Impact and Influence of Cyber-Physical Systems Research on Autonomous Aerospace Systems

Justin Bradley*
University of Nebraska-Lincoln, Lincoln, NE 68588, USA

Cody Fleming† and Kristin Y. Rozier‡
Iowa State University, Ames, IA 50011, USA

Amy Pritchett§
Penn State University, University Park, PA 16802, USA

Cyber-Physical Systems, as a discipline, is relatively new, appearing prominently between 2000-2010, but has rapidly made contributions in many disciplines. The aerospace industry has been a primary application domain for CPS from the beginning, with much focus on autonomous systems, particularly Unmanned Aircraft Systems. Aerospace systems provide an opportune, complex, safety-critical system on which to prove out CPS research, techniques, and strategies where robustness, agility, and provable performance guarantees are important. In this paper we survey and discuss the synthesis of core CPS research areas into four key areas of autonomous aerospace systems. We illuminate how these have been applied across aviation and space systems that interface with human operators and/or bystanders. We also discuss how computing and select CPS concepts more prominently applied in other applications could be applied in aerospace systems. Finally, we elaborate on existing shortcomings in cyber-physical aerospace systems and propose future avenues of exploration and application to pursue.

I. Introduction

Cyber-physical systems (CPSs) "combine[s] cyber systems (computational systems such as microprocessors and digital communication networks) with other physical systems (electromechanical, chemical, structural, and biological systems)" [1]. Generally, this applies to a large swath of engineered systems many of which have their own niche research areas. Likewise, the topics of research that contribute to CPS span a wide array of disciplines. This may lead some to conclude that CPS research is a catch-all term attempting to supersede or capture individual, more targeted areas of research. We distinguish between CPS research and CPSs and remark that many existing research disciplines contribute to the advancement of engineered CPS but may not be thought of as "CPS research." We define "CPS research" as research focused on advancing the "fundamental scientific and engineering principles that underpin the integration of cyber and physical elements across all application sectors."*. As a matter of practice, a few research areas have been identified as CPS research: control, data analytics, and machine learning including real-time learning for control, autonomy, design, Internet of Things (IoT), mixed initiatives including human-in- or

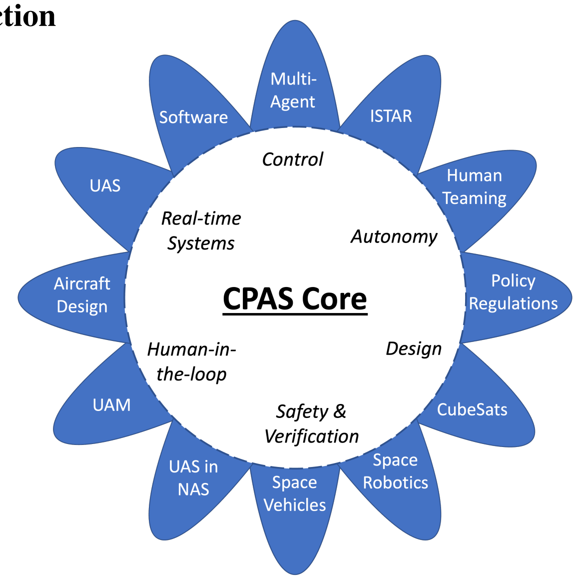

Fig. 1 Core cyber-physical aerospace research areas and application areas.

*Associate Professor, School of Computing, AIAA Associate Fellow
†Associate Professor, Mechanical Engineering, AIAA Senior Member
‡Black & Veatch Associate Professor, Department of Aerospace Engineering, AIAA Associate Fellow
§Professor and Department Head of Aerospace Engineering, AIAA Fellow
*NSF CPS Program slides `https://nsf.gov/attachments/297925/public/CPS2019-Webinar.pdf`

human-on-the-loop, networking, privacy, real-time systems, safety, security, and verification†; this list fluctuates as different topics emerge at prominent CPS venues.

To avoid yet another CPS survey paper (of which there are many), in this paper we focus on surveying the synthesis of control, autonomy, design, human-in- or human-on-the-loop, real-time systems, safety, and verification into autonomous aeronautical and astronautical systems (see Figure 1). While an individual survey paper could be written about each of these individual research areas applied to aerospace systems, we target the integration and synthesis of these concepts into the holistic design and deployment specifically targeted at aerospace systems and their attributes that include real-time operation of flight dynamics and strict requirements for safety. The intent is to give the reader a view of how CPS research is impacting integration efforts in deployed aerospace systems, the limitations of those efforts, and what areas may need more consideration.

II. CPS Origins, Surveys, and Challenges

Conceptually, cyber-physical systems stem from the field of cybernetics – the scientific study of control and communication in the animal and the machine [2, 3] – and hence has been around in various forms for many years. Of more relevance, the term "Cyber-Physical Systems," as a discipline, was coined in 2006 by Helen Gill at the National Science Foundation [4]. Since then the field has grown quickly and rapidly as researchers and practitioners wrestle with the science of cyber and physical integration in engineered systems. In the aerospace sector, the area of most rapid uptake of CPS concepts has perhaps been autonomous Unmanned Aircraft Systems (UASs) wherein significant constraints on Size, Weight, and Power (SWaP) have demanded novel methods for engineering capable, safe, resilient flight.

A challenge in the broader community appears to be how best to understand what constitutes a CPS and CPS research. For such a young discipline, a relatively large number of related surveys seek to summarize the state of the art and describe how CPS research applies to various disciplines, and the challenges that lie ahead in advancing the field. Here, we discuss existing CPS surveys and describe how our work here differs.

A. CPS Surveys

A large number of CPS survey papers review existing work in a broad array of research fields (as opposed to application sectors). This speaks to the power of CPS research to impact many research areas as well as the excitement of researchers in applying CPS concepts to their core areas. The areas of security and safety have the largest number of surveys [5–12], and includes surveys of CPS concepts in blockchains [13, 14], secure control [5], and even a survey of surveys in security in CPS [9]. These surveys discuss the unique challenge of tightly integrating cyber and physical components to detect both traditional computation-based attacks, as well as physical infrastructure attacks, ensure privacy, and maintain robust safety.

A number of surveys examine a cornerstone of CPS research – the development of novel modeling schemes [15–19] spanning mixed discrete/continuous systems, verifiable models, and models for model-based design paradigms [20–23]. Other central research areas such as control [24–26], data [27, 28], and networking [29, 30] have had multiple CPS-based surveys published. Testing has received less attention [31], suggesting perhaps that the community has yet to develop many strong CPS testing techniques. Related to this is a single survey on test case generation for software in safety-critical CPS [32], indicating opportunity for software engineering and CPS researchers to collaborate.

CPS surveys have also addressed application domains such as the medical [33–35], and power/smart grid sectors [11, 36–38]. Finally, researchers have recognized the importance of integrating fundamental CPS research concepts into education and have compiled surveys in this important area [39, 40].

B. Work Discussing Future CPS Challenges

Accompanying surveys on CPS research in application domains, and core concepts are publications discussing the various challenges that can be addressed by taking a CPS research approach. Several of these appeared early on as CPS research was just taking off [41–47], but discussions of such challenges have continued even as CPS has matured [19, 48, 49]. Addressing technological challenges via CPS research has also been a critical topic in several President's Council of Advisors on Science and Technology (PCAST) reports [50–52]

In this work we contribute a survey of CPS research to the aerospace domain. However, we focus particularly on the unique challenges in the aerospace industry and synthesize CPS research contributions applied to those challenges. We

†NSF CPS Program `https://beta.nsf.gov/funding/opportunities/cyber-physical-systems-cps`

aim to show how more holistic considerations of complex aerospace systems can help propel the industry toward a more robust, safe, and integrated future, particularly important in an industry in which so many legacy systems remain.

III. Real-time Systems & Control

Real-time systems and control have played an outsized role in the CPS community, starting with its origins in the DARPA and AFRL sponsored "Software Enabled Control" program [53–56]. Much of the contents of the seminal book [53] have gone on to become foci of the CPS community. Naturally, through practitioners and academics, much of this has led to advances in commercial aviation, aerospace defense, as well as the emerging small Unmanned Aircraft System (UAS) community. In this section we divide the impact of CPS research on aviation into two thrusts: 1) the influence of computing, its theory, concepts, and algorithms on aerospace control, and 2) the impact, and intersection of computing and control in aviation.

Influence of Computing on Aerospace Control High speed computing has enabled many new control paradigms. For example, the X-33 pushed computing and control boundaries significantly, in part because vehicle dynamics changed profoundly across flight conditions. More computationally intensive adaptive control strategies [57, 58], including neural nets (NN) [59] were leveraged to create full-flight envelope controllers avoiding the more traditional gain scheduling which was not feasible due to the explosion of possible operating points. NNs have continued to grow significantly in size and complexity requiring a mechanism to build in safety measures from the start that prevent adverse learning in the case of non-linear/discrete/hard-limit effects such as control saturation [60].

Much of the influence in the CPS community on control has stemmed from the growth of hybrid systems [53, 61], leveraging the ability of automata to model both continuous and discrete dynamics of a complex system [62, 63]. Hybrid systems were, and still are, prominent in many research efforts spanning everything from low-level control [64], to formation flight [65], to air traffic management [66].

Hybrid system modeling paradigms and extensions have expanded rapidly and many extensions have emerged allowing designers to model other cyber and physical phenomena holistically. Simultaneously, modeling tools, themselves a part of the cyber-physical design system, have grown significantly in capability and allowed for rapid design, iteration, model checking, and model-based design [67]. Model-based design [68, 69], wherein models are used to design, verify [70], synthesize [71], and produce executable code [72] for a portion of an aerospace system, ideally provide a direct coupling between implementation and design (see Figure 2). This reduces the opportunity for mismatches between implementation and design [73] and allows for designers to update, modify, and version designs, rather than implementation. The document "Software Considerations in Airborne Systems and Equipment Certification," DO-178B and DO-178C [74], provide acceptable means through which software may comply with federal regulations, and with which MBD tools (e.g., Matlab, Simulink) comply [21]. While not fully clear how much MBD and synthesis are used in large, real-world projects, NASA famously employed MBD on the Orion project about which they have written extensively [20, 75–78].

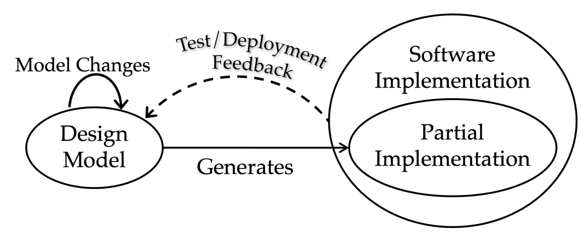

Fig. 2 Ideal Model-based Design Process.

The complexity of modern controllers composed of discrete, continuous, and neural net (NN) components require extensive analysis, testing and verification to ensure performance. As a result, formal methods [70] occupy a significant place in the CPS community and have been applied liberally in aerospace [79–81] (discussed more in Section VI). However, MBD, synthesis, verification and similar tools are costly in both time and money to build into production and research processes. For small businesses and research groups in the smaller, less costly aerospace sector (e.g., small UASs), this means these tools have not yet found widespread adoption despite the safety-critical nature of those systems. Indeed, ArduPilot[‡], an extremely popular open-source autopilot for autonomous systems, has relatively few formal models and primarily develops autonomous systems directly in software [82].

Aside from the interconnected thread of hybrid systems, synthesis, formal methods, and machine-learning, traditional discrete algorithms have also found extensive use in autonomous aerospace systems including graph search [83], gridding/mapping [84], and automata such as Markov Decision Processes [85, 86].

[‡]`https://ardupilot.org/`

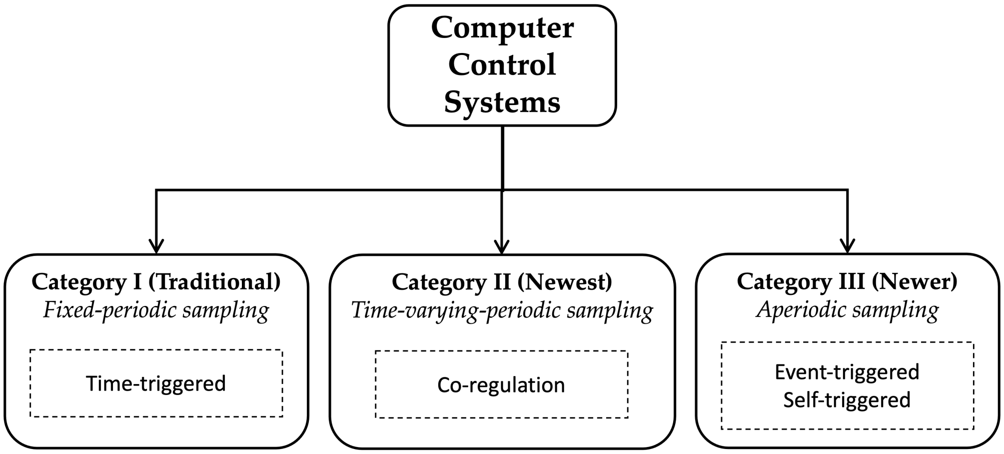

Fig. 3 Classes of Computer Controlled Systems.

Intersection of Computing and Control Computer control is ubiquitous and hence so is the struggle of ensuring assumptions about software execution, timing, and computer memory match reality – an exceptionally difficult task. Most relevant to safety-critical aerospace systems is the consideration of timing, and the enforcement of deadlines on hard real-time control tasks necessitating the employment of a real-time operating system (RTOS) [87–89], including assessing the schedulability of computing tasks [90].

While presumably standard practice in the commercial and defense aerospace sector, in the small UAS and satellite sector RTOSs may still be scarce owing to the complexity of writing RTOS software and deployment on inexpensive hardware. Most small autonomous autopilot packages (e.g., ArduPilot, PX4) utilize an RTOS (e.g., ChibiOS[§]), however, interacting software on companion computers are typically still Linux-based "best effort" implementations. The ubiquitous Robot Operating System[¶] is a microcosm of the current state of affairs. While ROS2 employs numerous improvements on the computational side (e.g., QoS, asynchronous messaging) ROS1 is often still the first choice for small businesses and researchers due to its prevalence in the community and the difficulty of switching to ROS2.

Assuming implementation on an RTOS, for the control engineer the difficulties start with selecting a fixed sampling period enabling the controller to be robust to anticipated worst-case scenarios. Over the years exploration of the topic has been applied in both aviation [91, 92] and spacecraft [93], though often sample rate selection remains based on "rules of thumb" [94].

More recently, as chip manufacturers have hit physical limits on clock rates, and advanced algorithms and neural nets have increased computational demands, a renewed effort to consider computation as a scarce resource have also seen a resurgence. On the control side, in direct contrast to time-triggered control, event-triggered and self-triggered control represent the pinnacle of computation utilization – only executing control inputs when needed, as triggered by some condition in the system. Such methods have primarily been applied in the research space to small UAS [95]. Limitations in analytical methods, performance, and perhaps more significantly, implementation, have hampered their wider deployment in safety critical systems. A primary disadvantage is the difficulty of ensuring computational timing guarantees for aperiodic tasks, as well as differentiation between failures and the lack of a triggering event [96].

From the control perspective, one view of the computational control paradigm can be seen in Figure 3 with traditional time-triggered/periodic control systems on the left, and event-triggered/aperiodic control systems on the right [97]. In the middle, Category II, in Figure 3, are the newest computer control paradigms wherein resources, and hence performance, are adjusted online according to the system's needs and objectives. Figure 4 shows how this type of system might work. Traditional fixed-periodic control systems select a single operating point typically based on worst-case anticipated scenarios (red line). Time-varying periodic control strategies allow the operating point to move along the blue line to adjust performance and resources on the fly.

Co-regulation [98] is an example of this strategy whereby the periodic sampling rate is dynamically adjusted in response to vehicle performance. This requires a new class of controllers and real-time schedulers [99, 100]. Co-regulation in both UAVs [101] and multi-agent UAV systems [102] show similar performance to time-triggered

§ `https://www.chibios.org/dokuwiki/doku.php`

¶ `https://www.ros.org/`

controllers, but has significant resource savings that can be put to use where the system needs it. Complementary work in adaptive resource management adjusts computing resources to algorithms that can make use of freed computation cycles [103]. Both of these strategies envision control as a service [104], the quality of which can be adjusted, and anytime algorithms [105, 106] that can be employed to leverage available resources.

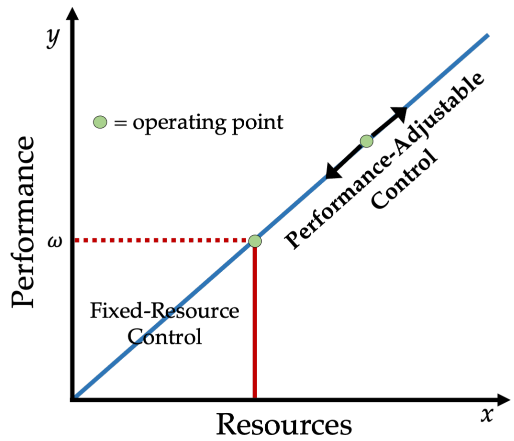

Fig. 4 Performance and Resource Adjustable Control.

Challenges An overarching challenge in the CPS control community has been to elevate computation to a first class citizen in the design, testing, and deployment of CPSs. In a modern aerospace vehicle computers physically impact the design of the structure, its power/energy budget, and computationally impact nearly every facet of the onboard controls, avionics, autonomy, and communication. Here we describe two main challenges for the cyber-physical aerospace control community to consider: 1) support for computational and performance adaptability; and 2) safety-critical control software.

To make aerospace systems more adaptable, their computation and performance must also be adaptable. Above, we listed several solutions to developing controllers able to dynamically adjust their performance and computational requirements. On the computing side, while a few supporting adaptive real-time computing task models have been developed, they have primarily targeted the automotive industry, and seen limited real-world deployment [107–109]. Task models that can allocate compute cycles to where they're most needed are required to allow existing anytime algorithms to leverage resources and produce improved results.

Along these same lines, machine-learning based control is rapidly emerging, and while still not widely deployed in safety-critical software, they likely will be in the future [110–112]. However, often the pursuit in ML-based control has been controllers that demonstrate excellent performance even at the expense of complexity. This limits their deployability on Size, Weight, and Power (SWaP) constrained systems. ML-based components of aerospace systems should be designed in conjunction, and in context with computation and other components of the system to ensure they're not overdesigned [113]. Additionally, to realize adaptable performance, ML-based perception and control strategies would benefit from design principles used in anytime algorithms since their "best" performance may not be needed at all parts of the flight envelope.

Software has become a critical aspect of the aerospace community in recent years. Most often, research in aerospace control software has focused on mismatches between the intended design and the execution of the software [114, 115], or conversely, sought to verify that software conforms with specifications [116]. Strategies to ensure this have included annotations in the software that correspond with desirable properties, essentially "integrating proofs within the code" [117]. Ultimately, efforts along these lines, like model-based design (discussed above), intrinsically assume that implementation may not conform to design, that decoupling between them will occur, and attempt to rectify it. We take the view that despite advances in MBD or synthesis, unless every aspect of a system can be modeled to the finest detail, control software *will* interact with handwritten software in some way in the holistic system. This suggests the need to study the control software itself, how it changes [73], how it interacts with other libraries, how bugs propagate [118] and get fixed, how unmodeled aspects of the implementation are changed, likely by hand, to make the system "work." This means expanding the view of control software to be considered a part of the CPS in its own right [119]. A more complete view of aerospace control software might have established metrics by which we could show how robust a particular software controller is to bug fixes, changes in linked libraries, improper units, or computational exceptions (e.g., segmentation fault).

IV. Autonomy

Different definitions and notions of autonomy abound, and some form of *automation* has been part of aerospace systems for decades. To provide focus and scope in the context of CPS, as well as the relationship and significance of autonomy to other topics of this paper, we leverage a recent definition from [120], which asserts that autonomy involves self-governance and self-directed behavior, relying on "artificial intelligence (AI)-based capabilities that allow it to respond to situations that were not preprogrammed or anticipated in the design". The key term for this review is *artificial intelligence*, but because the term itself is fraught with ambiguity and dissension, we will instead focus on a related but arguably more clear term of *machine learning*. Furthermore, to provide scope to our review and analysis, we will borrow from a conceptual framework familiar to autonomous driving, which breaks the autonomy stack into (1) perception, (2) planning, and (3) control [121]. We will therefore review techniques that use machine learning in one of these three areas in the context of aerospace engineering applications. We will then extend the discussion and conclude with several of the unsolved challenges facing the aerospace CPS community, including architectures, safety, and multi-agent systems.

Perception Navigation in aerospace has typically been assisted with (human) vision in aviation, inertial measurement units, and ground-based infrastructure such as radar. The rise of neural network-based machine vision, coincident with the increasing number of operational use cases for unpiloted aerial systems (UAVs or UAS), has resulted in a new class of techniques in perception for aerospace CPS. Examples also abound in space exploration [122–124].

Using vision also as a source of redundancy with respect to other navigation and localization aides that are traditional to aerospace (e.g. GPS, radar) is a primary motivator in [125]. Leveraging a notion of synergy and redundancy is also significant in [126, 127], which builds multirotor onboard sensing platform that uses intertial measurement units and altitude sensors in concert with an RGB camera in order to perform a complex UAV landing task without any additional infrastructure. Achieving reliable relative inertial measurements, and vision-based relative pose updates in UAV navigation applications is the focus in a related line of work [128, 129]. Vision is used in a multi-agent setting in [130], where micro-aerial vehicles coordinate their motions in order to track a (ground) person's trajectory.

In addition to state estimation and navigation, image processing is leveraged in the context of search and rescue or related applications with UAVs [131]. To provide increased situational awareness in marine search and rescue, several approaches use neural network-enabled machine vision for detection of people [132]. Perception is particularly important in the detection and prediction of fires, an increasingly important and relevant UAV application [133, 134].

This discussion has so far focused on the use of neural networks for vision-based perceptions, particularly as this relates to UAV applications. As machine vision and perception performance continues to advance [135–138], these applications will grow but come with the attendant questions about the reliability, trustworthiness, and explainability of these neural network architectures. Furthermore, the notion of perception is not limited to just vision, and more comprehensive and integrative localization and sensing suites are both necessary and are advancing [139–142].

Planning The planning literature is rich, vast, and varied, and we again emphasize that we are primarily focused on learning-enabled planning as it applies to aerospace CPS. Reinforcement learning is increasingly being used in the context of planning, including applications to hopping rovers [143], planetary rovers [144], and other spacecraft GNC applications [145].

There is also a growing line of work using supervised learning and neural networks for planning. Trajectory generation and optimization is a focus in [146], where the output of a pre-trained neural network is used as an initializer for a numerical optimization scheme. In [147], a neural network generates motion primitives of trajectories for dynamically-feasible quadrotor trajectories, and recurrent neural networks are seeing increased use for path planning and collision avoidance applications [148, 149].

Much like the perception tasks, machine learning (and in particular, the use of neural network architectures) is seeing a rapid rise in planning applications; and like perception – and any other safety-critical task associated with machine learning – there are still open questions about reliability, trustworthiness, and explainability. As we will discuss momentarily, the power and expressiveness of these learning architectures lend themselves to increasingly complex applications such as multi-agent trajectory planning [150–152] and prediction [153, 154].

Control We refer the reader to Section III for a discussion of machine learning-based control but note that much of this work blurs the line in our distinction between perception, planning, and control. For example, several papers investigate closed-loop systems with neural networks in the loop, which span planning and control applications [155–157] but have clear applicability to aerospace CPS.

Challenges There is practically an unlimited set of open questions and future research directions when it comes to autonomy broadly, and machine-learning enabled autonomy more specifically. We mention just a few and note that many other directions are related to the other sections of this paper, for example human-autonomy teaming, safety, and of course controls. This review focuses on three challenges. First, what are the appropriate architectures for integrating and enabling machine learning in the context of aerospace CPS? Second, are there appropriate notions of risk when it comes to machines making their own decisions and exhibiting self-governance? Finally, how do we test such systems?

A typical paper in the machine learning literature will report performance in terms of training and testing splits. On a classification task, for example, exemplary performance might involve numbers such as 90% or even 99% accuracy. As we will discuss momentarily, even state-of-the-art performance levels could have disastrous consequences in safety-critical CPS. However, aviation has achieved one of the great safety records in engineering history not because the individual components of an aircraft (or the air traffic control system) are 100% reliable. Rather, the system is safe in large part due to the *architecture*; that is, the individual components achieve certain performance characteristics, but it is their interactions and interfaces that lead to the total system behavior. In autonomy, and in particular for machine learning, it is therefore vital to think about how and where a learning-enabled system might fit. Several works have put forth a notion of *composition*, where individual parts are analyzed are reasoned about, and then with careful attention to interfaces, a total system property can be obtained [158–161]. In addition, there are recent control architectures that explicitly try to separate out traditional mathematical models (e.g., state space control models) and black box learning models in a unified control framework [162–164]. Despite these and other attempts to frame autonomy as a potential question of systems architecture, it is still unclear where and when these learning-enabled systems should be.

Though safety and verification are discussed more thoroughly in Section VI, it is important to think about risk in a slightly different way as it relates to autonomy. If future systems are to be truly endowed with self-governance and high-level decision-making, it is important to assess whether the systems themselves have the ability to reason about risk. In the context of robotics, recent work tries to develop appropriate risk metrics [165] or constrain the learning system itself according to risk [166, 167]. Other work attempts to consider risk at operation time [168]. The question remains: will these systems exhibit the same behavior towards risk as their human counterparts? This question then gives rise to whether they should; though somewhat philosophical in nature, such questions very quickly come to bear on engineering techniques and methods.

Testing has been a vital aspect of assuring aerospace systems for decades. However, many of the tasks being proposed in the autonomy literature have historically been performed by humans. Indeed, humans get "tested", but this type of testing is both qualitatively and quantitatively different than how we have historically tested software [169, 170]. Furthermore, the way that machine learning is currently tested – in the sense of train / validate / test splits [171, 172] – is likely inadequate for safety- or mission-critical aerospace CPS. For example, an even 96% test accuracy in terms of precision or recall may result in a catastrophic number of accidents or lives lost. This suggests the urgency that machine learning components should be architected within the broader systems context. Recent results suggest different ways of testing neural networks in more rigorous, transparent ways [173–176]. However, much more work is needed, not only in terms of techniques and metrics of testing machine learning algorithms, but also towards an overarching paradigm for test and validation of autonomous systems particularly in the context of aerospace CPS.

V. Humans Working With Cyber-Physical Aerospace Systems

Humans, and their activities, are integral to aerospace CPS in two ways. First, the definitions of *automation* and *autonomy* frame the machine as performing functions normally attributed to humans. Thus, the human operator's activities have been, and continue to be, the central point of definition of operations that CPS contribute to. While machine capabilities have increased over the century since the first demonstration of automatic control of an aircraft, to the point that air transport aircraft now can, in theory, almost completely fly themselves, this automatic functioning is allowed as long as nothing fails or changes in the intended route of flight, at which point the pilot needs to intervene (which is estimated to occur in roughly 90% of flights) [177].

Second, whenever CPS systems are assigned functions, the human operator is impacted. While no longer needing to execute some functions, the human operator still needs to perform any that haven't been automated. This leads to several concerns [178]. For example, while a higher-level of automation suggests that the machine is taking on more importance in system functioning, the human may consistently override or second-guess high levels of automation; this both lowers the actual operational value of the automation, and can be intensive for the human in terms of effort and information required. Likewise, so-called lower levels of automation that alert or suggests decision may effectively dictate a course of action to the human [179, 180], driving the system behavior even if the human is the actuator.

Function allocations driven by what the machine can do implicitly apply a function allocation strategy termed by Bailey as leftover allocation: automate as many functions as technology will permit, and assume the human will pick up which-ever functions are leftover [181]. In aerospace, commonly automated functions drive standard processes within clearly-defined conditions. However, this provides little support in off-nominal conditions, which is generally when the human needs the most support [182]. In such cases, the number of required human operators is no longer driven by workload in nominal conditions; instead, following the description of modern pilotage as "hours/weeks/months of boredom interrupted by seconds of sheer terror," the number of human operators required (somewhere, connected to the vehicle somehow) is driven by the worst-case, when they must be able to detect and take-over situations beyond the machine's capabilities.

Historically, humans have been framed as being better at cognitive tasks, and machines at the higher-bandwidth sensing-control-actuation type tasks, based on early attempts to frame what "Men are better at / machines are better at" [183]. This has been an explicit or implicit justification for a supervisory control paradigm, in which the human is expected to perform fairly cognitive tasks based on abstraction and problem solving, framed by Sheridan as principally serving as inputs to the computer/machine to determine its control activity [184]. This substantially changes the human's task from "needing to know the vehicle dynamics well enough to control them" – a skill-based task easily transferable between aircraft – to "needing to know the details of how the automation/autonomy works sufficient to program, command, confirm and intervene in its activities" – which, outside normal operations, can be an intensive task requiring knowledge specific to that piece of autonomy and not transferable to other systems or situations.

Typically, discussions of function allocation only consider which agent within a human-machine team will execute which action – here, defined as the authority for an action. However, the human remains legally responsible for the outcome. Thus, as automation and autonomy take on more, the human's role in a supervisory control paradigm is arguably harder. The human's tasks are "monitor" and "intervene," even as s/he has been pushed out of the sensing-control-actuation performance of the task. This demands that the operator do something that may look easy and passive given its lack of externally-observable activity: analyze whether the machine is doing the right activity by monitoring. This monitoring needs to occur frequently enough that it can catch any problem within the performance of the task quickly enough for successful intervention – for continuous control tasks such as autoflight, this monitoring needs to occur almost as continuously. This monitoring may need to consider all the inputs used by the machine as well as its outputs – the monitoring may trigger on simple determinations of machine actions that look to be out of range, but it may also need to perform a parallel assessment of what the operator thinks the actions should be, or even reverse-engineer why the machine outputs are being executed and decide whether that implied machine process model is correct. Only recently are studies starting to analyze the demands of this monitoring, and the effort, information and knowledge it requires [185–187].

Thus, in aerospace the decision of whether to engage and use the machine should not be viewed merely as a matter of "trusting" it: with aerospace's high standards for safety, a system that is correct 90% of the time, or 99% of the time, needs to be monitored and corrected for. Instead, human operators need to "trust but verify" at all times. Without a concrete basis for assessing if the automation is correct, humans often over- and under-trust the automation; either way, incorrect trust is viewed as human error, despite its basis in the function allocation [188]. Further, a human's operator choice to rely on the machine also considers the "interaction overhead" associated with programming it, invoking it and, in when thinking ahead to potential interventions, the difficulty and delay in intervening. Thus, a human may invoke a system even when they do not believe in its functioning if it is easy to command, monitor and intervene; conversely, a human operator may not invoke a system that they believe is 99% correct if it is hard to command and the difficulty and delay in intervening can be safety-critical.

It is common now to describe intelligent machines, such as those sought by CPS, as members of human-machine teams, an extension from the supervisory control one-human/one-machine model. However, where autonomy is viewed as a team member, its ability to communicate with others, in a form comprehensible to them, is of significant interest. This transcends the autonomy being "explainable" in a passive sense by an active human operator: It is important that team members are able to anticipate each other's information needs and provide information at useful, non-interruptive times [189, 190]. Timing must be considered as well: inter-leaved tasks can leave one agent waiting on another, and poorly timed communication can disrupt performance. An interruption may be demanded by circumstances, in which case it can spur knowledge acquisition [191] and facilitate decision-making performance [192]. Unfortunately, too often automation is "clumsy:" it unduly interrupts its human team members because, whereas humans can implicitly sense information about whether other team members would benefit from an interruption, automation historically cannot [193], for an example of efforts to improve this aspect of automation). With human-automated teams, predefined sets of function allocations may serve as more explicit coordination strategies, such as the playbook proposed by Miller and Parasuraman [194] and such as the function allocations that the pilot of a modern aircraft may invoke. Likewise, roles for the autonomy such as critiquing systems that monitor the operation and only interrupt if the human operator's activities or judgements do not match models of "good performance" can define effective machine roles [195].

Finally, the literature in cognitive engineering also examines the meta-cognitive activities of humans in safety-critical systems such as aerospace. These operations require the choice of which activities to perform when, balancing many simultaneous actions. Dynamic analysis is often required to identify situations where the interleaving of functions assigned to disparate agents requires significant co-ordination or idling as one waits on another, or where workload may accumulate, or where one agent will be unduly interrupting another.

Proposed notions of resilience describe a team's ability to work in unexpected situations as maintaining control of, or regulating those situations [196–198]. Resilience enables organizations to cope with the unexpected in a "robust yet flexible" manner [199]. However, the brittleness of machine functions (i.e., safe operation only within some defined set of operating conditions defined, for example, by the bounds of the control authority or the training set used in machine learning) reflects how allocation of key functions to the machine may degrade the team's resilience in unexpected situations. Likewise, resilience is fostered when a human agent may select strategies (courses of action) appropriate to the state of the environment and their own capabilities [200]. For example, Hollnagel's concept of cognitive control describes how humans adapt their activities (and sequence them) in response to their competency and their perception of resources available to them (such as information availability) and demands on them (such as subjective available time) [201, 202]. However, such adaptation can be constrained where the machine in a human-machine team dictates a specific sequence of activities from the human [203]. The adverse effects of such overly prescribed function allocations have been found to manifest in work-arounds or dis-use of machines [188, 200, 204].

These difficulties highlight the value of understanding the complete set of human activities in aerospace operations when designing CPS. Such understanding can serve several purposes. First, it can inform the design of important machine functions to better imitate what the human does. Second, it can highlight which machine functions would truly support the operations, versus which would potentially disrupt. Third, it highlights the need for machine behaviors that not only act independently – autonomously – but also interact with their team, including a human supervisor who is legally responsible and other agents outside their immediate vehicle, such as air traffic control or command and control.

Challenges Several important questions must be answered when fielding CPS into quintessentially human-driven operations – and CPS research and developments needs to be steered to ensure these answers are available. The first recognizes that CPS are envisioned to take over functions (or type of functions) normally attributed to humans: So, what-all do the humans actually do when performing these functions? This section has articulated a wide-range of activities normally attributed to humans. Some, commonly called the "taskwork," are obvious and clear-cut, but others are typically described as implicit or intangible: the meta-cognitive activities of constantly predicting and organizing the entire set of activity that is occurring, of re-designing the team allocation, and of preparing for likely future activities.

Second, no agent truly gets to act "autonomously" in aerospace operations: all must coordinate their activities with others, sometimes loosely and sometimes tightly integrating their activities with others. This section has articulated a wide range of teamwork activities. Unfortunately, machines are too often not good team players; CPS systems must not only understand how to perform their functions, but also how to coordinate these functions with the other agents' activities, and how (and when) to communicate about them in a manner useful to others in the operation who should not need to learn inner details of the CPS to understand its impact on the operation.

Finally, the choice to give CPS certain functions impacts others, who must adjust their pattern of activity around them. CPS should not be developed just around which functions it is easy or convenient for the technology designers, but instead need to be purposefully be constructed towards a role that supports the entire team. This criteria is particularly profound when a human operator is responsible for the outcome of the entire operation. In this case, it is advisable to the human to *not* employ CPS whose functions may not be appropriate to the situation, or where the difficulty or risk in commanding, programming and invoking the CPS – and then needing to monitor and potentially correct or intervene the functions of the CPS – are not worth the effort and disruption.

VI. Safety & Verification

Verification of CPS requires an integrated approach that weaves together a safety case from both traditional and formal verification techniques. The U.S. FAA, for example, documents verification requirements for flight certification in DO-178B [205], DO-178C [206], DO-333 [207], and DO-254 [208]. Intuitively, we need to *validate* that our system design meets the objectives of the given operational concept, and validate the set of system requirements, e.g., so that we know they are internally consistent, faithfully represent properties like safety and security, and sufficiently cover the range of system behaviors. We then need to use the validated system design and requirements in our effort to *verify* that the system does what we think it should do *and nothing else*. For example, we can verify that a system sufficiently often produces an acceptable output given certain inputs using traditional verification techniques like testing and simulation.

For another example, we can verify that our system does not spontaneously exhibit an unsafe behavior (i.e., exhibit the behavior on any input or no input at all) using formal verification techniques like model checking and theorem proving. Our objective in the end is to produce a cohesive argument or *safety case* for flight certification.

Highlights of Formal Methods in Autonomous Aerospace Systems Though they are still not as broadly understood or widely adopted as traditional verification techniques, formal methods have significantly impacted all areas of autonomous aerospace systems. Often, specific challenges from the development of new aerospace systems have served as the impetus for new areas of formal methods research and development. We highlight a few examples of advancements in formal methods driven by developments in aerospace autonomy.

Model Checking contributed to the early-design-stage of NASA Ames' Automated Airspace Concept [209], including finding rare scenarios where aircraft could execute commands that were not from their current controller [210], and compare multiple design options to prove which design parameters provide the lowest probability of an unresolved conflict between aircraft [211]. The impact of formal analysis of one, initial, partial design led to the challenge of formal design space exploration: comparing several possible system designs that differ in terms of design choices, capabilities, and implementations. We can apply formal techniques ranging from model checking to model-based fault-tree analysis to rigorously analyze the safety of different design solutions, comparing how different functional allocations impact the overall reliability of the automated air traffic control system [212]. Next, we faced the challenge of considering a design space with more than 20,000 designs for the NextGen air traffic control system. We introduced a compositional, modular, parameterized approach combining model checking with contract-based design to automatically generate large numbers of models from a possible set of components and their implementations, identifying novel as well as known problematic configurations [213]. The need to scale model checking to explore large design spaces faster drove the creation of a new algorithm, FuseIC3, learning and reusing information from solving related models to check the design space of 20K models over five times faster than before [214]. Utilizing logical techniques to discover design-space dependencies and dynamically order intermediate analysis steps sped up the verification of NASA's full design space by another 9.4x [215]. We have continued advancing algorithms for scalable design-space model checking, with impact both on automated air traffic control and on industrial CPS analysis, such as at IBM [216].

Runtime Verification (RV) can efficiently analyze the current mission, checking adherence to mission safety parameters in real time, on-board, in order to account for, e.g., safety violations caused by unexpected environmental occurances or other off-nominal conditions. Like simulation, RV analyzes single runs of the system instead of the entire design space, but RV uses formal verification techniques to provide real-time checks of complex-temporal properties while the system is operating [218]. The need to assure automated flights of NASA's wildfire surveillance aircraft, the Swift UAS drove the creation of the Realizable, Responsive, Unobtrusive Unit, an RV engine designed to satisfy those verification parameters [219, 220]. Unexpected emergent behavior by Robonaut2 on the International Space Station inspired a novel revision of RV via R2U2 in order to disambiguate between failure modes using only the space left over in the FPGA controlling Robonaut2's knee joint [221]. After an extensive survey of all currently-available verification tools, NASA's Lunar Gateway Vehicle System Manager (VSM) team selected R2U2 for operational verification [222]. The need to improve usability and trust in this real-time verification of a large-scale, complex safety-critical system drove advancements including easy-to-validate methods for set-based reasoning, transparent representations of assume-guarantee-contract requriements, and automated checks to answer the question of how do we check the checker [223].

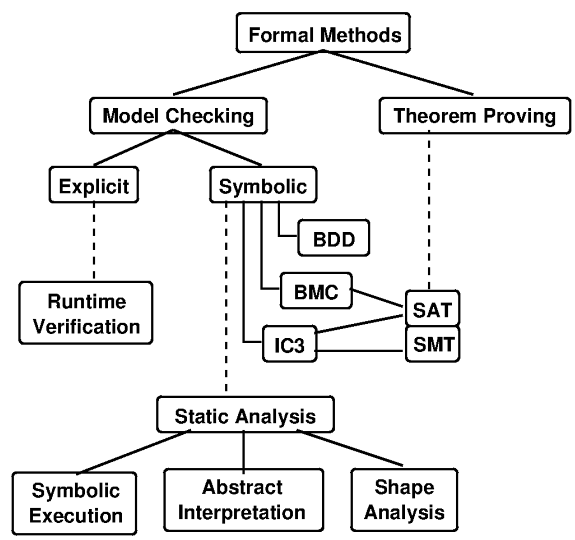

Fig. 5 The (growing) tree of formal methods; solid lines represent direct variations whereas dashed lines represent related derivations [217].

Challenges While traditional verification techniques benefit from broad understanding and adoption in the final stages of system design, CPS require a more integrated assurance approach incorporating both explicit validation and verification, both traditional and formal verification techniques, and application of assurance techniques across all stages of the system life-cycle. This need lays the groundwork for future challenges in safety and verification of autonomous aerospace systems. Recently, [217] defined a framework for certification of future reliable autonomous systems; we highlight some elements of that framework along with other aerospace-specific challenges here.

One major barrier to adoption of integrated CPS V&V techniques stems from the success of the "V-model" of system engineering. This "V-model" lays out the recommended stages of system design in the shape of the letter "V" with verification appearing exclusively as the last step of the design, and consisting only of traditional techniques. While this model may be ideal for the design of *physical* systems, where verification does not require formal methods and where testing does not make sense until there is a completed structure capable of responding to a test, it does not apply to software. As aerospace systems evolved into autonomous *cyber-physical* systems, the "V-model" no longer applies. Therefore, we need to address the challenge of social barriers to adoption of state-of-the-art CPS V&V techniques at every stage of the system lifecycle, from early system design to system runtime and maintenance. This challenge involves broad, industry-wide adoption of techniques including initial requirements validation, using model checking to check early-design-stage partial (not yet complete) designs for safety properties, generating software implementations from formally-verified designs, and incorporating on-board runtime verification for real-time checks that the current mission upholds safety requirements.

Technical challenges to safety and verification of autonomous aerospace systems include developing new formal methods to address verification needs of new types of (sub-)systems, scaling existing techniques to be capable of reasoning about ever-more-complex CPS, and automating the application of verification tools and techniques to make their use feasible. For example the Achilles heel of model checking is the state-space explosion problem: because model checking exhaustively checks all possible system behaviors to provide a provable assurance claim, analyzing a system with a very large behavior space can take an unacceptable amount of time, or elude the model checker entirely. New algorithms, heuristics, compositional verification techniques, and other technological advances can address this challenge; these are all active areas of research. For another example, applying rigorous verification techniques early in the design lifecycle requires significant time and expertise, but this investment can massively benefit later design stages if we design automation to enable this. Early design stage formal verification can additionally contribute to automated test-case generation and automated generation of verified software implementations, thus saving the system designers from performing these tasks manually while providing a higher level of assurance.

There is also a set of challenges that bridges the gap between social and technical challenges. Human system designers, engineers, and certification authorities need to be able to understand and trust formal methods: how do we both train engineers to trust the outcomes of formal verification and grow formal verification tools to transparently explain the reasoning behind their proofs? Explainability of formal methods remains an active research area, as do the related areas of automated certificate generation for model checking, and automated validation of formal specifications of system requirements. There is also a large community researching educational techniques, such as how to train engineers to think formally, design with a verification mindset, and deeply understand what different V&V techniques prove, and don't prove about the system under test [224]. This is particularly challenging in the context of aerospace engineering [225].

Building off of these social and technical challenges as well as the discussion of humans in aerospace CPS (section V), we must not only ask where does the machine fit in safety-critical operations. We must also ask how to then assure the safety of such human-autonomy teams? In addition to formal verification, we need more advanced hazard analysis capabilities and system safety methods. One such approach, called Systems-theoretic Process Analysis, or STPA [226], has shown promise in aerospace CPS applications because it can help assess unsafe decision-making as it relates to system state, and feedback about that system state. Other work from the system safety community explicitly analyzes the coordination (or lack thereof) between multiple agents within a system [227–229]. Pointing to the earlier discussion of the V-model and the historically late introduction of verification, these approaches provide a way to identify system architectures from a safety-driven perspective, particularly when the 'architectural elements' have decision-making or autonomous capability.

VII. Conclusion

The Cyber-Physical Systems community and its outgrowths, though relatively new, have already had significant impact on the aerospace community. CPS has sought to create a science of systems [230] and have addressed many facets of modern complex vehicles, most prominently, real-time systems and control, humans in CPS, autonomy, and safety, validation, and verification. Here we have focused on synthesizing the impact of the CPS research community on those key areas and identified areas for future growth and exploration.

Aerospace vehicles are becoming more complex, incorporating machine-learned algorithms, and navigating an increasingly crowded and uncertain national airspace. To meet the challenge of deploying safe, secure, reliable aerospace vehicles the community will need to continue to embrace new ideas from related areas and incorporate the best ideas into our community.

Acknowledgments

This work supported in part by NSF-2047971 and NASA 80NSSC19K1702.

References

[1] Lee, E. A., "Fundamental Limits of Cyber-Physical Systems Modeling," *ACM Transactions on Cyber-Physical Systems*, Vol. 1, No. 1, 2016, pp. 3:1–3:26. https://doi.org/10.1145/2912149, URL https://doi.org/10.1145/2912149.

[2] Wiener, N., *Cybernetics or Control and Communication in the Animal and the Machine*, Vol. 25, MIT press, 1965.

[3] Bradley, J., "Toward Co-Design of Autonomous Aerospace Cyber-Physical Systems," Ph.D. thesis, University of Michigan, 2014.

[4] Lee, E. A., and Seshia, S. A., *Introduction to Embedded Systems: A Cyber-Physical Systems Approach*, Mit Press, 2016.

[5] Ding, D., Han, Q.-L., Ge, X., and Wang, J., "Secure State Estimation and Control of Cyber-Physical Systems: A Survey," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 51, No. 1, 2021, pp. 176–190. https://doi.org/10.1109/TSMC.2020.3041121.

[6] Yaacoub, J.-P. A., Salman, O., Noura, H. N., Kaaniche, N., Chehab, A., and Malli, M., "Cyber-Physical Systems Security: Limitations, Issues and Future Trends," *Microprocessors and Microsystems*, Vol. 77, 2020, p. 103201. https://doi.org/10.1016/j.micpro.2020.103201, URL https://www.sciencedirect.com/science/article/pii/S0141933120303689.

[7] Wolf, M., and Serpanos, D., "Safety and Security in Cyber-Physical Systems and Internet-of-Things Systems," *Proceedings of the IEEE*, Vol. 106, No. 1, 2018, pp. 9–20.

[8] Humayed, A., Lin, J., Li, F., and Luo, B., "Cyber-Physical Systems Security—A Survey," *IEEE Internet of Things Journal*, Vol. 4, No. 6, 2017, pp. 1802–1831. https://doi.org/10.1109/JIOT.2017.2703172.

[9] Giraldo, J., Sarkar, E., Cardenas, A. A., Maniatakos, M., and Kantarcioglu, M., "Security and Privacy in Cyber-Physical Systems: A Survey of Surveys," *IEEE Design & Test*, Vol. 34, No. 4, 2017, pp. 7–17. https://doi.org/10.1109/MDAT.2017.2709310.

[10] Wu, G., Sun, J., and Chen, J., "A Survey on the Security of Cyber-Physical Systems," *Control Theory and Technology*, Vol. 14, No. 1, 2016, pp. 2–10.

[11] Sun, C.-C., Liu, C.-C., and Xie, J., "Cyber-Physical System Security of a Power Grid: State-of-the-Art," *Electronics*, Vol. 5, No. 3, 2016, p. 40. https://doi.org/10.3390/electronics5030040, URL https://www.mdpi.com/2079-9292/5/3/40.

[12] Shafi, Q., "Cyber Physical Systems Security: A Brief Survey," *2012 12th International Conference on Computational Science and Its Applications*, 2012, pp. 146–150. https://doi.org/10.1109/ICCSA.2012.36.

[13] Rathore, H., Mohamed, A., and Guizani, M., "A Survey of Blockchain Enabled Cyber-Physical Systems," *Sensors*, Vol. 20, No. 1, 2020, p. 282. https://doi.org/10.3390/s20010282, URL https://www.mdpi.com/1424-8220/20/1/282.

[14] Bodkhe, U., Mehta, D., Tanwar, S., Bhattacharya, P., Singh, P. K., and Hong, W.-C., "A Survey on Decentralized Consensus Mechanisms for Cyber Physical Systems," *IEEE Access*, Vol. 8, 2020, pp. 54371–54401. https://doi.org/10.1109/ACCESS.2020.2981415.

[15] Graja, I., Kallel, S., Guermouche, N., Cheikhrouhou, S., and Hadj Kacem, A., "A Comprehensive Survey on Modeling of Cyber-Physical Systems," *Concurrency and Computation: Practice and Experience*, Vol. 32, No. 15, 2020, p. e4850. https://doi.org/10.1002/cpe.4850, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.4850.

[16] Roehm, H., Oehlerking, J., Woehrle, M., and Althoff, M., "Model Conformance for Cyber-Physical Systems: A Survey," *ACM Transactions on Cyber-Physical Systems*, Vol. 3, No. 3, 2019, pp. 30:1–30:26. https://doi.org/10.1145/3306157, URL https://doi.org/10.1145/3306157.

[17] Ding, D., Han, Q.-L., Wang, Z., and Ge, X., "A Survey on Model-Based Distributed Control and Filtering for Industrial Cyber-Physical Systems," *IEEE Transactions on Industrial Informatics*, Vol. 15, No. 5, 2019, pp. 2483–2499. https://doi.org/10.1109/TII.2019.2905295.

[18] Bartocci, E., Deshmukh, J., Donzé, A., Fainekos, G., Maler, O., Ničković, D., and Sankaranarayanan, S., "Specification-Based Monitoring of Cyber-Physical Systems: A Survey on Theory, Tools and Applications," *Lectures on Runtime Verification: Introductory and Advanced Topics*, edited by E. Bartocci and Y. Falcone, Lecture Notes in Computer Science, Springer International Publishing, Cham, 2018, pp. 135–175. https://doi.org/10.1007/978-3-319-75632-5_5, URL https://doi.org/10.1007/978-3-319-75632-5_5.

[19] Lee, E. A., "The past, present and future of cyber-physical systems: A focus on models," *Sensors*, Vol. 15, No. 3, 2015, pp. 4837–4869. https://doi.org/10.3390/s150304837.

[20] Tamblyn, S., Henry, J., and King, E., "A Model-Based Design and Testing Approach for Orion GN #x00026;C Flight Software Development," *2010 IEEE Aerospace Conference*, 2010, pp. 1–12. https://doi.org/10.1109/AERO.2010.5446802.

[21] Erkkinen, T., and Potter, B., "Model-Based Design for DO-178B with Qualified Tools," *AIAA Modeling and Simulation Technologies Conference and Exhibit*, 2009.

[22] Aarenstrup, R., *Managing Model-Based Design*, The MathWorks, 2015. URL https://www.mathworks.com/content/dam/mathworks/mathworks-dot-com/solutions/model-based-design/managing-model-based-design/MBD_Book_PDF_Version.pdf.

[23] Porter, J., Karsai, G., Völgyesi, P., Nine, H., Humke, P., Hemingway, G., Thibodeaux, R., and Sztipanovits, J., "Towards Model-Based Integration of Tools and Techniques for Embedded Control System Design, Verification, and Implementation." *MoDELS Workshops*, Springer, 2008, pp. 20–34. URL http://link.springer.com/content/pdf/10.1007/978-3-642-01648-6.pdf#page=30.

[24] Zacchia Lun, Y., D'Innocenzo, A., Smarra, F., Malavolta, I., and Di Benedetto, M. D., "State of the Art of Cyber-Physical Systems Security: An Automatic Control Perspective," *Journal of Systems and Software*, Vol. 149, 2019, pp. 174–216. https://doi.org/10.1016/j.jss.2018.12.006, URL https://www.sciencedirect.com/science/article/pii/S0164121218302681.

[25] Guan, X., Yang, B., Chen, C., Dai, W., and Wang, Y., "A Comprehensive Overview of Cyber-Physical Systems: From Perspective of Feedback System," *IEEE/CAA Journal of Automatica Sinica*, Vol. 3, No. 1, 2016, pp. 1–14. https://doi.org/10.1109/JAS.2016.7373757.

[26] Bradley, J., and Atkins, E., "Optimization and Control of Cyber-Physical Vehicle Systems," *Sensors*, Vol. 15, No. 9, 2015, p. 23020. https://doi.org/10.3390/s150923020, URL http://www.mdpi.com/1424-8220/15/9/23020.

[27] Xu, L. D., and Duan, L., "Big Data for Cyber Physical Systems in Industry 4.0: A Survey," *Enterprise Information Systems*, Vol. 13, No. 2, 2019, pp. 148–169. https://doi.org/10.1080/17517575.2018.1442934, URL https://doi.org/10.1080/17517575.2018.1442934.

[28] Atat, R., Liu, L., Wu, J., Li, G., Ye, C., and Yang, Y., "Big Data Meet Cyber-Physical Systems: A Panoramic Survey," *IEEE Access*, Vol. 6, 2018, pp. 73603–73636. https://doi.org/10.1109/ACCESS.2018.2878681.

[29] Molina, E., and Jacob, E., "Software-Defined Networking in Cyber-Physical Systems: A Survey," *Computers & Electrical Engineering*, Vol. 66, 2018, pp. 407–419. https://doi.org/10.1016/j.compeleceng.2017.05.013, URL https://www.sciencedirect.com/science/article/pii/S0045790617313368.

[30] Chaâri, R., Ellouze, F., Koubâa, A., Qureshi, B., Pereira, N., Youssef, H., and Tovar, E., "Cyber-Physical Systems Clouds: A Survey," *Computer Networks*, Vol. 108, 2016, pp. 260–278. https://doi.org/10.1016/j.comnet.2016.08.017, URL https://www.sciencedirect.com/science/article/pii/S1389128616302699.

[31] Abbaspour Asadollah, S., Inam, R., and Hansson, H., "A Survey on Testing for Cyber Physical System," *Testing Software and Systems*, edited by K. El-Fakih, G. Barlas, and N. Yevtushenko, Springer International Publishing, Cham, 2015, pp. 194–207. https://doi.org/10.1007/978-3-319-25945-1_12.

[32] Sadri-Moshkenani, Z., Bradley, J., and Rothermel, G., "Survey on Test Case Generation, Selection and Prioritization for Cyber-Physical Systems," *Software Testing, Verification and Reliability*, Vol. 32, No. 1, 2022, p. e1794. https://doi.org/10.1002/stvr.1794, URL https://onlinelibrary.wiley.com/doi/abs/10.1002/stvr.1794.

[33] Dey, N., Ashour, A. S., Shi, F., Fong, S. J., and Tavares, J. M. R. S., "Medical Cyber-Physical Systems: A Survey," *Journal of Medical Systems*, Vol. 42, No. 4, 2018, p. 74. https://doi.org/10.1007/s10916-018-0921-x, URL https://doi.org/10.1007/s10916-018-0921-x.

[34] Cheng, A. M. K., "Cyber-Physical Medical and Medication Systems," *Distributed Computing Systems Workshops, 2008. ICDCS'08. 28th International Conference On*, IEEE, 2008, pp. 529–532.

[35] Lee, I., and Sokolsky, O., “Medical cyber physical systems,” *Design Automation Conference (DAC), 2010 47th ACM/IEEE*, IEEE, 2010, p. 743–748.

[36] Cintuglu, M. H., Mohammed, O. A., Akkaya, K., and Uluagac, A. S., “A Survey on Smart Grid Cyber-Physical System Testbeds,” *IEEE Communications Surveys & Tutorials*, Vol. 19, No. 1, 2017, pp. 446–464. https://doi.org/10.1109/COMST.2016.2627399.

[37] Khaitan, S. K., and McCalley, J. D., “Cyber Physical System Approach for Design of Power Grids: A Survey,” *2013 IEEE Power & Energy Society General Meeting*, 2013, pp. 1–5. https://doi.org/10.1109/PESMG.2013.6672537.

[38] Karnouskos, S., “Cyber-physical systems in the smartgrid,” *Industrial Informatics (INDIN), 2011 9th IEEE International Conference on*, IEEE, 2011, p. 20–23.

[39] Marwedel, P., Mitra, T., Grimheden, M. E., and Andrade, H. A., “Survey on Education for Cyber-Physical Systems,” *IEEE Design & Test*, Vol. 37, No. 6, 2020, pp. 56–70. https://doi.org/10.1109/MDAT.2020.3009613.

[40] Committee on 21st Century Cyber-Physical Systems Education, Computer Science and Telecommunications Board, Division on Engineering and Physical Sciences, and National Academies of Sciences, Engineering, and Medicine, *A 21st Century Cyber-Physical Systems Education*, National Academies Press, Washington, D.C., 2016. https://doi.org/10.17226/23686, URL https://www.nap.edu/catalog/23686.

[41] Lee, E. A., “Cyber physical systems: Design challenges,” *Object Oriented Real-Time Distributed Computing (ISORC), 2008 11th IEEE International Symposium on*, IEEE, 2008, p. 363–369.

[42] Wolf, W., “Cyber-Physical Systems,” *Computer*, Vol. 42, No. 3, 2009, pp. 88–89.

[43] Rajkumar, R. R., Lee, I., Sha, L., and Stankovic, J., “Cyber-Physical Systems: The next Computing Revolution,” *Proceedings of the 47th Design Automation Conference*, ACM, 2010, pp. 731–736.

[44] Shi, J., Wan, J., Yan, H., and Suo, H., “A Survey of Cyber-Physical Systems,” *2011 International Conference on Wireless Communications and Signal Processing (WCSP)*, 2011, pp. 1–6. https://doi.org/10.1109/WCSP.2011.6096958.

[45] Klesh, A. T., Cutler, J. W., and Atkins, E. M., “Cyber-Physical Challenges for Space Systems,” *Cyber-Physical Systems (ICCPS), 2012 IEEE/ACM Third International Conference on*, 2012, p. 45–52. https://doi.org/10.1109/ICCPS.2012.13.

[46] Kim, K.-D., and Kumar, P. R., “Cyber–Physical Systems: A Perspective at the Centennial,” *Proceedings of the IEEE*, Vol. 100, No. 13, 2012, pp. 1287–1308.

[47] Lee, I., Sokolsky, O., Chen, S., Hatcliff, J., Jee, E., Kim, B., King, A., Mullen-Fortino, M., Park, S., Roederer, A., and Others, “Challenges and Research Directions in Medical Cyber–Physical Systems,” *Proceedings of the IEEE*, Vol. 100, No. 1, 2012, pp. 75–90.

[48] Sztipanovits, J., Ying, S., Cohen, I., Corman, D., Davis, J., Khurana, H., Mosterman, P. J., Prasad, V., and Stormo, L., “Strategic R&D Opportunities for 21st Century Cyber-Physical Systems,” *Steeting Commitee for the Foundation fo Innovation in Cyber-Physical Systems, National Institute of Standards and Technology (NIST)*, 2013.

[49] Gunes, V., Peter, S., Givargis, T., and Vahid, F., “A Survey on Concepts, Applications, and Challenges in Cyber-Physical Systems,” *KSII Transactions on Internet and Information Systems (TIIS)*, Vol. 8, No. 12, 2014, pp. 4242–4268. https://doi.org/10.3837/tiis.2014.12.001, URL https://www.koreascience.or.kr/article/JAKO201403760397435.page.

[50] PCAST 2011, “Ensuring American Leadership in Advanced Manufacturing,” Tech. rep., President’s Council of Advisors on Science and Technology (PCAST), Executive Office of the President, Jun. 2011. URL http://www.whitehouse.gov/sites/default/files/microsites/ostp/pcast-advanced-manufacturing-june2011.pdf.

[51] PCAST 2012, “Report to the President on Capturing Domestic Competitive Advantage in Advanced Manufacturing,” Tech. rep., President’s Council of Advisors on Science and Technology (PCAST), Executive Office of the President, Jul. 2012. URL http://www.whitehouse.gov/sites/default/files/microsites/ostp/pcast_amp_steering_committee_report_final_july_27_2012.pdf.

[52] PCAST 2013, “Report to the President Immediate Opportunities for Strengthening the Nation’s Cybersecurity,” Tech. rep., President’s Council of Advisors on Science and Technology (PCAST), Executive Office of the President, Nov. 2013. URL http://www.whitehouse.gov/sites/default/files/microsites/ostp/PCAST/pcast_cybersecurity_nov-2013.pdf.

[53] Samad, T., and Balas, G., *Software-Enabled Control: Information Technology for Dynamical Systems*, John Wiley & Sons, 2003.

[54] Johnson, E., Schrage, D., and Vachtsevanos, G., "Software Enabled Control Experiments with University Operated Unmanned Aircraft," *Infotech@ Aerospace*, 2005, p. 6956.

[55] Singh, L., Plump, J., McConley, M., and Appleby, B., "Software Enabled Control: Autonomous Agile Guidance and Control Synthesis for a UAV in Partially Unknown Urban Terrain," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2003, p. 5674.

[56] Schrage, D. P., and Vachtsevanos, G., "Software-Enabled Control for Intelligent UAVs," *Proceedings of the 1999 IEEE International Symposium on Computer Aided Control System Design (Cat. No. 99TH8404)*, IEEE, 1999, pp. 528–532.

[57] Doman, D., Ngo, A., Leggett, D., Saliers, M., and Pachter, M., "Development of a Hybrid Direct-Indirect Adaptive Control System for the X-33," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2000, p. 4156.

[58] Doman, D. B., and Ngo, A. D., "Dynamic Inversion-Based Adaptive/Reconfigurable Control of the X-33 on Ascent," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 2, 2002, pp. 275–284.

[59] Johnson, E., Calise, A., El-Shirbiny, H., and Eysdyk, R., "Feedback Linearization with Neural Network Augmentation Applied to X-33 Attitude Control," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2000, p. 4157.

[60] Stevens, B. L., Lewis, F. L., and Johnson, E. N., *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, John Wiley & Sons, 2015.

[61] Alur, R., *Principles of Cyber-Physical Systems*, MIT Press, 2015.

[62] Alur, R., Courcoubetis, C., Henzinger, T. A., and Ho, P.-H., *Hybrid Automata: An Algorithmic Approach to the Specification and Verification of Hybrid Systems*, Springer-Verlag, 1993.

[63] Engell, S., Frehse, G., and Schnieder, E., *Modelling, Analysis, and Design of Hybrid Systems*, Springer, 2002.

[64] Casau, P., Cabecinhas, D., and Silvestre, C., "Hybrid Control Strategy for the Autonomous Transition Flight of a Fixed-Wing Aircraft," *IEEE Transactions on control systems technology*, Vol. 21, No. 6, 2012, pp. 2194–2211.

[65] Zelinski, S., Koo, T. J., and Sastry, S., "Hybrid System Design for Formations of Autonomous Vehicles," *Decision and Control, 2003. Proceedings. 42nd IEEE Conference On*, Vol. 1, IEEE, 2003, pp. 1–6.

[66] Tomlin, C., Pappas, G. J., and Sastry, S., "Conflict Resolution for Air Traffic Management: A Study in Multiagent Hybrid Systems," *IEEE Transactions on automatic control*, Vol. 43, No. 4, 1998, pp. 509–521. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=664154.

[67] Lygeros, J., Tomlin, C., and Sastry, S., "On Controller Synthesis for Nonlinear Hybrid Systems," *Decision and Control, 1998. Proceedings of the 37th IEEE Conference On*, Vol. 2, IEEE, 1998, pp. 2101–2106.

[68] Luckcuck, M., Farrell, M., Sheridan, O., and Monahan, R., "A Methodology for Developing a Verifiable Aircraft Engine Controller from Formal Requirements," *2022 IEEE Aerospace Conference (AERO)*, IEEE, 2022, pp. 1–12.

[69] Ferretto, D., "Towards a Formalized Model-Based Process for the Design of High-Speed Aircraft and Related Subsystems," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 2022, p. 09544100221141942.

[70] Bozzano, M., Bruintjes, H., Cimatti, A., Katoen, J.-P., Noll, T., and Tonetta, S., "Formal Methods for Aerospace Systems," *Cyber-physical system design from an architecture analysis viewpoint*, 2017, pp. 133–159.

[71] Wang, K., Gong, Z., Hou, Y., Zhang, M., Liu, C., and Chen, R., "Model Based Design and Procedure of Flight Control System for Unmanned Aerial Vehicle," *2020 3rd International Conference on Unmanned Systems (ICUS)*, IEEE, 2020, pp. 763–768.

[72] McAree, O., Aitken, J. M., and Veres, S. M., "A Model Based Design Framework for Safety Verification of a Semi-Autonomous Inspection Drone," *2016 UKACC 11th International Conference on Control (CONTROL)*, IEEE, 2016, pp. 1–6.

[73] Balasubramaniam, B., Bagheri, H., Elbaum, S., and Bradley, J., "Investigating Controller Evolution and Divergence through Mining and Mutation*," *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS)*, 2020, pp. 151–161. https://doi.org/10.1109/ICCPS48487.2020.00022.

[74] Rierson, L., *Developing Safety-Critical Software: A Practical Guide for Aviation Software and DO-178C Compliance*, CRC Press, 2013. URL https://books.google.com/books?hl=en&lr=&id=VlxfJ1qmZ8AC&oi=fnd&pg=PP1&dq=do-178c&ots=w615C4-DP9&sig=vnkru17KadSviy8DQqcB7reBKhE.

[75] Jackson, M., and Henry, J., "Orion GN&C Model Based Development: Experience and Lessons Learned," *AIAA Guidance, Navigation, and Control Conference*, 2012, p. 5036.

[76] McVittie, T. I., Sindiy, O. V., and Simpson, K. A., "Model-Based System Engineering of the Orion Flight Test 1 End-to-End Information System," *2012 IEEE Aerospace Conference*, IEEE, 2012, pp. 1–11.

[77] Odegard, R. L., "Model-Based GN and C Simulation and Flight Software Development for Orion Missions beyond LEO," *2014 IEEE Aerospace Conference*, Big Sky, Montana, United States, 2014. URL https://ntrs.nasa.gov/search.jsp?R=20140003581.

[78] Simpson, K. A., Sindiy, O. V., and McVittie, T. I., "Orion Flight Test 1 Architecture—Observed Benefits of a Model Based Engineering Approach," *2012 IEEE Aerospace Conference*, IEEE, 2012, pp. 1–7.

[79] Kausch, H., Pfeiffer, M., Raco, D., Rumpe, B., and Schweiger, A., "Correct and Sustainable Development Using Model-based Engineering and Formal Methods," *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*, IEEE, 2022, pp. 1–8.

[80] Cofer, D., "Formal Methods in the Aerospace Industry: Follow the Money," *International Conference on Formal Engineering Methods*, Springer, 2012, pp. 2–3.

[81] Lopez, D. M., Johnson, T., Tran, H.-D., Bak, S., Chen, X., and Hobbs, K. L., "Verification of Neural Network Compression of ACAS Xu Lookup Tables with Star Set Reachability," *AIAA Scitech 2021 Forum*, American Institute of Aeronautics and Astronautics, 2021. https://doi.org/10.2514/6.2021-0995, URL https://arc.aiaa.org/doi/abs/10.2514/6.2021-0995.

[82] Bradley, J. M., and Bagheri, H., "Control Software: Research Directions in the Intersection of Control Theory and Software Engineering," *AIAA Scitech 2020 Forum*, 2020, p. 2102. https://doi.org/10.2514/6.2020-2102, URL https://doi.org/10.2514/6.2020-2102.

[83] Eubank, R. D., Bradley, J., and Atkins, E., "Energy-Aware Multiflight Planning for an Unattended Seaplane: Flying Fish," *Journal of Aerospace Information Systems*, 2016, pp. 1–19. https://doi.org/10.2514/1.I010484, URL http://dx.doi.org/10.2514/1.I010484.

[84] Primatesta, S., Rizzo, A., and la Cour-Harbo, A., "Ground Risk Map for Unmanned Aircraft in Urban Environments," *Journal of Intelligent & Robotic Systems*, Vol. 97, No. 3, 2020, pp. 489–509. https://doi.org/10.1007/s10846-019-01015-z, URL https://doi.org/10.1007/s10846-019-01015-z.

[85] Ong, H. Y., and Kochenderfer, M. J., "Markov Decision Process-Based Distributed Conflict Resolution for Drone Air Traffic Management," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 1, 2017, pp. 69–80. https://doi.org/10.2514/1.G001822, URL https://doi.org/10.2514/1.G001822.

[86] Plowcha, A., Sun, Y., Detweiler, C., and Bradley, J., "Predicting Digging Success for Unmanned Aircraft System Sensor Emplacement," *2018 International Symposium on Experimental Robotics*, 2018 International Symposium on Experimental Robotics, Buenos Aires, Argentina, 2018.

[87] Beard, R. W., and McLain, T. W., *Small Unmanned Aircraft: Theory and Practice*, Princeton University Press, 2012.

[88] Castillo, P., Albertos, P., Garcia, P., and Lozano, R., "Simple Real-time Attitude Stabilization of a Quad-rotor Aircraft With Bounded Signals," *Proceedings of the 45th IEEE Conference on Decision and Control*, 2006, pp. 1533–1538. https://doi.org/10.1109/CDC.2006.377137.

[89] Kim, J.-H., Sukkarieh, S., and Wishart, S., "Real-Time Navigation, Guidance, and Control of a UAV Using Low-Cost Sensors," *Field and Service Robotics*, Springer, 2003, pp. 299–309.

[90] Liu, J., *Real-Time Systems*, Prentice Hall, 2000.

[91] Katz, P., and Powell, J. D., "Selection of Sampling Rate for Digital Control of Aircrafts," Tech. rep., Sep. 1974. URL https://ntrs.nasa.gov/search.jsp?R=19750002880.

[92] Powell, J. D., and Katz, P., "Sample rate selection for aircraft digital control," *AIAA Journal*, Vol. 13, No. 8, 1975, p. 975–979.

[93] Zappulla, R., and Romano, M., "A Systematic Approach to Determining the Minimum Sampling Rate for Real-Time Spacecraft Control," *27th AAS/AIAA Spaceflight Mechanics Meeting, AAS Paper*, 2017, pp. 17–424.

[94] Franklin, G. F., Workman, M. L., and Powell, D., *Digital Control of Dynamic Systems*, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1998.

[95] Guerrero-Castellanos, J. F., Téllez-Guzmán, J. J., Durand, S., Marchand, N., and Alvarez-Muñoz, J. U., "Event-Triggered Nonlinear Control for Attitude Stabilization of a Quadrotor," *Unmanned Aircraft Systems (ICUAS), 2013 International Conference On*, 2013, pp. 584–591. https://doi.org/10.1109/ICUAS.2013.6564736.

[96] Kopetz, H., *Real-Time Systems: Design Principles for Distributed Embedded Applications*, Springer, 2011.

[97] Åström, K. J., and Bernhardsson, B. M., "Comparison of Riemann and Lebesgue sampling for first order stochastic systems," *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, Vol. 2, IEEE, 2002, p. 2011–2016.

[98] Bradley, J., and Atkins, E., "Coupled Cyber-Physical System Modeling and Coregulation of a CubeSat," *IEEE Transactions on Robotics*, Vol. 31, No. 2, 2015, pp. 443–456. https://doi.org/10.1109/TRO.2015.2409431.

[99] Zhang, X., and Bradley, J., "Controller Design for Time-varying Sampling, Co-Regulated Systems," *IEEE Control Systems Letters*, 2020, pp. 1–1. https://doi.org/10.1109/LCSYS.2020.3044261.

[100] Zhang, X., and Bradley, J., "Stability Analysis for a Class of Resource-Aware, Co-Regulated Systems," *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 193–200. https://doi.org/10.1109/CDC40024.2019.9029267, URL https://doi.org/10.1109/CDC40024.2019.9029267.

[101] Zhang, X., and Bradley, J., "Rethinking Sampled-Data Control for Unmanned Aircraft Systems," *Sensors*, Vol. 22, No. 4, 2022. https://doi.org/10.3390/s22041525, URL https://www.mdpi.com/1424-8220/22/4/1525.

[102] Fernando, C., Detweiler, C., and Bradley, J., "Co-Regulated Consensus of Cyber-Physical Resources in Multi-Agent Unmanned Aircraft Systems," *Electronicsweek*, Vol. 8, No. 5, 2019, p. 569.

[103] Agrawal, M., Cofer, D., and Samad, T., "Real-time adaptive resource management for advanced avionics," *Control Systems, IEEE*, Vol. 23, No. 1, 2003, pp. 76–86. https://doi.org/10.1109/MCS.2003.1172831.

[104] Atkins, E. M., and Sanner, R. M., "QoS Tradeoffs for Guidance, Navigation, and Control," *2002 IEEE Aerospace Conference Proceedings*, Vol. 7, IEEE, Big Sky, Montana, 2002, p. 7–3333–7–3341 vol.7. https://doi.org/10.1109/AERO.2002.1035311.

[105] Hong, S., Lee, S. U., Huang, X., Khonji, M., Alyassi, R., and Williams, B. C., "An Anytime Algorithm for Chance Constrained Stochastic Shortest Path Problems and Its Application to Aircraft Routing," *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 475–481.

[106] Sujit, P. B., and Beard, R., "Multiple UAV Path Planning Using Anytime Algorithms," *2009 American Control Conference*, IEEE, 2009, pp. 2978–2983.

[107] Buttazzo, G. C., Lipari, G., and Abeni, L., "Elastic Task Model for Adaptive Rate Control," *Proceedings 19th IEEE Real-Time Systems Symposium (Cat. No.98CB36279)*, 1998, pp. 286–295. https://doi.org/10.1109/REAL.1998.739754.

[108] Buttazzo, G. C., Bini, E., and Buttle, D., "Rate-Adaptive Tasks: Model, Analysis, and Design Issues," *2014 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2014, pp. 1–6. https://doi.org/10.7873/DATE.2014.266.

[109] Kim, J., Lakshmanan, K., and Rajkumar, R. R., "Rhythmic tasks: A new task model with continually varying periods for cyber-physical systems," *Proceedings of the 2012 IEEE/ACM Third International Conference on Cyber-Physical Systems*, IEEE Computer Society, 2012, p. 55–64.

[110] Fisac, J. F., Akametalu, A. K., Zeilinger, M. N., Kaynama, S., Gillula, J., and Tomlin, C. J., "A General Safety Framework for Learning-Based Control in Uncertain Robotic Systems," *IEEE Transactions on Automatic Control*, 2018.

[111] Gillula, J. H., and Tomlin, C. J., "Guaranteed Safe Online Learning via Reachability: Tracking a Ground Target Using a Quadrotor," *2012 IEEE International Conference on Robotics and Automation*, IEEE, 2012, pp. 2723–2730.

[112] Robinette, P. K., Heiner, B. K., Ravaioli, U., Hamilton, N., Johnson, T. T., and Hobbs, K. L., "Reinforcement Learning Heuristics for Aerospace Control Systems," *2022 IEEE Aerospace Conference (AERO)*, 2022, pp. 1–12. https://doi.org/10.1109/AERO53065.2022.9843224.

[113] Muvva, K., Bradley, J. M., Wolf, M., and Johnson, T., "Assuring Learning-Enabled Components in Small Unmanned Aircraft Systems," *AIAA Scitech 2021 Forum*, 2021. https://doi.org/10.2514/6.2021-0994, URL https://arc.aiaa.org/doi/abs/10.2514/6.2021-0994.

[114] Feron, E., and Alegre, F., "Control Software Analysis, Part I Open-Loop Properties," *arXiv preprint arXiv:0809.4812*, 2008. URL https://arxiv.org/abs/0809.4812.

[115] Feron, E., and Alegre, F., “Control Software Analysis, Part II: Closed-loop Analysis,” *arXiv preprint arXiv:0812.1986*, 2008. URL https://arxiv.org/abs/0812.1986.

[116] Wiels, V., Delmas, R., Doose, D., Garoche, P.-L., Cazin, J., and Durrieu, G., “Formal Verification of Critical Aerospace Software,” *AerospaceLab*, , No. 4, 2012, pp. p–1. URL https://hal.archives-ouvertes.fr/hal-01184099/.

[117] Feron, E., “From Control Systems to Control Software,” *IEEE Control Systems Magazine*, Vol. 30, No. 6, 2010, pp. 50–71. https://doi.org/10.1109/MCS.2010.938196, URL http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5643477.

[118] Ore, J.-P., Detweiler, C., and Elbaum, S., “Phriky-Units: A Lightweight, Annotation-free Physical Unit Inconsistency Detection Tool,” *Proceedings of the 26th ACM SIGSOFT International Symposium on Software Testing and Analysis*, ACM, New York, NY, USA, 2017, pp. 352–355. https://doi.org/10.1145/3092703.3098219, URL http://doi.acm.org/10.1145/3092703.3098219.

[119] Sampigethaya, K., and Poovendran, R., “Aviation Cyber–Physical Systems: Foundations for Future Aircraft and Air Transport,” *Proceedings of the IEEE*, Vol. 101, No. 8, 2013, pp. 1834–1855.

[120] Clarke, J.-P. B., and Tomlin, C. J., “Some Steps toward Autonomy in Aeronautics,” *The Bridge*, Vol. 50, No. 2, 2020.

[121] Grigorescu, S., Trasnea, B., Cocias, T., and Macesanu, G., “A survey of deep learning techniques for autonomous driving,” *Journal of Field Robotics*, Vol. 37, No. 3, 2020, pp. 362–386.

[122] Furfaro, R., Bloise, I., Orlandelli, M., Di Lizia, P., Topputo, F., Linares, R., et al., “Deep learning for autonomous lunar landing,” *2018 AAS/AIAA Astrodynamics Specialist Conference*, Vol. 167, Univelt, 2018, pp. 3285–3306.

[123] Campbell, T., Furfaro, R., Linares, R., and Gaylor, D., “A deep learning approach for optical autonomous planetary relative terrain navigation,” *Spaceflight Mechanics*, Vol. 160, 2017, pp. 3293–3302.

[124] Izzo, D., Märtens, M., and Pan, B., “A survey on artificial intelligence trends in spacecraft guidance dynamics and control,” *Astrodynamics*, Vol. 3, No. 4, 2019, pp. 287–299.

[125] Warren, M., Greeff, M., Patel, B., Collier, J., Schoellig, A. P., and Barfoot, T. D., “There’s No Place Like Home: Visual Teach and Repeat for Emergency Return of Multirotor UAVs During GPS Failure,” *IEEE Robotics and Automation Letters*, Vol. 4, No. 1, 2019, pp. 161–168. https://doi.org/10.1109/LRA.2018.2883408.

[126] Rodriguez-Ramos, A., Sampedro, C., Bavle, H., Moreno, I. G., and Campoy, P., “A Deep Reinforcement Learning Technique for Vision-Based Autonomous Multirotor Landing on a Moving Platform,” *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1010–1017. https://doi.org/10.1109/IROS.2018.8594472.

[127] Falanga, D., Zanchettin, A., Simovic, A., Delmerico, J., and Scaramuzza, D., “Vision-based autonomous quadrotor landing on a moving platform,” *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, IEEE, 2017, pp. 200–207.

[128] Hinzmann, T., Schneider, T., Dymczyk, M., Schaffner, A., Lynen, S., Siegwart, R., and Gilitschenski, I., “Monocular visual-inertial SLAM for fixed-wing UAVs using sliding window based nonlinear optimization,” *International Symposium on Visual Computing*, Springer, 2016, pp. 569–581.

[129] Hinzmann, T., Cadena, C., Nieto, J., and Siegwart, R., “Flexible Trinocular: Non-rigid Multi-Camera-IMU Dense Reconstruction for UAV Navigation and Mapping,” *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 1137–1142. https://doi.org/10.1109/IROS40897.2019.8967651.

[130] Tallamraju, R., Price, E., Ludwig, R., Karlapalem, K., Bülthoff, H. H., Black, M. J., and Ahmad, A., “Active perception based formation control for multiple aerial vehicles,” *IEEE Robotics and Automation Letters*, Vol. 4, No. 4, 2019, pp. 4491–4498.

[131] Surmann, H., Worst, R., Buschmann, T., Leinweber, A., Schmitz, A., Senkowski, G., and Goddemeier, N., “Integration of uavs in urban search and rescue missions,” *2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, IEEE, 2019, pp. 203–209.

[132] Qingqing, L., Taipalmaa, J., Queralta, J. P., Gia, T. N., Gabbouj, M., Tenhunen, H., Raitoharju, J., and Westerlund, T., “Towards Active Vision with UAVs in Marine Search and Rescue: Analyzing Human Detection at Variable Altitudes,” *2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2020, pp. 65–70. https://doi.org/10.1109/SSRR50563.2020.9292596.

[133] De Vivo, F., Battipede, M., and Johnson, E., “Infra-red line camera data-driven edge detector in UAV forest fire monitoring,” *Aerospace Science and Technology*, Vol. 111, 2021, p. 106574.

[134] Lee, W., Kim, S., Lee, Y.-T., Lee, H.-W., and Choi, M., "Deep neural networks for wild fire detection with unmanned aerial vehicle," *2017 IEEE international conference on consumer electronics (ICCE)*, IEEE, 2017, pp. 252–253.

[135] Geirhos, R., Narayanappa, K., Mitzkus, B., Thieringer, T., Bethge, M., Wichmann, F. A., and Brendel, W., "Partial success in closing the gap between human and machine vision," *Advances in Neural Information Processing Systems*, Vol. 34, 2021, pp. 23885–23899.

[136] Li, Z., Liu, F., Yang, W., Peng, S., and Zhou, J., "A survey of convolutional neural networks: analysis, applications, and prospects," *IEEE transactions on neural networks and learning systems*, 2021.

[137] Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., and Dosovitskiy, A., "Do vision transformers see like convolutional neural networks?" *Advances in Neural Information Processing Systems*, Vol. 34, 2021, pp. 12116–12128.

[138] Khan, A., Sohail, A., Zahoora, U., and Qureshi, A. S., "A survey of the recent architectures of deep convolutional neural networks," *Artificial intelligence review*, Vol. 53, No. 8, 2020, pp. 5455–5516.

[139] Nobis, F., Geisslinger, M., Weber, M., Betz, J., and Lienkamp, M., "A deep learning-based radar and camera sensor fusion architecture for object detection," *2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, IEEE, 2019, pp. 1–7.

[140] Liang, M., Yang, B., Chen, Y., Hu, R., and Urtasun, R., "Multi-task multi-sensor fusion for 3d object detection," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7345–7353.

[141] Yeong, D. J., Velasco-Hernandez, G., Barry, J., and Walsh, J., "Sensor and sensor fusion technology in autonomous vehicles: A review," *Sensors*, Vol. 21, No. 6, 2021, p. 2140.

[142] Kivelevitch, E. H., Dionne, G., Roose, T., Arora, P., Fanous, B., Salvo, R., Pellissier, V., Losada, R., and Gentile, R., "Sensor Fusion Tools in Support of Autonomous Systems," *AIAA Scitech 2019 Forum*, 2019, p. 0384.

[143] Hockman, B., and Pavone, M., "Stochastic motion planning for hopping rovers on small solar system bodies," *Robotics research*, Springer, 2020, pp. 877–893.

[144] Pflueger, M., Agha, A., and Sukhatme, G. S., "Rover-IRL: Inverse Reinforcement Learning With Soft Value Iteration Networks for Planetary Rover Path Planning," *IEEE Robotics and Automation Letters*, Vol. 4, No. 2, 2019, pp. 1387–1394. https://doi.org/10.1109/LRA.2019.2895892.

[145] Ge, D., Cui, P., and Zhu, S., "Recent development of autonomous GNC technologies for small celestial body descent and landing," *Progress in Aerospace Sciences*, Vol. 110, 2019, p. 100551.

[146] Watanabe, T., and Johnson, E. N., "Trajectory Generation using Deep Neural Network," *2018 AIAA Information Systems-AIAA Infotech@ Aerospace*, AIAA, 2018, p. 1893.

[147] Lakis, M. J., and Daher, N., "A Simple Neural Network for Efficient Real-time Generation of Dynamically-Feasible Quadrotor Trajectories," *2021 IEEE 3rd International Multidisciplinary Conference on Engineering Technology (IMCET)*, IEEE, 2021, pp. 113–118.

[148] Schlichting, M. R., Notter, S., and Fichter, W., "LSTM-based spatial encoding: Explainable path planning for time-variant multi-agent systems," *AIAA Scitech 2021 Forum*, 2021, p. 1860.

[149] Schlichting, M. R., Notter, S., and Fichter, W., "Long Short-Term Memory for Spatial Encoding in Multi-Agent Path Planning," *Journal of Guidance, Control, and Dynamics*, Vol. 45, No. 5, 2022, pp. 952–961.

[150] Semnani, S. H., Liu, H., Everett, M., De Ruiter, A., and How, J. P., "Multi-agent motion planning for dense and dynamic environments via deep reinforcement learning," *IEEE Robotics and Automation Letters*, Vol. 5, No. 2, 2020, pp. 3221–3226.

[151] Everett, M., Chen, Y. F., and How, J. P., "Motion planning among dynamic, decision-making agents with deep reinforcement learning," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 3052–3059.

[152] Chen, Y. F., Everett, M., Liu, M., and How, J. P., "Socially aware motion planning with deep reinforcement learning," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2017, pp. 1343–1350.

[153] Sekhon, J., and Fleming, C., "A spatially and temporally attentive joint trajectory prediction framework for modeling vessel intent," *Learning for Dynamics and Control*, PMLR, 2020, pp. 318–327.

[154] Sekhon, J., and Fleming, C., "SCAN: A Spatial Context Attentive Network for Joint Multi-Agent Intent Prediction," *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35, 2021, pp. 6119–6127.

[155] Everett, M., Habibi, G., Sun, C., and How, J. P., "Reachability analysis of neural feedback loops," *IEEE Access*, Vol. 9, 2021, pp. 163938–163953.

[156] Everett, M., Habibi, G., and How, J. P., "Robustness analysis of neural networks via efficient partitioning with applications in control systems," *IEEE Control Systems Letters*, Vol. 5, No. 6, 2020, pp. 2114–2119.

[157] Watanabe, T., and Johnson, E., "Automatic landing on a moving platform using deep neural network estimation," *7th AHS Technical Meeting on VTOL Unmanned Aircraft Systems and Autonomy*, 2017, p. 1.

[158] Păsăreanu, C. S., Gopinath, D., and Yu, H., "Compositional verification for autonomous systems with deep learning components," *Safe, Autonomous and Intelligent Vehicles*, Springer, 2019, pp. 187–197.

[159] Bakirtzis, G., Fleming, C. H., and Vasilakopoulou, C., "Categorical semantics of cyber-physical systems theory," *ACM Transactions on Cyber-Physical Systems*, Vol. 5, No. 3, 2021, pp. 1–32.

[160] Bakirtzis, G., Subrahmanian, E., and Fleming, C. H., "Compositional thinking in cyberphysical systems theory," *Computer*, Vol. 54, No. 12, 2021, pp. 50–59.

[161] Bakirtzis, G., Vasilakopoulou, C., and Fleming, C. H., "Compositional Cyber-Physical Systems Modeling," *Proceedings of the 2019 Applied Category Theory Conference (ACT 2020)*, 2020, p. 0. https://doi.org/10.4204/EPTCS.333.9.

[162] Jafarzadeh, H., and Fleming, C., "Learning model predictive control for connected autonomous vehicles," *2019 IEEE 58th Conference on Decision and Control (CDC)*, IEEE, 2019, pp. 2336–2343.

[163] Jafarzadeh, H., and Fleming, C., "DMPC: A data-and model-driven approach to predictive control," *Automatica*, Vol. 131, 2021, p. 109729.

[164] Jafarzadeh, H., and Fleming, C., "Gaussian Process-based Model Predictive Controller for Connected Vehicles with Uncertain Wireless Channel," *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, IEEE, 2021, pp. 3515–3520.

[165] Majumdar, A., and Pavone, M., "How should a robot assess risk? towards an axiomatic theory of risk in robotics," *Robotics Research*, Springer, 2020, pp. 75–84.

[166] Chow, Y., Ghavamzadeh, M., Janson, L., and Pavone, M., "Risk-constrained reinforcement learning with percentile risk criteria," *The Journal of Machine Learning Research*, Vol. 18, No. 1, 2017, pp. 6070–6120.

[167] Ferlez, J., Elnaggar, M., Shoukry, Y., and Fleming, C., "Shieldnn: A provably safe nn filter for unsafe nn controllers," *arXiv preprint arXiv:2006.09564*, 2020.

[168] Dunham, J., Johnson, E., Feron, E., and German, B., "Application of Dempster–Shafer Networks to a Real-Time Unmanned Systems Risk Analysis Framework," *Journal of Aerospace Information Systems*, Vol. 18, No. 9, 2021, pp. 616–631.

[169] ARP4761, S., "Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment (1996)," , 1996.

[170] Society of Automotive Engineers Aerospace, *Guidelines for Development of Civil Aircraft and Systems: SAE ARP 4754 Rev. A.*, SAE, 2010.

[171] Zhang, J. M., Harman, M., Ma, L., and Liu, Y., "Machine learning testing: Survey, landscapes and horizons," *IEEE Transactions on Software Engineering*, 2020.

[172] Mitchell, T. M., *Machine learning*, McGraw Hill, 1997.

[173] Sekhon, J., and Fleming, C., "Towards Improved Testing For Deep Learning," *2019 IEEE/ACM 41st International Conference on Software Engineering: New Ideas and Emerging Results (ICSE-NIER)*, 2019, pp. 85–88. https://doi.org/10.1109/ICSE-NIER.2019.00030.

[174] Usman, M., Sun, Y., Gopinath, D., Dange, R., Manolache, L., and Păsăreanu, C. S., "An overview of structural coverage metrics for testing neural networks," *International Journal on Software Tools for Technology Transfer*, 2022, pp. 1–13.

[175] Pei, K., Cao, Y., Yang, J., and Jana, S., "Deepxplore: Automated whitebox testing of deep learning systems," *proceedings of the 26th Symposium on Operating Systems Principles*, 2017, pp. 1–18.

[176] Sun, Y., Huang, X., Kroening, D., Sharp, J., Hill, M., and Ashmore, R., "Structural test coverage criteria for deep neural networks," *ACM Transactions on Embedded Computing Systems (TECS)*, Vol. 18, No. 5s, 2019, pp. 1–23.

[177] National Academies of Sciences, Engineering, and Medicine, "Emerging Hazards in Commercial Aviation—Report 1: Initial Assessment of Safety Data and Analysis Processes," 2022.

[178] Feigh, K. M., and Pritchett, A. R., "Requirements for effective function allocation: A critical review," *Journal of cognitive engineering and decision making*, Vol. 8, No. 1, 2014, pp. 23–32.

[179] Pritchett, A. R., "Reviewing the role of cockpit alerting systems," *Human Factors and Aerospace Safety: An International Journal: No. 1*, Taylor and Francis, 2017, pp. 5–38.

[180] Pritchett, A. R., "Aviation automation: General perspectives and specific guidance for the design of modes and alerts," *Reviews of human factors and ergonomics*, Vol. 5, No. 1, 2009, pp. 82–113.

[181] Bailey, R. W., *Human performance engineering: Using human factors/ergonomics to achieve computer system usability*, Prentice-Hall, Inc., 1989.

[182] Norman, D. A., "The 'problem'with automation: inappropriate feedback and interaction, not 'over-automation'," *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, Vol. 327, No. 1241, 1990, pp. 585–593.

[183] Fitts, P. M., "Human engineering for an effective air-navigation and traffic-control system." 1951.

[184] Sheridan, T. B., and Telerobotics, A., "Human supervisory control," , 1992.

[185] Mumaw, R. J., Billman, D., and Feary, M. S., "Analysis of pilot monitoring skills and a review of training effectiveness," 2020.

[186] Villa, A., "Computational Modeling of Monitoring within Human-Autonomy Teams with Novel Delegations of Authority and Responsibility," 2022.

[187] Villa, A., and Pritchett, A., "Computational Modeling of Monitoring within Human-Autonomy Teams with Distributed Information," *AIAA SCITECH 2022 Forum*, 2022, p. 2438.

[188] Parasuraman, R., and Riley, V., "Humans and automation: Use, misuse, disuse, abuse," *Human factors*, Vol. 39, No. 2, 1997, pp. 230–253.

[189] Hollenbeck, J. R., Ilgen, D. R., Sego, D. J., Hedlund, J., Major, D. A., and Phillips, J., "Multilevel theory of team decision making: Decision performance in teams incorporating distributed expertise." *Journal of applied psychology*, Vol. 80, No. 2, 1995, p. 292.

[190] Hutchins, E., *Cognition in the Wild*, MIT press, 1995.

[191] Zellmer-Bruhn, M. E., "Interruptive events and team knowledge acquisition," *Management science*, Vol. 49, No. 4, 2003, pp. 514–528.

[192] Speier, C., Valacich, J. S., and Vessey, I., "The influence of task interruption on individual decision making: An information overload perspective," *Decision sciences*, Vol. 30, No. 2, 1999, pp. 337–360.

[193] Dorneich, M. C., Ververs, P. M., Mathan, S., Whitlow, S., and Hayes, C. C., "Considering etiquette in the design of an adaptive system," *Journal of Cognitive Engineering and Decision Making*, Vol. 6, No. 2, 2012, pp. 243–265.

[194] Miller, C. A., and Parasuraman, R., "Designing for flexible interaction between humans and automation: Delegation interfaces for supervisory control," *Human factors*, Vol. 49, No. 1, 2007, pp. 57–75.

[195] Guerlain, S. A., Smith, P. J., Obradovich, J. H., Rudmann, S., Strohm, P., Smith, J. W., Svirbely, J., and Sachs, L., "Interactive critiquing as a form of decision support: An empirical evaluation," *Human factors*, Vol. 41, No. 1, 1999, pp. 72–89.

[196] Hollnagel, E., *Barriers and accident prevention*, Routledge, 2016.

[197] Hollnagel, E., Woods, D. D., and Leveson, N., *Resilience engineering: Concepts and precepts*, Ashgate Publishing, Ltd., 2006.

[198] Sheridan, T. B., "Risk, human error, and system resilience: fundamental ideas," *Human factors*, Vol. 50, No. 3, 2008, pp. 418–426.

[199] Chialastri, A., and Pozzi, S., "Resilience in the aviation system," *international conference on computer safety, reliability, and security*, Springer, 2008, pp. 86–98.

[200] Feigh, K. M., and Pritchett, A. R., "Modeling work for cognitive work support system design in operational control centers," *Journal of Cognitive Engineering and Decision Making*, Vol. 4, No. 1, 2010, pp. 1–26.

[201] Feigh, K., Pritchett, A., Denq, T., and Jacko, J., "Contextual control modes during an airline rescheduling task," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 50, SAGE Publications Sage CA: Los Angeles, CA, 2006, pp. 457–461.

[202] Hollnagel, E., *Human reliability analysis: Context and control*, Academic Press, 1993.

[203] Feigh, K. M., "Incorporating multiple patterns of activity into the design of cognitive work support systems," *Cognition, Technology & Work*, Vol. 13, No. 4, 2011, pp. 259–279.

[204] Kirlik, A., "Modeling strategic behavior in human-automation interaction: Why an" aid" can (and should) go unused," *Human factors*, Vol. 35, No. 2, 1993, pp. 221–242.

[205] Radio Technical Commission for Aeronautics, "DO-178B – Software Considerations in Airborne Systems and Equipment Certification," , 1992. URL https://www.rtca.org/content/standards-guidance-materials.

[206] Radio Technical Commission for Aeronautics, "DO-178C/ED-12C – Software Considerations in Airborne Systems and Equipment Certification," , 2012. URL https://www.rtca.org/content/standards-guidance-materials.

[207] Radio Technical Commission for Aeronautics, "DO-333 – Formal Methods Supplement to DO-178C and DO-278A," , 2011. URL https://www.rtca.org/content/standards-guidance-materials.

[208] Radio Technical Commission for Aeronautics, "DO-254 – Design Assurance Guidance for Airborne Electronic Hardware," , 2000. URL https://www.rtca.org/content/standards-guidance-materials.

[209] Erzberger, H., and Heere, K., "Algorithm and operational concept for resolving short-range conflicts," *Proc. IMechE G J. Aerosp. Eng.*, Vol. 224, No. 2, 2010, pp. 225–243. https://doi.org/10.1243/09544100JAERO546, URL http://pig.sagepub.com/content/224/2/225.abstract.

[210] Zhao, Y., and Rozier, K. Y., "Formal Specification and Verification of a Coordination Protocol for an Automated Air Traffic Control System," *Science of Computer Programming Journal*, Vol. 96, No. 3, 2014, pp. 337–353.

[211] Zhao, Y., and Rozier, K. Y., "Probabilistic Model Checking for Comparative Analysis of Automated Air Traffic Control Systems," *Proceedings of the 33rd IEEE/ACM International Conference On Computer-Aided Design (ICCAD 2014)*, IEEE/ACM, San Jose, California, U.S.A., 2014, pp. 690–695.

[212] Mattarei, C., Cimatti, A., Gario, M., Tonetta, S., and Rozier, K. Y., "Comparing Different Functional Allocations in Automated Air Traffic Control Design," *Proceedings of Formal Methods in Computer-Aided Design (FMCAD 2015)*, IEEE/ACM, Austin, Texas, U.S.A, 2015.

[213] Gario, M., Cimatti, A., Mattarei, C., Tonetta, S., and Rozier, K. Y., "Model Checking at Scale: Automated Air Traffic Control Design Space Exploration," *Proceedings of 28th International Conference on Computer Aided Verification (CAV 2016)*, LNCS, Vol. 9780, Springer, Toronto, ON, Canada, 2016, pp. 3–22. https://doi.org/10.1007/978-3-319-41540-6_1.

[214] Dureja, R., and Rozier, K. Y., "FuseIC3: An Algorithm for Checking Large Design Spaces," *Proceedings of the 17th Conference on Formal Methods in Computer-Aided Design (FMCAD)*, IEEE/ACM, Vienna, Austria, 2017, pp. 164–171.

[215] Dureja, R., and Rozier, K. Y., "More Scalable LTL Model Checking via Discovering Design-Space Dependencies (D^3)," *Proceedings of the 20th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, Lecture Notes in Computer Science (LNCS), Vol. part I, 10805, Springer-Verlag, 2018, pp. 309–327.

[216] Dureja, R., and Rozier, K. Y., "Incremental Design-space Model Checking Via Reusable Reachable State Approximations," *Formal Methods in System Design (FMSD) Journal*, Vol. online, No. 1, 2022. https://doi.org/{https://doi.org/10.1007/s10703-022-00389-5}.

[217] Fisher, M., Mascardi, V., Rozier, K. Y., Schlingloff, H., Winikoff, M., and Yorke-Smith, N., "Towards a Framework for Certification of Reliable Autonomous Systems," *Autonomous Agents and Multi-Agent Systems (JAAMAS)*, Vol. 35, No. 8, 2021. https://doi.org/https://doi.org/10.1007/s10458-020-09487-2.

[218] Rozier, K. Y., "From Simulation to Runtime Verification and Back: Connecting Single-Run Verification Techniques," *Proceedings of the Spring Simulation Conference (SpringSim)*, Society for Modeling & Simulation International, Tucson, AZ, USA, 2019, pp. 1–10. https://doi.org/TBD, URL https://dl.acm.org/doi/10.5555/3338027.3338054.

[219] Schumann, J., Rozier, K. Y., Reinbacher, T., Mengshoel, O. J., Mbaya, T., and Ippolito, C., "Towards Real-time, On-board, Hardware-supported Sensor and Software Health Management for Unmanned Aerial Systems," *Proceedings of the 2013 Annual Conference of the Prognostics and Health Management Society (PHM2013)*, 2013, pp. 381–401.

[220] Reinbacher, T., Rozier, K. Y., and Schumann, J., "Temporal-Logic Based Runtime Observer Pairs for System Health Management of Real-Time Systems," *Proceedings of 20th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'14)*, Vol. LNCS 8413, Springer, 2014, pp. 357–372. https://doi.org/10.1007/978-3-642-54862-8_24.

[221] Kempa, B., Zhang, P., Jones, P. H., Zambreno, J., and Rozier, K. Y., "Embedding Online Runtime Verification for Fault Disambiguation on Robonaut2," *Proceedings of the 18th International Conference on Formal Modeling and Analysis of Timed Systems (FORMATS)*, Lecture Notes in Computer Science (LNCS), Vol. TBD, Springer, Vienna, Austria, 2020, p. TBD. https://doi.org/TBD, URL http://research.temporallogic.org/papers/KZJZR20.pdf.

[222] Dabney, J. B., Badger, J. M., and Rajagopal, P., "Adding a Verification View for an Autonomous Real-Time System Architecture," *Proceedings of SciTech Forum*, AIAA, 2021, p. Online. https://doi.org/https://doi.org/10.2514/6.2021-0566.

[223] Kempa, B., Johannsen, C., and Rozier, K. Y., "Improving Usability and Trust in Real-Time Verification of a Large-Scale Complex Safety-Critical System," *Ada User Journal*, Vol. September, 2022.

[224] FME Teaching Committee, "Formal Methods Teaching Workshop (FMTea)," Online: https://fmtea.github.io/, 2023.

[225] Rozier, K. Y., "On Teaching Applied Formal Methods in Aerospace Engineering," *Proceedings of the Formal Methods Teaching Workshop (FMTea) at the 3rd World Congress on Formal Methods*, Lecture Notes in Computer Science (LNCS), Vol. 11758, Springer, Porto, Portugal, 2019, pp. 111–131. https://doi.org/https://doi.org/10.1007/978-3-030-32441-4_8, URL https://fmtea.github.io/.

[226] Fleming, C. H., Spencer, M., Thomas, J., Leveson, N., and Wilkinson, C., "Safety assurance in NextGen and complex transportation systems," *Safety science*, Vol. 55, 2013, pp. 173–187.

[227] Fleming, C. H., and Leveson, N., "Integrating systems safety into systems engineering during concept development," *INCOSE International Symposium*, Vol. 25, Wiley Online Library, 2015, pp. 989–1003.

[228] Fleming, C. H., and Leveson, N. G., "Early concept development and safety analysis of future transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, Vol. 17, No. 12, 2016, pp. 3512–3523.

[229] Sun, M., and Fleming, C. H., "A new safety-guided design methodology to complement model-based safety analysis for safety assurance," *arXiv preprint arXiv:2205.00152*, 2022.

[230] Sztipanovits, J., Koutsoukos, X., Karsai, G., Kottenstette, N., Antsaklis, P., Gupta, V., Goodwine, B., Baras, J., and Wang, S., "Toward a Science of Cyber–Physical System Integration," *Proceedings of the IEEE*, Vol. 100, No. 1, 2012, pp. 29–44.