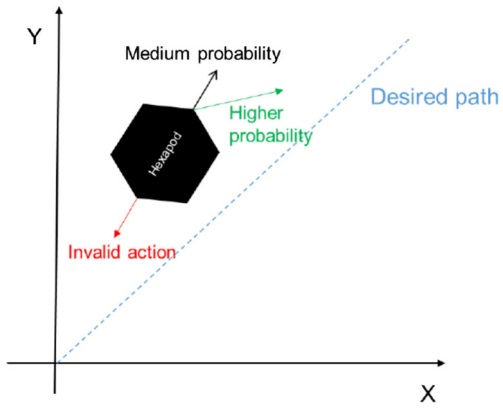


Fig. 9 Example visualization of the hexapod MDP

motion, but will diminish the leg's stretchable length, so the upper view of the leg's workspace will have the same arc shape but with reduced size. The impact of locked joint failures on the leg's workspace will cause the hexapod to derail from its original path in an unexpected manner.

To examine the effect of locked joint failure in a graph theoretic aspect, we can consider one unit of a hexapod robot as a graph with 18 nodes (6 legs $\times$ 3 joints). The aforementioned algebraic connectivity can be used to observe the effect of locked joint failure on the hexapod's resilience. Demonstration of the effect of locked joint failure on algebraic connectivity is shown in Table 1, where λ_2 is measured in normal condition, then with one joint failure, and finally with two joint failures.

It can be seen that the incremental addition of failure modes decreases the algebraic connectivity, and when there are two failure modes present, the graph becomes disconnected overall.

An MDP (Markov Decision Process) is formulated as the self-organization strategy for a hexapod under locked joint failure. The state space S can be defined as the global position of the hexapod, and the action space A as the necessary control inputs that enable the robot to move from one valid state to another. More specifically, the state refers to the global x, y, and θ (orientation) of the hexapod, and the corresponding actions are the Cartesian and angular velocities to be applied. As shown in Fig. 9, the desired path from initial position to goal position is known, and the reward function is defined so that higher reward (or probability) is assigned to action resulting toward the desired path. Finite deterministic case (discrete state/action space) of the MDP algorithm is assumed for simplicity. The solution of the MDP will be a policy that maps the optimal action for each state for the hexapod to move along the desired path.

The self-organization strategy for a hexapod is tested in VREP (Virtual Robot Experimentation Platform), which is an open source robot simulator with an integrated development environment. The hexapod starts at the origin and the diagonal path of y = x on the XY plane is assumed to be the desired path for the hexapod to travel. A failure mode is then implemented to the hexapod by dislocating the left-front Coxa joint as the locked joint failure in the corresponding leg. Finally, the proposed self-organization method is applied in the simulator, where the hexapod's

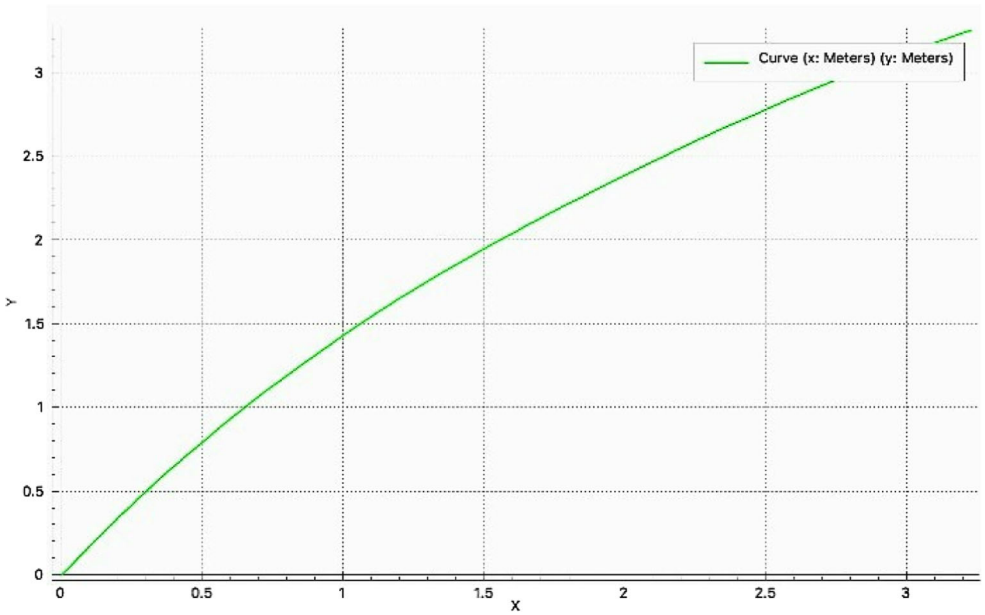


Fig. 10 Self-organizing hexapod path under failure mode

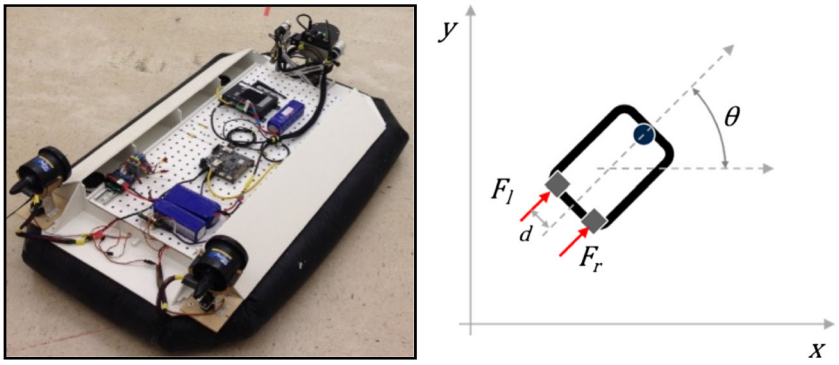


Fig. 11 The autonomously operable hovercraft with two differential thrusts (left), and 2D hovercraft dynamics and kinematics representation (right)

traveling behavior is governed by an MDP. The resulting path is shown in Fig. 10, where the hexapod starts away from the desired path but steadily turns its direction in the process to reach the desired path. In terms of resilience, the main goal for the hexapod was to reach the desired path, so the trajectory in this case may not be ideal in the perspective of time or total travel distance.

To demonstrate the efficacy of the reconfiguration strategy, an autonomously operable under-actuated hovercraft is used as a testbed [41, 42]. The hovercraft operates with two differential thrust fans with electrical motors and a LIDAR sensor for simultaneous localization and mapping, as depicted in Fig. 11

The hovercraft is assumed to move in two-dimensional planar motion; thus, it is an under-actuated system given two input controls. Equations 11 are the system dynamics model; x and y are absolute positions on the ground fixed coordinate, θ is a heading angle, $\dot{X}$ is a velocity, $\ddot{X}$ is an acceleration, m is the mass, J is the moment of inertia of the hovercraft, d is the distance between a thruster and an imaginary longitudinal line crossing the mass center while assuming that the mass center coincides with the geometric center, and F_l & F_r are left and right thrust forces, respectively. Based on the system dynamics equations, the state is $\mathbf{x}=\{x,y,\theta,\dot{x},\dot{y},\dot{\theta}\}^T$, and the input is $\mathbf{u}=\{F_l,F_r\}^T$. Han, and Zhao evaluated the underactuated hovercraft controllability [43]. The analysis showed that the existence of the yaw torque can guarantee the system controllability. It implies that one thrust motor failure does not affect the controllability as long as the other motor can produce proper torque values.

$$\begin{aligned}\ddot{x} &= -\tfrac{d_t}{m}\dot{x} + F_l \cdot \cos\theta + F_r \cdot \cos\theta \\ \ddot{y} &= -\tfrac{d_t}{m}\dot{y} + F_l \cdot \sin\theta + F_r \cdot \sin\theta \\ \ddot{\theta} &= -\tfrac{d_r}{J}\dot{\theta} + d(F_r - F_l)\end{aligned} \tag{11}$$

Table 2 shows the system properties used for the following experiments.

The hovercraft test mission is to take off at origin (0, 0) and follows waypoints (80, 80), finally reach the destination (160, 80). During the mission, the right thrust motor degrades due to an armature winding short as illustrated in Fig. 12.

As a pre-identified critical fault mode, armature winding short is selected at a thrust motor. A winding short increases effective resistance of a motor circuit. It implies the waste of electric energy: i.e., the loss of efficiency. Because of the inefficiency, eventually, the effective thrust force of the faulty motor decreases causing control asymmetry. If the winding short gets intense, then it is no longer possible to produce any thrust forces, and it makes the hovercraft uncontrollable anymore.

Table 2 System properties

Parameters	Values	Description
m (kg)	11.8	Vehicle mass
J ($kg \cdot m^2$)	1	Moment of Inertia
d (m)	0.25	Moment arm
d_t (–)	0.05	Frictional damping (translation)
d_r (–)	0.005	Frictional damping (rotation)
F_{max} (N)	2	Control input constraint (max.)
F_{min} (N)	0	Control input constraint (min.)

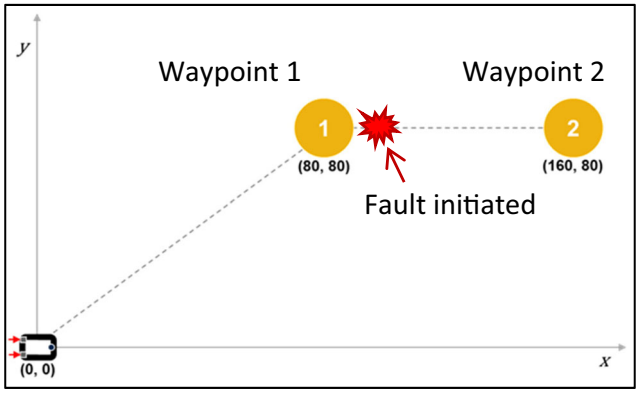


Fig. 12 Test mission and scenario

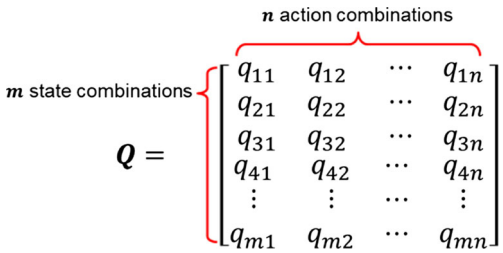


Fig. 13 Q-value table

For the given scenario, RL states were formulated below:

$$\mathbf{s}_t^{RL} = \{d_t, \dot{d}_t, \sigma_t, \dot{\sigma}_t\}^T \tag{12}$$

where d_t is a remaining distance (shortest), and σ_t is a state of fault estimated by a fault diagnosis module.

Actions are:

$$\alpha = \{\alpha_L, \alpha_R\}^T \tag{13}$$

where L and R denote left and right adaptation, respectively.

For a simple proof of concept, a tabular-based Q-learning was used as shown in Fig. 13. State and action spaces were discretized into 20 and 2 levels, respectively; thus, ($_{20}C_2$-by-4) Q-value table was made. Through training, the Q-value table was gradually updated until converged.

Figure 14 shows comparisons of hovercraft trajectories. Figure 14a is a trajectory of healthy hovercraft controller by a dynamic-inversion controller. It successfully followed waypoints as designed. Figure 14b is a case when the right thrust motor went degraded gradually during the operation without control reconfiguration. As expected, it eventually got unstable and could not reach the target. Figure 14c shows a trajectory when the hovercraft used an MPC-DDP controller without adjusting the adaptation parameter. The trajectory shows that an MPC-DDP controller could guide the hovercraft to the direction of a target after the fault mode, but still could not reach the destination at the end. It was because the right fault motor went completely off (failure) before completing the mission, and so the hovercraft lost its controllability. Figure 14d is a successful reconfiguration accomplishing the mission with proper adjustment of the adaptation parameter by RL.

The reconfigurable control strategy was tested also on the Georgia Tech VTOL UAV called GTMax (Fig. 15). Under the auspices of the DARPA Software Enabled Control (SEC) program, the GTMax was instrumented appropriately

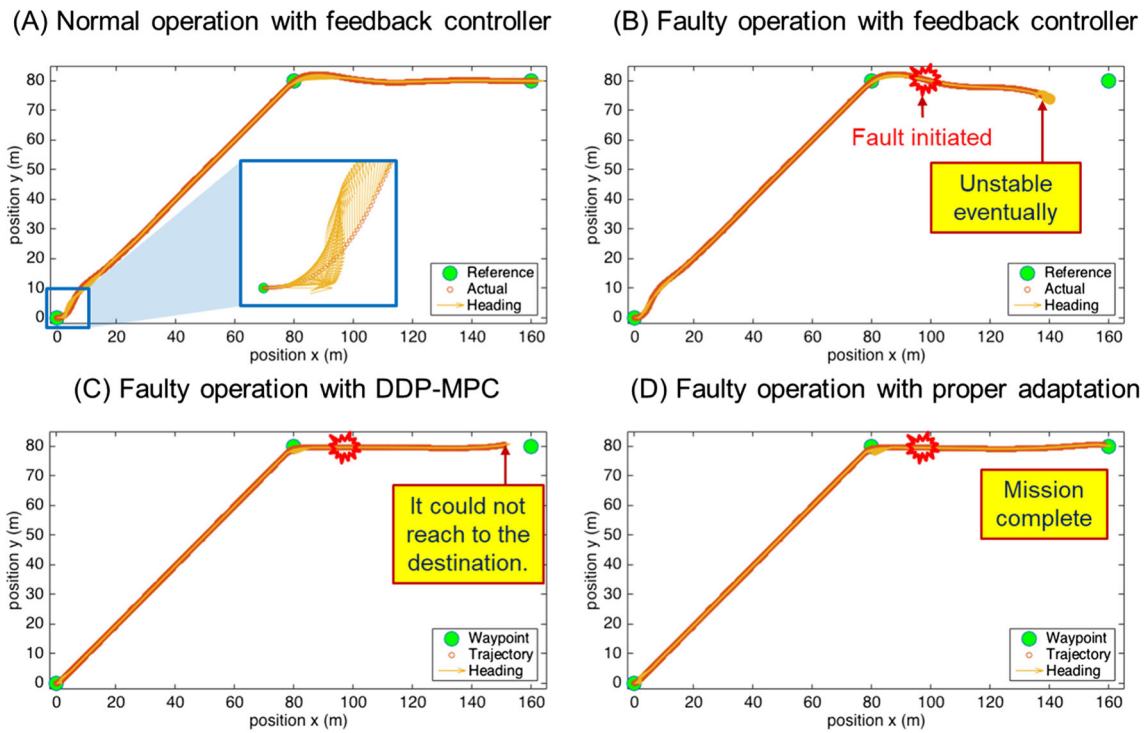


Fig. 14 Comparisons of hovercraft trajectories: **a** Healthy, **b** Faulty without reconfiguration, **c** Faulty with a fixed adaptation parameter, and **d** Faulty with RL-based adaptation

Fig. 15 The Georgia Tech VTOL UAV

and flown to execute a bob-up maneuver. During flight, a failure mode was injected in the collective and a variable rotor speed used to stabilize the vehicle.

5 Safety in Hostile Environments – Risk Assessment, Evaluation, Assurance and Control

Healthy and Resilient Assets Mean Safer Operations We pursue an integrated and verifiable methodology to **safety assurance** (Fig. 16). Safety assurance enables the evaluation of the effectiveness of risk management strategies.

The basic constituents of the framework include:

- The process begins with the identification of potential hazards - hardware, software, the environment, and human factors are major sources of hazards. We adopt Predictive Models to analyze hazards and assess their impact on vehicle safety.
- Define safety margins - Safety margins are defined and designed as an automatic envelope protection system. We will adopt a probabilistic approach to safety assurance and define appropriate safety margins in the context of risk assessment.
- Safety assurance: A probabilistic approach is pursued for safety assurance.
- Risk index and risk control - Risk is quantified in terms of the scenario of events leading to hazard exposure, the likelihood of the scenario and a measure of its consequences. Concepts of envelope protection make use of on-line learning adaptive neural networks to generate on-line dynamic models exploited to estimate limits on controller commands.
- Optimize system design on the basis of safety analysis methods

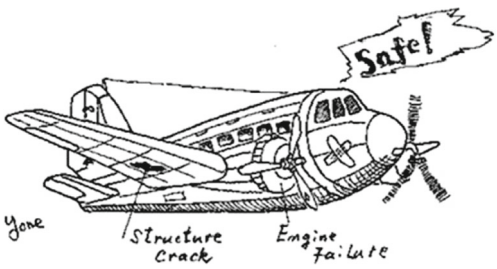


Fig. 16 Safety

5.1 Safety Assurance: Confidence, Risk and Risk Control

Confidence is a measure of reliability, i.e. how reliable a statistical result is, expressed as a percentage, and indicating the probability of the outcome from a decision system (fault declaration, control effectiveness, prognostic horizon, safety assurance) being correct. Confidence is usually linked to the concept of risk. The latter, with focus on aerospace applications, is associated with the probability of component/subsystem failures and the probability of aircraft loss-of-control for a chosen control configuration. Consider the notion of a safe set, depicted in Fig. 17, and employed to quantify criteria associated with risk.

A number of safety and hard limits, i.e. reduced aircraft envelope, are employed to define the safe set. The control objective is to derive the optimal reconfiguration policy that minimizes the probability of excursion outside of a predetermined safe set, or maintain the vehicle dynamics within the safe set. It is commonly anticipated that the flight envelope or safe set may shrink in the presence of a severe fault/failure condition. Under these circumstances, the reconfiguration strategy's intent is to bring the aircraft back within the redefined safe set. In this case, the concept of risk is associated with the inverse of the distance between the current state of the system and a critical safe envelope, assuming certain operating conditions, as it will be detailed in the sequel. Confidence, on the other hand, is

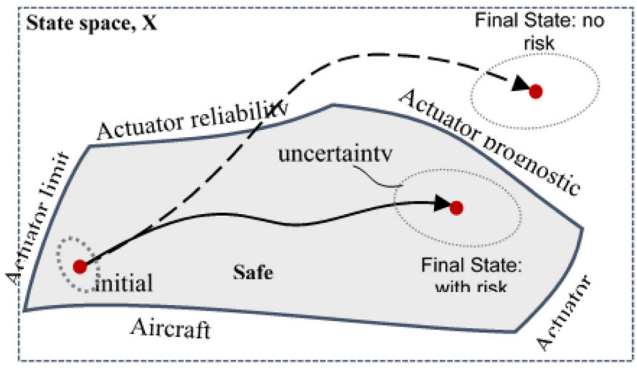


Fig. 17 Safe set definition

the probability that the system is bounded by an interval that is obtained from the conditional distribution.

Optimal risk management policies are aiming to reconfigure the aircraft flight controller appropriately when damage/hazard is detected/isolated and is severe enough to compromise the operational response of the asset. In such situations, the probability of component failure and the probability of aircraft loss-of-control for a chosen control configuration may be estimated as:

$$p^g_{s_0} = Prob^g_{s_0} s_k \in A \text{ for all } k \in [ON] | s_0 \in A \tag{14}$$

This is a stochastic reachability problem, where s_k is a state at time k, A is a safe set, and g is a control policy (control gain selection for linear case).

We choose the reconfiguration policy (detailed in the resilient design and operation of a UAS section of this document) such that this probability is maximized (thus risk is minimized).

The control mode is selected to minimize the probability of mission failure and may be computed on-line as:

$$V^*_o = \sup_g p^g_{s_o}(A) \tag{15}$$

$V_k$is a value function at time k.$V^*_N(s_k)$ is initialized as $V^*_N(s_k) = 1_A(s_k)$. $1_A$is an indicator function (1 if s_k is within A, 0 otherwise).

This control policy formulation maximizes the probability of remaining inside the safe set A defined by the stability limits and the Remaining Useful Life (RUL) limit of the failing component/subsystem. Examples from the aircraft domain are plentiful addressing issues of actuator and control faults, among others.

Within the general framework for risk assessment and risk management, we explore means to take corrective action with acceptable risk, i.e. we are seeking an essential link between failure prognosis and reconfigurable control. Towards that goal, we suggest ways to quantify risk and uncertainty. The notion of confidence may come in two different perspectives: In the first one, the user specifies an acceptable level of confidence and we are seeking to quantify risk, while in the second, an acceptable risk level is given and the confidence in taking a corresponding corrective action is sought.

For that purpose, we borrow concepts from actuarial science and we define a quantity called Fault Value at Risk (FVaR) to provide on-line an estimate of the severity of the fault/failure condition under study. $FVaR(t, t_{prognosis})$ is the maximum increase in fault dimension $l(t)$ that can occur within time$t$ after the time of $t_{prognosis}$. The FVaR at the confidence level α is given by the smallest number $l(t)$ such that the probability that the damage (degradation, fault dimension) $L(t)$ exceeds $l(t)$ is not larger than (1- α), i.e.

$$FVaR(tt_{prognosis}) = \inf(l(t) \in \Re : PL(t) > l(t) | y_{t_{prognosis}} \leq 1-\alpha) \tag{16}$$

Assuming that the fault dimension coincides with the first component x^1_t of the state vector, then the 95% confidence FVaR can be computed as:

$$FVaR(tt_{prognosis}) \Leftrightarrow \alpha = 0.95 = \int_{-\infty}^{FVaR(t,t_{prognosis})} \hat{p}(x^1_t | y_{t_{prognosis}}) dx^1_t \tag{17}$$

The FVaR function provides information about the future condition of the system. Based on this fact, it is also possible to create a risk index that would consider the difference between the expected value of the hazard zone (which generally defines the most frequent value of the fault dimension at the failure time instant) and the FVaR computed with 95% confidence.

$$Risk_{FVaR}(tt_{prognosis}) = (E\{HazardZone\} - FVaR(t, t_{prognosis}))^{-1} \tag{18}$$

Different load conditions will lead to dissimilar risk functions. The risk measurement is calculated via an estimate of the probability of violating specified limits.

Fault Value at Risk (FVaR):

$$\alpha = \int_{-\infty}^{FVaR(t_{future}, t_{prognosis})} \hat{p}(x_{t_{future}} | v_{t_{prognosis}}) dx_{t_{future}} \tag{19}$$

For a given confidence α, compute: $FVaR(t_{future}\text{-}t_{prognosis})$

Solve via a recursive algorithm – The argument is the predicted pdf (Fig. 18).

Safety Margins and Risk Assessment The objective is to evaluate risk in decision making to safeguard the safety of the asset, i.e., taking action (control reconfiguration, for example) based on fault/failure evidence, prognostic information, etc. Safety margins are defined and designed as an automatic envelope protection system. Within the system's operating envelope, it may be possible for the system's behavioral modes to escape from the stable region of operation, under severe stress conditions, endangering its safety and survivability. Concepts of envelope protection make use of on-line learning adaptive neural networks to generate on-line dynamic models exploited to estimate limits on controller commands. The work borrows from research conducted at Georgia Tech and implemented on the Georgia Tech's GTMax, a VTOL UAV [44]. Recent advances in flight control systems enable autonomous maneuvering that can challenge an Unmanned Aerial Vehicle's (UAV) flight envelope [45]. Typically, conservative hard limits are set

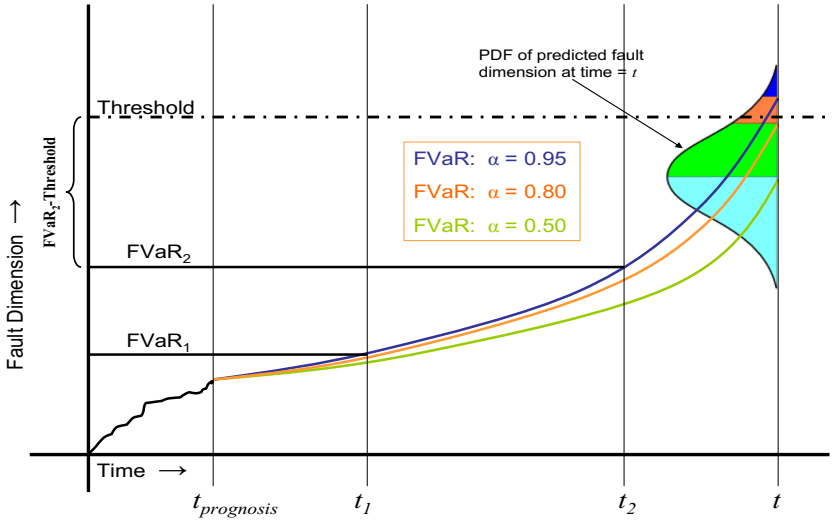


Fig. 18 Fault value at risk for varying confidence levels

in a UAV's flight control channels as maximum and minimum allowable command inputs. An effective automatic envelope protection system will reduce the compromise between safety and performance, thus improving the overall confidence of safe operations of UAVs especially during aggressive maneuvering close to their operational limits. The limit avoidance via command limiting is evaluated in flight for the case of rotor stall limit avoidance, as an example, on the Georgia Tech unmanned helicopter test bed. The Envelope Protection System (EPS) was evaluated as part of the GTMax Rotary Wing Experiments during the initial, mid-term and final demonstration of the SEC program [46]. We consider an envelope protection strategy via command limiting algorithms, their integration with the low-level adaptive flight control algorithms, and software-in-the-loop (SITL) simulation for demonstration purposes. The proximity of a limit parameter to its limit boundary is called the limit margin. The limit margin is easily determined when the instantaneous value of the limit parameter can be calculated directly using sensor measurements. However, a challenge is the determination of *future* limit margins to prevent limit violations due to the dynamic nature of limit parameter response. In an unmanned vehicle application, where envelope protection is to be automated, the system can be set-up in two fundamental architectures to incorporate a modular envelope protection design into a UAV's existing flight control system: Control Limiting and Command Limiting. Figure 19 depicts the control limiting architecture. A preferred method for an automatic EPS is the use of command limiting, shown schematically in Fig. 20. In this architecture, the model is viewed as a combination of the low-level controller and the vehicle, instead of the vehicle dynamics only. We pursue appropriate modeling, estimation and control methods in defining and implementing safety margins via control and command limiting techniques. The scheme is used to address two types of potential UAV hazards while the vehicle is executing extreme maneuvers: A load factor limit and a rotor stall limit.

Safety Assurance: A Probabilistic Approach The objective is to optimize system design and UAS operations based on safety analysis methods. In probabilistic design, the probability of failure is defined as the probability of violating a set of engineering criteria called "limit state". Let the state of the component be described by the parameters, $X = \{x_1, x_2, \ldots, x_k$. These design variables represent the time-dependent demands and capacities of the component/system under test. Demands can be expressed in terms of loads, stresses, hazards or other draws on the component/system. Capacities are the ability of the component to meet these demands, and are expressed as

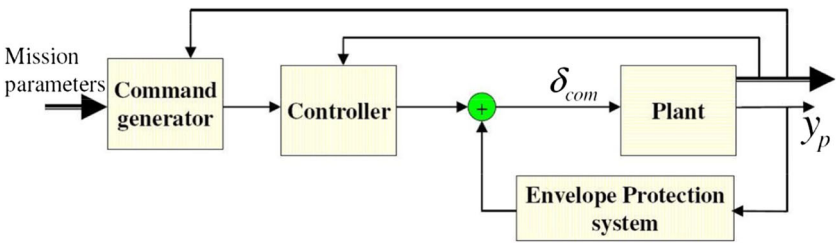


Fig. 19 Control limiting architecture

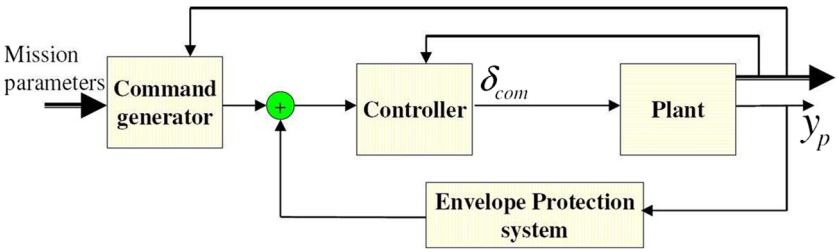


Fig. 20 Command limiting architecture

strengths, resistances, etc. These design variables are often stochastic, due to inherent uncertainties.

The safety of a component is quantified as a function of its design variables, g(X). This limit state function describes three regions in the design space, as shown in Fig. 21.

Any component state that lies in the g(X)<0 region is said to be unsafe or failed. Usually, the limit state function is described as a classic stress-strength problem, such that theprobability of failure, p_f, described by overlap of component state PDF, f_X, and limit state function $g(X)$.

$$g(\sigma_R(\mathbf{X})\sigma_S(\mathbf{X})) = \sigma_R(\mathbf{X}) - \sigma_S(\mathbf{X}) \tag{20}$$

where $\sigma_R(\mathbf{X})$ and $\sigma_S(\mathbf{X})$ are the cumulative strengths (capacities) and stresses (demands), respectively, and the unsafe region is described as any state where the component stress exceeds its strength.

Because the design variables $x_1, x_2, \ldots, x_k$ are stochastic, the component state must be represented as a joint probability density function (PDF),$f_X(x_1, x_2, \ldots, x_k)$. The amount of this PDF that lies in the unsafe region is the same as the probability that the component is in a failing state, and thus the probability of failure, p_f is expressed via the integral:

$$p_f = \int \cdots \int_{g(X)<0} f_x(X_1 X_2, \ldots, X_n) dx_1 dx_2 \ldots dx_n \tag{21}$$

Issues to consider are how the design space and limit state function are defined, how to describe or estimate the pdf, $f_X(x_1, x_2, \ldots, x_k)$, and how to solve the integral and obtain p_f. Solution of the integral numerically or analytically is a daunting task. We seek therefore approximations to a feasible solution.

The first-order safety/reliability method (FOSM) preserves only the first-order terms of g(X), linearizing the limit state function about what is called the design point.

$$g(X) \approx g(\mu_X) + \sum_{i=1}^{n} \frac{\partial g}{\partial X_i} (X_i - \mu_{X_i}) + \cdots \tag{22}$$

The design point, denoted by X*, is chosen as the nearest point to the component state, X, on the limit state curve g(X) = 0. For convex regions defining pf, linearizing about the nearest point produces the best possible first-order approximation for pf. The distance from the current component state, X, to the nearest point on the limit state curve, X*, is proportional the safety/reliability of X. A term, β, is defined as the safety index of the current state, such that $\beta \propto ||X^* - X||$. Formally, the safety index is the unitized risk, the dimensionless coefficient of variation$\beta = \frac{\mu_Z}{\sigma_Z}$, where Z ≡ g(X).

If Z is normally distributed and g(X) is linear, then p_f is simply defined as the cumulative distribution function (CDF) at the first coefficient of variation,

$$p_f = \Phi(-\frac{\mu_Z}{\sigma_Z}) = \Phi(-\beta) = 1 - \Phi(\beta) \tag{23}$$

where $\Phi(\cdot)$ is the CDF of the standard normal variant (Fig. 22).

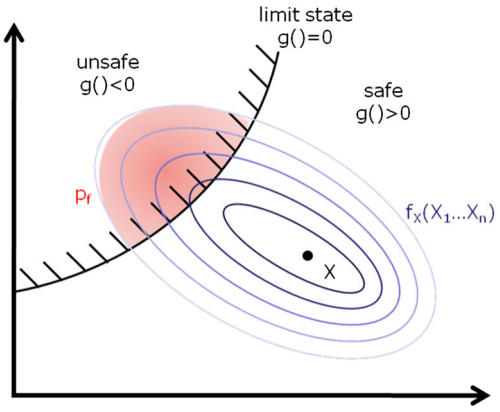


Fig. 21 Definition of three limit states

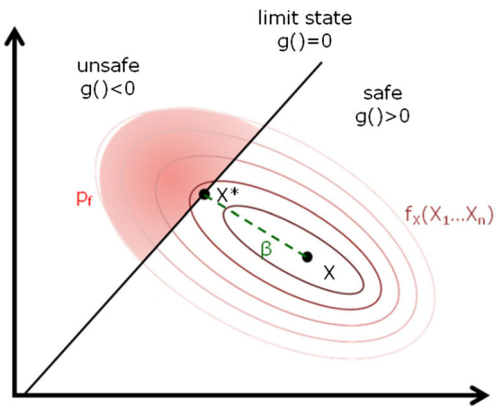


Fig. 22 $g(X)$ linearized about X^*

Many algorithms are available to solve for μ_Z and σ_Z or β—or its variants. Three standard approaches are the mean value first-order second-moment (MVFOSM) and the advanced first-order second moment (AFOSM) for normal variables and AFOSM for non-normal variables. Two steps are involved in these approximation methods to compute easily the integral. The first step is to simplify the integrand $f_X(\mathbf{X})$ so that its contours become more regular and symmetric, and the second step is to approximate the integration boundary g(X)=0.

We introduce a time-dependent FOSM to calculate the probability of failure as a function of time.

Step One: Formulate the time-dependent limit state function g(X, t)

We assume all the random variables X are mutually independent, and the limit state can be written as a time-dependent classic "stress-strength" problem of the form

$$g(\sigma_R(\mathbf{X},t)\sigma_S(\mathbf{X}t)) = \sigma_R(\mathbf{X},t) - \sigma_S(\mathbf{X}t) \tag{24}$$

where $\sigma_R(\mathbf{X}t)$ and $\sigma_S(\mathbf{X}t)$are the cumulative strengths (capacities) and stresses (demands) at time t.

For each time increment, the random design variable is calculated using the component/system degradation model as a function of the previous damage status, time increment, model parameters Θ and load profile U:

$$X_i^{(t+\Delta t)} = \eta(X_i^{(t)}, \Delta t, U, \Theta) \tag{25}$$

The limit state is updated $g(\sigma_R(\mathbf{X},t)\sigma_S(\mathbf{X}t))$for each point in time.

Step Two: calculate the probability of failure at each time point pf (ti) using the FOSM algorithm:

Simplify the integrand $f_X(\mathbf{X})$:

Transform the original random variables from the original design space (X-space) to standard normal space (U-space) by Rosenblatt transformation, which preserves the same quantity of the cdfs of the random variables before and after the transformation [47],

$$F_{X_i}(x_i) = \phi(u_i) \tag{26}$$

$\phi(\cdot)$ is the cdf of the standard normal distribution.

After the transformation, the limit state function becomes g(U), and the probability integration reduces to

$$\mathrm{P_f} = \mathrm{Prob}\{g(\mathbf{U}) \le 0\} = \int \cdots \int_{g(\mathbf{U})\le 0} \phi_U(\mathbf{u})\mathrm{d}\mathbf{u} \tag{27}$$

Approximate the integration boundary g(U)=0.

FOSM uses a linear approximation (the first order Taylor expansion). To minimize the accuracy loss, it is preferable to expand the limit state function g(U) at a point that has the highest value of the probability density, which is termed as the Most Probable Point (MPP).

The solution to the model, MPP, is denoted by $\mathbf{u}^* = (u_1^*, u_2^*, \ldots, u_n^*)$. The MPP is the shortest distance point from the limit state g(U)=0 to the origin O in U-space. The minimum distance $\beta = ||\mathbf{u}^*||$ is called the safety/reliability index.

Step Three: Repeat Step 2 until a time trace of failure distribution pf (t) is obtained.

Step Four: Fit a life distribution to get the time-dependent safety/reliability distribution F(t).

g(X) is linearized about the "design point" X* on the limit state. The distance between state X and the nearest design point X* is the "safety" of the component state.

Let β be the safety index:

$$\begin{array}{ll} \beta = \frac{\mu_z}{\sigma_z} & \min imize \;\; ||X - X^*|| \\ \beta \propto ||X - X^* & g(X^*) = 0 \end{array} \tag{28}$$

Risk Index and Risk Control A risk index considers the difference between the expected value of the hazard zone (which generally defines the most frequent value of the fault dimension or safety margin at the failure time instant) and the FVaR computed with 95% confidence,

$$Risk_{FVaR}(t, t_{prognosis}) = (E[HazardZone] - FVaR(t, t_{prognosis}))^{-1} \tag{29}$$

A risk score may be defined as:

$$RiskScore = (SererityValue) \times (LikelihoodValue) \tag{30}$$

The severity value is estimated from the fault/failure analysis while a likelihood value is calculated from the proximity of the current state to the limit state.

Risk Assessment begins with the definition of a risk matrix whose columns describe the risk categories from catastrophic to negligible and rows designate frequency of occurrence from frequent to extremely improbable. The numerical entries start with a score of 1 for the extremely improbable and negligible risk ending with a score of 25 for the most frequent and catastrophic. The color coding suggests areas of high risk (red), medium risk (yellow) and low risk factors (green). A detailed risk analysis requires a statistically sufficient data base to arrive at an accurate risk matrix whose entries are probability density functions. Risk management or risk control is intended to limit risk to aceptable bounds by developing and aplying tools/methods for improved system safety. The FAA lists in the System Safety Handbook severity definitions. Figure 23 is a typical risk matrix.

Safety hazards include: lost UAV – out of range or to the enemy, UAV in erroneous state, unexpected human interaction with UAV, erroneous target discrimination,

Fig. 23 The risk matrix

Risk	Catastrophic	Hazardous	Major	Minor	Negligible
Frequent	25	20	15	10	5
Occasional	20	16	12	8	4
Remote	15	12	9	6	3
Improbable	10	8	6	4	2
Extremely Improbable	5	4	3	2	1

enemy jamming or taking control, loss or inadequate situational awareness, battle damage to UAV, provision for emergency safety operator/pilot, UAV exposure to toxic substances.

We propose to pursue appropriate modeling, estimation and control methods in defining and implementing safety margins via control and command limiting techniques. The scheme is used to address two types of potential UAV hazards while the vehicle is executing extreme maneuvers: A load factor limit and a rotor stall limit.

Safety Risk Management provides a workflow for a formal process to describe the system, identify hazards, assess risk, and control/minimize risk. We define possible hazard scenarios, quantify their frequency of occurrence and estimate/predict their consequences. Once the hazard analysis tasks are set in probabilistic terms, the design for risk assessment takes over and addresses quantitative and qualitative risk factors. Safety Assurance enables the evaluation of the effectiveness of risk management strategies and ensures compliance with oversight entities. The process begins with the identification of potential hazards, as noted previously. The safety nmanagement and assurance framework incorporates a dynamic risk management system that is dynamically updated and scales appropriately to risk. We will adopt Predictive Models to analyze hazards and assess their impact on UAS safety. Figure depicts a Risk Matrix listing potential hazard categories vs frequency of occurrence. Because of the inherent uncertainty associated with hazards, models and risk assessment, the variables/parameters are expressed as PDFs.

Confidence is a measure of safety/reliability, i.e. how reliable/safe a statistical result is, expressed as a percentage, and indicating the probability of the outcome from a decision system (fault declaration, prognostic horizon, etc.) being correct. Confidence is usually linked to the concept of risk. Within the general framework for risk assessment and risk management, we will explore means to take corrective action with acceptable risk, i.e. we are seeking an essential link between failure prognosis, exceedance of safety margins and reconfigurable control to maintain the system integrity.

For that purpose, we borrow concepts from actuarial science [14] and we define a quantity called Fault Value at Risk (FVaR) to provide on-line an estimate of the severity of the fault/failure condition or safety margin exceedance.$FVaR(t, t_{prognosis})$ is the maximum increase in fault dimension $l(t)$ that can occur within time$t$ after the time of $t_{prognosis}$. The FVaR at the confidence level α is given by the smallest number $l(t)$ such that the probability that the damage (degradation, fault dimension) $L(t)$ exceeds $l(t)$ is not larger than (1- α).

We intend to apply these algorithmic developments to prognostic routines, safety assessment for critical unmanned system components / subsystems.

The Application Domain Resilience and safety methods are finding a rich application domain in rotor wing and other aircraft, unmanned autonomous systems, “smart” manufacturing processes, industrial processes, among others.

Modeling/Simulation/Visualization Platforms To demonstrate the efficacy of the research, high fidelity modeling, simulation and visualization tools are ready in mature states. Microsoft Excel is a great simulation and visualization tool with the power of macro capability. For more math-oriented tasks, Matlab and Simulink are highly popular in the lab from data acquisition, manipulation and system dynamics modeling to data analysis, visualization and user-friendly Graphical User Interfaces (GUIs). Powerful math and engineering library enables users easily write scripts. Simulink in Matlab allows even easier ways of modeling by a drag-and-drop and connection interface for electrical, pneumatic and many other systems. For the cases when statistically enhanced analysis is required, SAS-JMP is used. It has various statistical function libraries handling large-size data. In addition, JMP has flexible charting capability and its

own script language. It enables our lab to show statistical results in a custom-designed GUI. Robotics simulation is carried out with Robot Operating Systems (ROS) and Gazebo environment. ROS is a middle-ware, which handles message packets from/to each different component module including sensors and controllers. Gazebo is a high-fidelity 3D rigid body dynamics simulator. It runs in ROS; thus, each module does not have to be different between software simulations and hardware tests. As system-of-systems simulation capability, ModelCenter is an integration framework of multiple and cascading simulations. It allows multiple case simulations at once by Monte Carlos simulation or a Design-of-Experiments setup.

With missions and parameters defined, the next step is to consider several possible modeling approaches. The possible options are:

- Physics Based: Gazebo/ROS, USARSim
- Analytical: Using closed-loop equations and experimental data to run sizing analyses on changing configuration.
- Matlab/Simulink with CAD: Virtual design of the vehicle in CAD with configurations and use a Matlab/Simulink model to see the performance variations based on the setup.

6 Conclusions

Unmanned autonomous systems are making their presence felt in all sectors of our economy from the military to civilian and the industrial arenas. Unfortunately, technologies to assure their resilience to extreme internal/external disturbances are not keeping pace with their exponentially increasing numbers. It is documented that unmanned systems too frequently are failing to execute their designated mission. To improve UAS availability, safety and reliability when operating in uncertain and cluttered environments, this paper introduced a holistic framework for the design and operation of unmanned systems even when subjected to hazards/threats that may endanger their operational integrity. Hazard analysis and prognostics methods for reliability were stated. The enabling technologies of the resilient UAS control methods borrowed from immunity and self-healing concepts as well as control techniques to impart on such critical systems properties of resilience and safety. The design and operation of resilient and healthy systems assures their safety. Probabilistic methods serve as the risk assessment criteria which is crucial to safety assurance/management. Open questions remain on the theoretical front to expand the introduction of verifiable algorithms and ascertain that they can be fully implemented on-platform meeting stringent computational requirements. Success stories will assist to show proof of concept and make these emerging technologies acceptable to the user.

References

1. David, R.A., Nielsen, P.: Defense Science Board Summer Study on Autonomy. Defense Science Board Washington United States (2016)
2. Holling, C.S.: Ecosystems the complexity of economic, ecological and social systems. Ecosystems **4**, 390–405 (2001)
3. Gunderson, L.H., Allen, C.R., Hollng, C.S.: Foundations of Ecological Resilience (2009)
4. Hollnagel, E., Woods, D.D., Leveson, N.: Resilience Engineering: Concepts and Precepts. Ashgate Publishing, Ltd (2007)
5. Hollnagel, E.: Resilience engineering: Why, what, and how (2007)
6. Prehofer, C., Bettstetter, C.: Self-organization in communication networks: principles and design paradigms. IEEE Commun. Mag. **43**(7), 78–85 (2005)
7. Heylighen, F.: The science of self-organization and adaptivity. The Encyclopedia of Life Support Systems **5**(3), 253–280 (2001)
8. Zhang, Y., Jiang, J.: Bibliographical review on reconfigurable fault-tolerant control systems. Ann. Rev. Control **32**(2), 229–252 (2008). https://doi.org/10.1016/j.arcontrol.2008.03.008
9. Clements, N.S.: Fault tolerant control of complex dynamical systems. Doctoral dissertation Georgia Institute of Technology, Atlanta, USA (2003)
10. Ge, J., Kacprzynski, G.J., Roemer, M.J., Vachtsevanos, G.: Automated contingency management design for UAVs. In: AIAA 1st Intelligent Systems Technical Conference, pp. 20–22. https://doi.org/10.2514/6.2004-6464 (2004)
11. Drozeski, G.R., Saha, B., Vachtsevanos, G.: A fault detection and reconfigurable control architecture for unmanned aerial vehicles. In: Aerospace Conference. IEEE. https://doi.org/10.1109/AERO.2005.1559597 (2005)
12. Tang, L., Kacprzynski, G.J., Goebel, K., Saxena, A., Saha, B., Vachtsevanos, G.: Prognostics-enhanced automated contingency management for advanced autonomous systems. Prognostics and Health Management, International Conference, IEEE, pp. 1–9, IEEE. https://doi.org/10.1109/PHM.2008.4711448 (2008)
13. Brown, D.W., Georgoulas, G., Bole, B., Pei, H.L., Orchard, M., Tang, L., Saha, B., Saxena, A., Goebel, K., Vachtsevanos, G.: Prognostics enhanced reconfigurable control of electro-mechanical actuators. In: Annual Conference of the Prognostics and Health Management Society (2009)
14. Bole, B., Tang, L., Goebel, K., Vachtsevanos, G.: Adaptive load allocation for prognosis-based risk management. In: Annual Conference of the Prognostics and Health Management Society, pp. 1–10 (2011)
15. Bole, B.M.: Load allocation for optimal risk management in systems with incipient failure modes. Doctoral dissertation Georgia Institute of Technology. Atlanta, USA (2013)
16. Scientific American, April 1, 2015
17. Vachtsevanos, G., Lewis, F., Roemer, M., Hess, A., Wu, B.: Intelligent Fault Diagnosis and Prognosis for Engineering Systems. Wiley (2006)
18. Zhang, B., Khawaja, T., Patrick, R., Vachtsevanos, G., Orchard, M.E., Saxena, A.: Application of blind deconvolution denoising in failure prognosis. IEEE Trans. Instrum. Meas. **58**(2), 303–310 (2009)

19. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**, 28 (2015)
20. Deng, L., Yu, D.: Deep learning: Methods and applications. Found Trends Signal Process **7**(3–4), 1–199 (2014)
21. Bengio, Y.: Learning deep architectures for AI (PDF). Found. Trends Mach. Learn. **2**(1), 1–127 (2009)
22. Roemer, M., Byington, C., Kacprszynski, G., Vachtsevanos, G., Goebel, K.: Prognostics in Systems Health Management with Aerospace Applications, pp. 281–295. Wiley (2011)
23. Orchard, M.: A particle-filtering based framework for on-line fault diagnosis and failure prognosis. School of Electrical and Computer Engineering, Georgia Institute of Technology. Atlanta, GA : s.n., Ph.D Dissertation (2007)
24. De Martin, A., Jacazio, G., Vachtsevanos, G.: Windings fault detection and prognosis in electro-mechanical flight control actuators operating in active-active configuration IJPHM, (best paper award) (2016)
25. Shyy, W., Lian, Y., Tang, J., Vileru, D., Liu, H.: Aerodynamics of low Reynolds Number Flyers. Cambridge University Press, Cambridge (2007)
26. Chapman, A., Mesbahi, M., Swarms, U.AV.: In Valavanis, K.P., Vachtsevanos, G. J. (eds.) Models and Effective Interfaces, Handbook of Unmanned Aerial Systems, pp. 1987–2019. Springer (2015)
27. Establishing Trust Is Greatest Challenge to Increasing System Autonomy, Aviation Week & Space Technology Sep 13, (2010) p. 48
28. Giordano, V., Ballal, P., Lewis, F.L., Turchiano, B., Zhang, J.B.: Supervisory control of mobile sensor networks: matrix formulation, simulation and implementation. IEEE Trans. Syst. Man Cybern. – Part B **36**(4), 806–819 (2006)
29. Mireles, J., Lewis, F.L.: Intelligent material handling: development and implementation of a matrix-based discrete-event controller. IEEE Trans. Ind. Electron. **48**(6), 1087–1097 (2001)
30. Theodorakopoulos, G., Baras, J.S.: On trust models and trust evaluation metrics for Ad Hoc networks. IEEE J. Select. Areas Commun. **24**(2), 318–328 (2006)
31. Orchard, M., Tang, L., Goebel, K., Vachtsevanos, G.: A novel RSPF approach to prediction of high-risk, low-probability failure events. In: First Annual Conference of the Prognostics and Health Management Society – PHM09. San Dieg (2009)
32. Edwards, D., Orchard, M., Tang, L., Goebel, K., Vachtsevanos, G.: Impact of input uncertainty on failure prognostic algorithms: Extending the remaining useful life of nonlinear systems. Prognostics and Health Management Conference (2010)
33. Orchard, M.: A particle-filtering based framework for on-line fault diagnosis and failure prognosis. School of Electrical and Computer Engineering, Georgia Institute of Technology. Atlanta, GA : s.n., 2007. Ph.D Dissertation
34. Balchanos, M.G.: A probabilistic technique for the assessment of complex dynamic system resilience. Doctoral dissertation Georgia Institute of Technology. Atlanta, USA (2012)
35. Tran, H.T.: A complex networks approach to designing resilient system-of-systems. Doctoral dissertation Georgia Institute of Technology. Atlanta, USA (2015)
36. Lucas, C.: Self-Organizing Systems Usenet FAQ (2003)
37. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction, vol. 1. MIT Press, Cambridge (1998)
38. Altunas, N., Imal, E., Emanet, N., Ozturk, C.N.: Reinforcement learning-based mobile robot navigation. Turkish J. Electric. Eng. Comput. Sci. **24.3**, 1747–1767 (2016)
39. Cully, A., Clune, J., Tarapore, D., Mouret, J.: Robots that can adapt like animals. Nature **521**(7553), 503–507 (2015). https://doi.org/10.1038/nature14422
40. Yang, J.: Fault-tolerant gait generation for locked joint failures. In: 2003 IEEE International Conference Systems, Man and Cybernetics, p. 8. https://doi.org/10.1109/ICSMC.2003.1244216 (2003)
41. Kim, K., Lee, Y., Oh, S., Moroniti, D., Mavris, D., Vachtsevanos, G.J., Papamarkos, N., Georgoulas, G.: Guidance, navigation, and control of an unmanned hovercraft. In: 2013 21st Mediterranean Conference Control & Automation (MED), pp. 380–387. IEEE. https://doi.org/10.1109/MED.2013.6608750 (2013)
42. Sconyers, C., Lee, Y., Kim, K., Oh, S., Mavris, D., Oza, N., Mah, R., Martin, R., Raptis, I.A., Vachtsevanos, G.J.: Diagnosis of fault modes masked by control loops with an application to autonomous hovercraft systems. International Journal of Prognostics and Health Management (2013)
43. Han, B., Zhao, G.L.: Course-keeping control of underactuated hovercraft. J. Marine Sci. Appl. **3**(1), 24–27 (2004). https://doi.org/10.1007/BF02918642
44. Johnson, E., Kannan, S.: Adaptive trajectory control for autonomous helicopters. J. Guid. Control Dyn. **28**(3), 534–538 (2005)
45. Yavrucuk, I., Unnikrishnan, S., Prasad, J.V.R.: Envelope protection in autonomous unmanned aerial vehicles. In: Proceedings of the American Helicopter Society 59th Annual Forum, vol. 2, pp. 2000–2010. Phoenix (2003)
46. Heck, B., Wills, L., Vachtsevanos, G.: Software technology for implementing reusable, distributed control systems. IEEE Control Syst. Mag. **23**(1), 21–35 (2003). (IEEE Control Systems Magazine Outstanding Paper Award for the years 2002-2003)
47. Rosenblatt, M.: Remarks on a multivariate transformation. Ann. Math. Statist. **23**(3), 470–472 (1952)

George Vachtsevanos is a Professor Emeritus of Electrical and Computer Engineering at the Georgia Institute of Technology. He was awarded a B.E.E. degree from the City College of New York in 1962, an M.E.E. degree from New York University in 1963 and the Ph.D. degree in Electrical Engineering from the City University of New York in 1970. He directs the Intelligent Control Systems laboratory at Georgia Tech where faculty and students are conducting research in intelligent control, neuro-technology, fault diagnosis and prognosis of large-scale dynamical systems and control technologies for Unmanned Aerial Vehicles. His work is funded by government agencies and industry. He has published over 300 technical papers and is a senior member of IEEE and fellow of the PHM society. Dr. Vachtsevanos was awarded the IEEE Control Systems Magazine Outstanding Paper Award for the years 2002-2003 (with L. Wills and B. Heck). He was also awarded the 2002-2003 Georgia Tech School of Electrical and Computer Engineering Distinguished Professor Award and the 2003-2004 Georgia Institute of Technology Outstanding Interdisciplinary Activities Award.

Benjamin Lee is an Electrical Engineering graduate student at the Georgia Institute of Technology since 2013. He obtained his Bachelor's degree in Electrical Engineering from the Georgia Institute of Technology in 2013. He has participated in researches of structural health monitoring using acoustic waves, developing and testing of user interfaces for situation awareness in life support systems, and developing tracking control system for a unicycle mobile robot. His main research areas include resilient system design methodology and self-organizational control methods.

Sehwan Oh is a Ph.D. candidate in the ASDL at the Georgia Institute of Technology since 2010. He has participated in graduate researches of a turbine engine model regression analysis, Navy transformable ship design, risk analysis of the integration of unmanned aerial vehicle systems into the national airspace system, and smart and sustainable campus design and analysis. His main research areas include resilience system design methodology and control reconfiguration.

Dr. Michael Balchanos is research faculty with the Daniel Guggenheim School of Aerospace Engineering, where he serves as the Naval Systems Research lead under the Defense and Space (D&S) Division at the Aerospace Systems Design Laboratory (ASDL). His areas of expertise include research work in dynamic systems modeling and simulation methods, as well as SoS-level integration techniques for enabling decision support in complex and resilient systems design, involving several applications such as smart energy infrastructures, electric reconfigurable naval ships and Unmanned Aerial Vehicles (UAVs). He has also been developing similar methods for Electric Vehicle (EV) energy-based sizing and optimization applications, as well as the development of SoS-level frameworks for the connected autonomous mobility ecosystem of the future. He obtained his Diploma in Physics from the Aristotle University of Thessaloniki, Greece and his M.Sc. and Ph.D. degrees in Aerospace Engineering from Georgia Tech.