Complex systems engineering theory is a scientific theory

Complex systems engineering theory is a scientific theory

Eric Feron*
King Abdullah University of Science and Technology, Thuwal, Saudi Arabia

The proper design of complex engineering systems is what allows corporations and nations to distinguish themselves in a global competition for technical excellence and economic well-being. After quickly reviewing the central elements of systems engineering, we map all of them onto concepts of mathematics such as theorems and proofs, and onto scientific theories. This mapping allows the protagonists of complex systems engineering and design to map existing techniques from one field to the others; it provides a surprising number of suggestions for improving system design, especially system architecture, by leveraging advanced mathematical and / or scientific concepts in a productive way. In return, mathematicians and computer scientists can benefit from this bridge by bringing to bear many of their automated theorem provers to help with the design of complex systems. Clear classifications of what is "hard" and what is "easy" in mathematical proofs can instantaneously map onto similar appreciations for system design and its reliance on engineers' creativity. Last, understanding system design from the mathematical-scientific viewpoint can help the system engineer think more maturely about organizing the multitude of tasks required by systems engineering.

I. Introduction

Systems Engineering is concerned with putting several elements together to produce an emergent property that none of the individual components is capable of perform alone. Certain systems can be very simple. For example, the assembly of wax and wick gives a candle that can produce light for a long duration, something that neither the wax nor the wick may do alone. Likewise, the synergy of amadou and flint gives a mechanism to light up the candle, something neither element may do alone [1, vol II, p. 244]. A functionally equivalent, but far more complex modern illustration is the assembly of a lightbulb with the hugely complex electrical, world-wide power network via a socket. Some systems may be quite complex, such as in trains, computers, refineries and power plants. These systems are complex because they involve many items whose individual abilities are not capable, alone, of producing what the system produces. In general, it is considered good practice that the system have some use, that is, that the emergent property be supportive of an objective need. The need might be individual or societal. However, expressing a need of deep significance is not always necessary, especially at the didactic levels when students still develop their skills through education and practice and for whom designing an entire functional system may greatly exceed the duration of their academic studies. The emergent property expressed as a need for putting the system together is, however, a powerful source of motivation that should not be discouraged when it arises in anyone's mind.

Systems Engineering has long been present as the core of what constitutes "engineering". In popular culture, the "genius engineer" is often hailed as a single individual who has put together an evil piece of powerful machinery. The "genius engineer" is also often hailed as the single individual who develops a piece of machinery that "saves the world". Nowhere are these characters better illustrated than in Hergé's famous comic strips series, "Les aventures de Tintin" and "Les aventures de Jo, Zette, et Joko". In the former, the "good engineer" is decidedly the "Professeur Tournesol", who, by coming up with a series of sometimes bizarre inventions, proposes genius systems to meet or even anticipate the increasingly challenging needs of Hergé's technical ecosystem, including machinery to explore the bottom of the Sea, see "Le secret de la Licorne" [2] and "Le trésor de Rackham le Rouge" [3]. In "les aventures de Joe, Zette et Joko", Ingénieur Legrand is a responsible head-of-household who comes up with many systems, ranging from suspended bridges, see "La Vallée des Cobras" [4], to stratospheric airplanes, see "Le testament de Mr. Pump" [5] and "Destination New York" [6]. Evil engineers and scientists are also well-represented, especially in "Le Manitoba ne répond plus" [7], where a mad scientist has designed an enormous piece of machinery including a submarine base and a secret weapon aimed at ransoming the western world through piracy attacks on transatlantic cruise ships. Parodic versions of the "mad engineer/scientist" litter the literature and movie scenes, such as in "The Spy who shagged me", where subjects of derision include not only the concept of evil scientist by introducing Dr. Evil and his alter ego "Mini-Me", but also

*Professor, Biological, Electrical, Computer, and Mechanical Engineering, Thuwal 23955, AIAA Fellow, eric.feron@kaust.edu.sa

modern universities by emphasizing the necessity of his obtaining a doctoral degree to become capable of realizing his evil dream.

It is worthy to note that all these cultural efforts reflect a highly individualistic mentality commonly found in the "western hemisphere", where a single individual ends up spearheading and achieving major scientific and technical advances for the good or the bad of humanity. Newton (multiple mathematical and scientific discoveries), Gauss (mathematics) , Hilbert (Mathematics), Renard and Krebs (first dirigible, 1884), the Wright brothers (heavier than air flight, 1903), von Braun (first human landing on the moon) and Qian Xuesen (Chinese space program) are all credited, individually or in small groups of two or three at most, with the entirety of the discoveries attributed to them. More recently, Black Scholes (Option pricing), Wolfgang Ketterle (atom engineering and manipulation), Albert Fert (Magnetoresistance), Elon Musk (SpaceX and Tesla) are witnesses to this continuing fascination for the "lone genius". Reality about systems engineering is increasingly far from this story scheme however, with entire teams of people working together to produce today's wonders of the world. Witnesses of this trend are Boeing's 787, Airbus' A380, or the Moderna Covid Vaccine that may not be attributed to any outstanding individuals but, rather, outstanding teams of individuals instead. Going further, the conception, design, implementation, validation, operation and retirement of large systems increasingly includes computers and robots whose absence would make all current complex systems impossible to build. In that regard, the celebration of the group of engineers who contributed to the construction of North Korea's first missile, "Ri Hong Sop, head of North Korea's Nuclear Weapons Institute, and Hong Sung Mu, deputy director of the ruling Workers' Party of Korea's munitions industry department" [8], is a testimony to the softening stance of the country on a "winner takes all", since it might be argued that a truly totalitarian regime would have found the sole contributor to the program to be its political leader. Looking at today's China, recent progress in quantum computing and particle entanglement for communications do not emphasize individuals, but rather teams, to be recognized as true inventors and demonstrators of "first of a kind" research achievements [9]. The growing necessity of large teams that includes scientists, engineers, mathematicians, computers and robots to produce significant technological innovations can be justified by the increased complexity of systems to be built. To most, the tour de force consisting of the James Webb telescope or Ingenuity, the Ingenuity Mars Helicopter cannot be thought of outside the context of a large team effort with no "winner" as might be construed by increasingly antiquated beliefs. One of the last bastions of such thinking lies with universities whereby most, if not all, faculty promotions are based on individual performance as opposed to high-performance teams, thereby being more revealing of American corporate structures than true research achievement structures. The learned scientific and technological societies are not foreign to this notion, by awarding membership to individual performers not only based on their individual performance, but also the amount of money they have been able to be responsible for at some point or some other. This cheap ersatz is, indeed, revealing of the team-based nature of the citation while maintaining the superficial and increasingly outdated notion of "winner".

All that said, the vanishing possibility of one individual making a significant contribution to systems engineering remains. It is also an attempt at justifying the sole authorship of this paper. "Doing engineering systems and engineering science" now requires large teams, studying the *science* that underlies systems engineering can be done, though it might be initiated by an individual provided the proper level of abstraction and core elements of knowledge are utilized. It is with this thought in mind that the author believes he has been able to make a dent into defining a science of system engineering. In that regard, the author is indebted to Edward Crawley, whose realism and confidence have formed the intellectual and material basis allowing this paper to be written in near-optimal conditions [10]. Other authors dear to the author's heart are Pierre Teilhard de Chardin, on the one hand, and Edgar Morin, on the other hand. The first one provides the author with the view that no documented work, including his, is definite enough to claim full eternity, because the world as we know it is consistently dynamic and in accelerated evolution [11]. Morin is also present because his exceptional longevity combined with a brilliant mind has allowed him to develop one of the most thorough and most current theories of complex systems, with very well-articulated views allowing him to understand complex systems ranging from early physical systems to advanced human societies in an integrated fashion [12]. For the curious mind, Morin provides a very useful discussion on some of his other predecessors with whom the present work is closely related, including von Bertalanffy [13], and Ashby [14]. Compared with von Bertalanffy's, the present work attempts to position complex systems engineering science as deeply as possible in the direction of science and mathematics while carefully avoiding the introduction of "systems magic" *, that is, unsubstantiated moves or irrational thinking. Just like Teilhard, Morin's work is there to be always renewed and advanced in the light of the latest evolutions of the world.

*The author would like to thank Dimitri Mavris for coining this expression during a common student PhD defense.

II. Systems engineering today

Systems engineering covers a very large array of activities often summarized by the sequence of repeated acts of Conceiving, Designing, Implementing, Operating, and retiring a system, as inspired by Crawley's "CDIO" educational concept [15]. Often arranged in the famous "Vee diagram" for safety-critical systems, shown in Fig. 1, or spiral, waterfall and "agile" diagrams otherwise, these tasks include themselves other tasks, and they are detailed below.

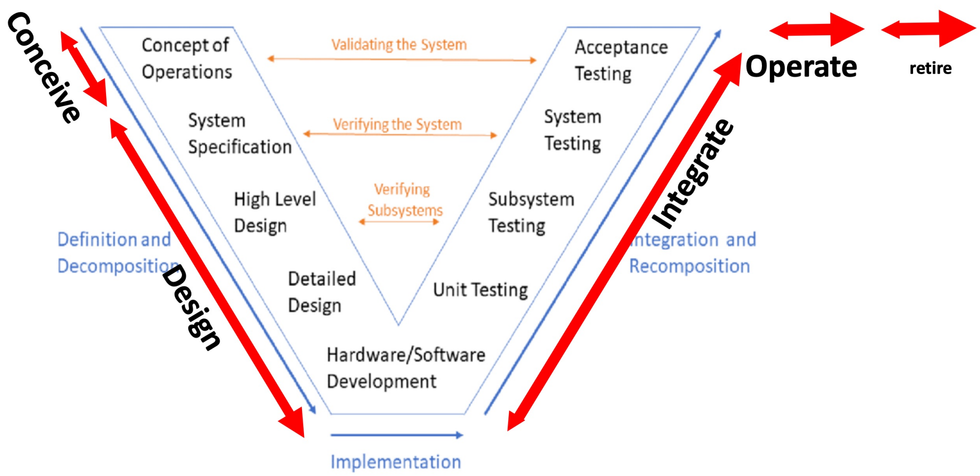

Fig. 1 The Vee diagram for System Engineering

A. Conceiving

The conception of complex systems is probably one of the most exalting areas of engineering, because it is about identifying societal needs and addressing them on time at affordable costs. Some systems have been conceived centuries before they have been built. This is the case of many of Da Vinci's inventions, which include the helicopter. When a conceived system is built too early, the system may be technologically or economically infeasible. Such is the case of the first Iridium satellite constellation, whose low bandwidth and high operating costs led to the quasi-disappearance of Motorola its sponsor [16]. When a conceived system is built too late, the system may have already been developed by the competition. Thus, conceiving a system is notoriously independent of whether the system will eventually be built or not, and many system concepts remain "in boxes" instead of turning into products. There seems to be a remarkable number corporate entities who came up with remarkable system concepts, but who were incapable of performing their development on-time. This is the case of Kodak, which is known worldwide for its camera films. Very early on, Kodak's engineers came up with the concept of digital camera, which was then discarded for fear that the new concept would damage Kodak's traditional business. When Kodak eventually adopted the electronic camera concept and designed its own, it was technologically and logistically so far behind its competition that it could not catch up and eventually dwindled to a corporate entity worth only one hundredth the top value it reached in 1996 [17]. "Just-in-time", successful system conception is therefore a tricky business that requires a lot of astuteness to properly weigh needs, technical feasibility, and, possibly, the need new engineering science. Gathering as much knowledge as possible is the sure way to address a dilemma that is similar to the alerting system concept derived from signal detection theory [18]. Early conception may lead to decisions made in the wake of a "false alarm", that is, an opportunity that is only an illusion and can yield significant economic losses for corporate agents. Late conception may lead to useless decisions performed in the wake of a "missed detection", whereby the true opportunity was recognized earlier by the competition, often leading the corporate agent to far bigger economic losses. A probabilistic interpretation of the range of possible policies to trigger the initiation of a project or not can be adapted from [18] to produce a "Corporate System Operating Curve" shown in Figure 2. A key element of the "Corporate System operating curve" is that it solely depends on the technical excellence of the corporate entity involved. A corporate entity endowed with less technical and human excellence is always "dominated" by a corporate entity with more technical and human excellence.

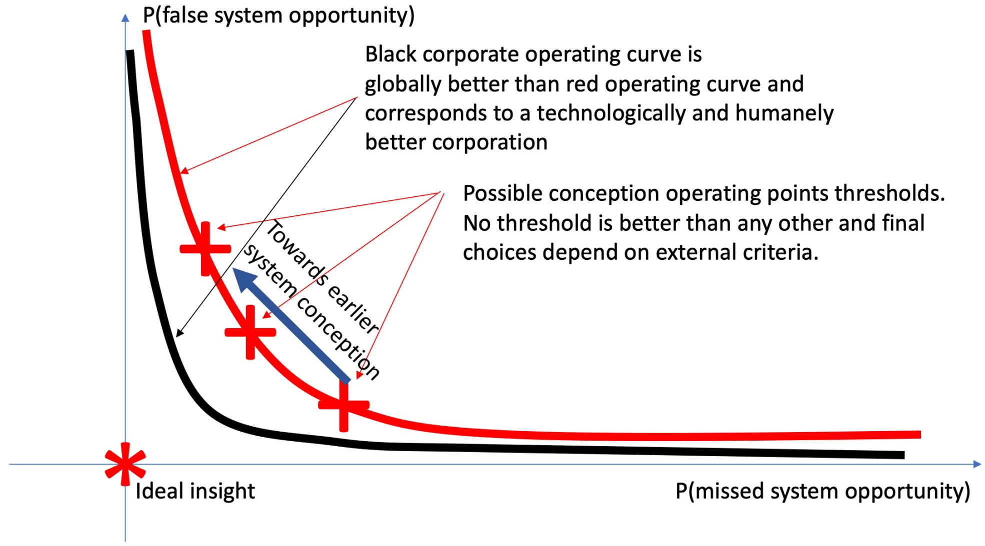

Fig. 2 Corporate Operating Curves and system conception trade-off

Except in rare cases (global dominance of one operating point over all others), the choice of an operating point, that is, the threshold to transform an initial conception into a full system development project is heavily dependent of external factors, such as the available budget, market share, social urgency, and other factors, not the least being the time horizon involved and sociological factors. Choosing to develop a self-driving car early may make sense for a young startup carried by reputation and an "investment wave", essentially motivated by the prospect of a costly purchase. However, such a decision may be senseless for a large automotive manufacturer and its surrounding socio-system, for whom maintaining a solid and established market share over the short term to guarantee employee paychecks and shareholder rewards is more important. Better informed corporations always end up taking better decisions, that is, decisions based on globally better corporate operating curves will, on average, become more profitable. A well-known case at the strategic level is the Koch conglomerate, which purchased entire oil pipelines to use them as reliable sensors of current and future fossil fuel needs, thereby enabling efficient pre-positioning to optimally anticipate evolving needs [19].

B. Designing

Designing is the phase where the systems engineer proceeds with planning the embodiment of the concept. Designing entails several phases, including: Defining the system's functional requirements and holding all parties accountable to them; refining the requirements and identifying "nonfunctional requirements", all the way down to identifying elements of hardware and software that, in principle, can be matched by existing components or must be custom-designed. Owing to the imperfection of this refinement task, control tasks of an analytic nature must be introduced for the purpose of making sure that lower level requirements and, eventually, hardware, are "consistent" with upper-level requirements. Each sub-phase is very important and contains many opportunities for development and maturation, now described in more detail.

1. Definition of functional requirements

This step is critical as it conditions the value and cost of the entire follow-up process. The functional requirements capture, in no ambiguous terms, what the system is about: For example, a functional requirement for a lighting system might be to produce at least 100 lumen continuously, while maintenance phases may occur no more often than twice per yer and not exceed three minutes. Functional requirements are the exact point at which the system developer and its client meet. For that reason, they have been taking an increasingly important role in the redaction of contracts and they form a legal basis upon which client and developer can base arguments in case of "downstream disagreements" on development and operating costs, reliability, time to obsolescence, etc. The client can then use the functional requirements to argue they have not been developed by the technology provider. The provider can use the same functional requirements to deflect the addition of unplanned functional requirements during the design process, also named "requirements creep", which has been denounced as a major source of uncontrolled cost increases for large engineering projects. To the extent legal language allows it, it is possible to bring nuance to the functional requirements, for example to rank them in order of importance, difficulty, cost, or other objective factors.

2. Extracting non-functional requirements that emerge from functional requirements, all the way to elemental levels that may be either purchased, custom-developed, or further studied: Architecting the system

Architecting is one of the most creative parts of the system design process: Indeed, the entire decomposition of the system's functional requirements into lower-level requirements down to its more mature component requires extensive knowledge, astuteness, and speed. Comparable to its civil engineering embodiment [20], a good architect possesses encyclopedic knowledge allowing him to connect all components to arrive at a final result that meets the functional requirements. A tribute to the creativity involved in architecture, successful systems are often named after their architects: For example, the Palais Brongniard, the Eiffel Tower, and the Palais Garnier, located in Paris, are all named after their architects. The recognition of architects as the pivotal elements of a given engineering feat is not universal: St Peter's Basilica does not bear the name of Michelangelo, and the Airbus A320 does not bear the name of Bernard Ziegler, although both individuals can be justifiably called the architects of their realizations, with Bernard Ziegler's key contribution being the fly-by-wire architecture of the A320 [21]. On the other hand, Russian tradition has always assigned the name of aircraft to their architects(Gourevich, Mikoyan, Sukhoi and Tupolev). The noble role of architecture, which mixes creative talent with uncompromising rigor, makes it the subject of some of the most modern textbooks devoted to systems engineering, such as [10]. A good architect is fundamentally a creative individual who knows how to apply a large knowledge base to think of simple architectures that very likely fit within the rules defined by technical, safety, financial, or social requirements.

3. Analytical system verification and validation

Often dubbed "verification and validation of the left leg of the Vee", these verification tasks are tedious but very necessary, because the consequences of a faulty consistency check at the "ideal level" can be disastrous when truly expensive steps involving hardware are undertaken. What system verification lacks in direct creativity constitutes, nevertheless, an infinite source of creativity for designing tools, progressively encompassing increasingly advanced system evaluation methods ranging from precise requirements tracing to formal software evaluation methods, finding process-based system design evaluations on the way. The profusion of new and legacy companies specialized in "analytical system verification" is a witness of the importance of these subjects in system design. The analytical verification and validation steps are typically done by using a mix of simulations and more synthetic, analytical methods. At the lowest levels of detail, one often finds well-identified elements: Detailed software requirements, electric, electronic and mechanical components. These components usually come with a specification sheet that summarizes their important characteristics. These specification sheets, whenever they are available, form a very attractive basis to initiate the analytic verification and validation of detailed system design. The analytical verification of complex assemblies of parts is particularly important when safety or huge capital expenditures are at stake. For example, structural engineers involved with the design of the wings of an aircraft spend huge efforts on the accurate prediction of the wing's destruction load: Too low, and the wing presents a possible safety liability [22]. Too high and the wing is too heavy, therefore inefficient. Accuracy is critical because the effective load to destruction test is performed when the "iron bird", or aircraft ground model is completed, and many additional expenses have been engaged.

C. Integrating

Building the system begins as soon as physical and executable software components are ordered by the system engineering company. System building consists of assembling the system itself, on the one hand, and regularly auditing and executing control operations on what has been built, on the other hand.

1. Parts purchase

The process of purchasing and assembling parts, whether for prototypes or for the first production system, signifies the first engagement of the overall system engineering process with hardware and operational elements. As such, it also marks the beginning of often-irreversible, far more expensive activities that find their justification in the foregoing design phase. When parts are purchased, it is very important that they comply with the specifications used during design. While many parts have a shelf life that largely exceeds the system's lifetime, a great deal of concern must be addressed regarding the parts with very short shelf life. This is the case of electronic components at the time when this paper is written. Short shelf life often implies "instability" of specifications and performance, which may have a very significant impact on the resulting product. This is the case with avionics, which abruptly faced the shortage of traditional single-core processors and had to switch to multicore systems successfully, but at great pains [23]. This instability can be felt even for the simplest of systems. For example, the educational system shown in Fig. **??** has suffered from the extreme instability and diversity of Arduino models and manufacturers.

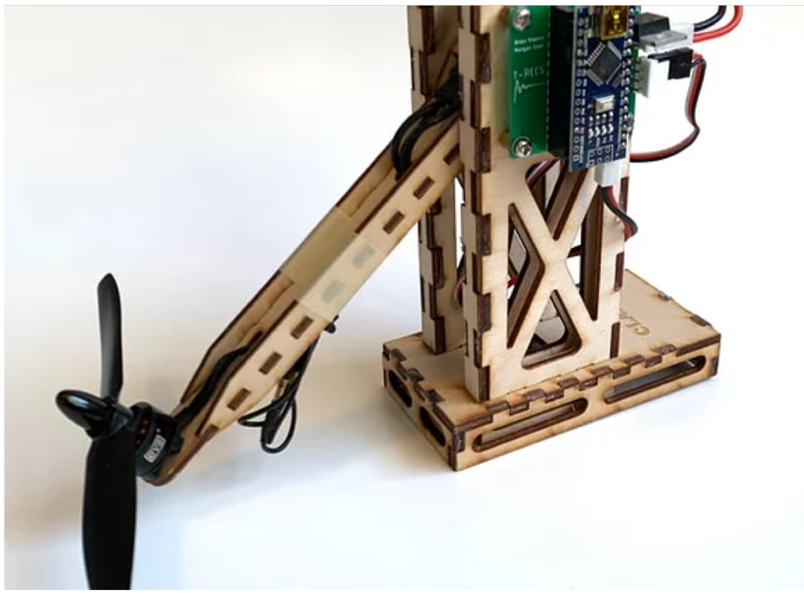

Fig. 3 Educational one-degree-of-freedom helicopter experiment with faulty Arduino board (courtesy Tangibles that Teach).

As a result, an entire batch of the manufactured systems was defective and could not be operated by its intended users. Perhaps surprisingly, software, which is, in appearance, easiest to change at the touch of a keyboard, has also shown to be one of the most stable elements of large complex systems, especially when the systems are safety-critical. This no-nonsense stability has been, in part, caused by the numerous accidents following careless handling of software development processes [24].

2. Parts assembly

Parts assembly, whether electrical or mechanical, is necessary for the purpose of putting a working system, or at least a credible system prototype, together. This apparently benign activity can often offer very serious challenges, and the most entertaining and critical incidents may be found in the aerospace literature. Often, the *interfaces* between assembled parts constitute an especially generous source of issues; for example, it was discovered during its construction that matching electric rods of the Airbus A380 were not aligned from one fuselage section to the next, owing to incompatible versions of the same computer-aided design software used at two different Airbus design offices [25].

Assembly may also occur, for example, at the interface of hardware and software, where multiple issues can also arise: For example, software programmed for floating-point arithmetic might meet fixed-point arithmetic processors or vice-versa, and upward-downward compatibility rules very often find themselves at a loss when mutual acceptance rules are stressed by safety requirements. Issues of insufficient hardware memory are also very common, especially if software requirement creep is not kept under control. Note such issues do not yield unsuccessful outcomes everywhere: Recently developed optimizing, qualified compilers [26] have generated a lot of interest to palliate the memory shortage of many airborne systems subject to very tight certification requirements.

3. Quality Control mechanisms

The state of the art in today's systems engineering requires extensive analysis and/or testing of all parts and assembled subsystems for the purpose of making sure that the expected behavior (presumably obtained during the "left side of the

Vee") is met by what is experimentally observed. The case of dissonance between expected and observed behavior may be problematic, and needs to be recorded and resolved in any case. The absence of comprehensive resolution and, possibly, redesign, may lead to far more costly issues later in the system engineering process. While intermediate control operations associated with subsystems are named "verification", the final control operation related to the assembled system is called "validation". With one person's system being his/her prime's subsystem, certain validation experiments in the context of a subcontractor may be verification experiments in the context of the prime contractor. In general, validation is excessively important because it implicitly refers to the terms of a contract made, implicitly or explicitly, between a client and a service or system provider. Usually, the validation test is the last step before delivery to the client, if the system is highly customized and one-of-a-kind, or before manufacturing if a large number of such systems is considered for production.

D. System operation, maintenance, and system retirement and memory

The operation of systems, especially complex systems, constitute an endless source of inspiration and investigation. Neglecting system retirement and not maintaining system memory is often ill-advised, in part because much can be learned from the autopsy and well-organized legacy of past systems, much of which can be "recycled" for the development of future systems, regardless of the technological innovation that may also come into its making.

1. System operation

The operation of a system is primarily aimed at bringing the intended utility, or functionality, to the client or clients who have ordered the system in the first place. However, and as the systems involved become increasingly complex, there is a growing quality of dialog between the client and the system developer throughout the system's life, with corresponding contractual modalities. Indeed, the operation of a system provides also a goldmine of information (and not only) to the technology developer, while growing insight from the developer may lead to very useful upgrades for the system. As systems grow in complexity, the contractual nature of the relation between manufacturers and operators changes dramatically. For example, aircraft engine manufacturers and their clients often operate via leasing mechanisms, which offer far more contractual flexibility than a simple sale / purchase transactions. In other cases, the manufacturer and its client operate at a very high level of fusion and share their Research and Development activities; such is the case with Thales and Australia's air traffic control services, Airservices Australia [27]. One positive aspect of this fused relationship is the ability for Australia to be far more agile than its Western world allies to welcome experimentation with futuristic concepts, such as Advanced Air Mobility, which benefits many corporations such as Boeing-owned Wisk [28]. Another example of never-ending relation between system manufacturers and their clients include the automotive world. Car recalls, most often for safety, are commonplace and usually follow the occurrence of non-immediate system operation issues and their resolution. If resolution is impossible or rejected by the manufacturer, continued vehicle operation may interfere with engineering ethics [29]. At airlines, it is well-known that some of the most interesting engineering jobs are in with Maintenance, Repairs and Operations (MRO), where many engineering and scientific tasks are assumed daily to identify aircraft malfunctions resulting from damage or wear and tear. In that regards, Delta is famous for running its own JT8D jet engine derivatives maintenance and repair operation because the airline kept flying DC-9 aircraft derivatives well past its domestic competitors in the USA as long as energy prices were low.

2. System retirement

System retirement consists of disposing of the system after its end of life. Depending on the nature of the system, retirement may be performed gradually or all at once. It may "leave traces behind" or not, depending on the attention spent by society to its environment, its safety, or other factors. One of the emblematic representatives of the issues surrounding system retirement is the civil nuclear industry, where the retirement of used combustible, on the one hand, and that of the nuclear plants, on the other hand was unanticipated at best. Other emblems include the story of the French sisterships aircraft carriers "Clémenceau" and "Foch", who experienced drifting across oceans after the countries who agreed to welcome them for demolition eventually refused them for environmental reasons [30, 31]. Retiring systems is also the source of multiple opportunities. This is the famously the case for the Ford model T, whose retired cars were scrupulously examined to identify parts that suffered no or little wear, with the goal of reducing their quality so their wear and tear characteristics be better aligned with those of the car as a whole. Similar comments hold today with the repetitive use of this procedure on the Lockheed C-130, a turboprop aircraft in the longest production run ever from 1954 onward. The large number of C-130 being retired while the production system is still in production keep providing very valuable information for incremental improvements of newer models, resulting in an airframe whose performances are difficult to match safely by brand new designs. Today's C-130 closest competitor is the Brazilian Embraer KC-390 [32]. The jury is still out, deciding the value of this aircraft relative to its precursor.

E. The role of the human

Humans are, more often than not, key components of a complex system. Either the system is designed to handle or manipulate humans, such as in aircraft, automobiles, or radiation therapy machines, or they are em operated by humans. In the first case, the immediate consequence is the imperative requirement for absolute safety, especially if a commercial transaction supports the exposure of the individual to the system. Indeed, the human life is considered sufficiently precious that it is not preferred to be given a value in engineering design considerations. Instead, the system is required to meet difficult and often unverifiable safety criteria, such as the requirement for accidents not to occur more than once per billion hours of operation for commercial aircraft. In the second case, when the human *operates* the system, the human is, de facto, *part of* the system. In that case, the human is often described as one of the most capable, most adaptable form of non-engineered systems. Teilhard de Chardin epitomizes this admiration for humanity in [11]. More recently, some societies faithfully maintain the culture of the super-hero, that is, the culture of a human being who is endowed with super-powers. This is particularly the case in the Anglo-Saxon world where super-heroes have been populating the minds of every child since the early 20th century: Think of Superman, Spiderman, and Zatana as single, "winner-takes-all" superheroes. More recently, teams of superheroes, such as The Incredibles, keep this myth of humanity's extra-ordinary skills while adapting to new socio-cultural norms. The enduring superhero myth might be tied to the great level of resistance expressed by the Anglo-Saxon industry to move ahead with altering the royal status of the human in many high-performance systems. The quasi-universal belief that humans are universally adaptable to new systems has led many people to over-rely on the human in ways that (i) abuse their technical abilities and (ii) grossly exaggerate their responsibility in case of system malfunction. The idealization of the human leads many of them to their loss, as beautifully allegorized by Mérimée [33] † and illustrated by numerous technological system failures.

1. Emerging engineering science

One of the most difficult, yet fascinating aspects of system design is the identification of the needs for engineering science. Much of engineering science is loosely driven by engineering needs and tends to look at natural and physical phenomena regardless of their eventual use. Engineering science driven by engineering challenges offers the perspective of immense satisfactions if successful. This "top-down" approach to Engineering science often meets with the far more common "bottom-up" approach favored by many engineering scientists. Top-down engineering science achievements include, for example, the understanding of aerodynamic and aeroelastic phenomena [34, 35] , or Lithium-Ion electricity storage [36]. Conversely, bottom-up science, such as the discovery of the double helix structure of the DNA [37] is progressively becoming engineering science and leading to innumerable innovations in bio-engineering. The same can be said about Newton's formulation of the classical mechanics hypothesis, whose application to engineering system design is now universal.

III. Mathematical and scientific theories

Mathematical and scientific theories are essential to the theory of systems engineering that will follow in this report. Theories can, in general, be seen as a coherent ensemble of results based on a limited number of mathematical axioms and many un-invalidated scientific hypotheses.

A. Mathematical theories

When only axioms are involved, the theory is mathematical in nature. The axioms supporting a given mathematical theory form a "small", coherent set of assumptions that are necessary to create a number of "results", expressed as theorems and supported by definitions. The definitions are arbitrary and never right nor wrong, except if they are grammatically incorrect. For example, the notion of equivalence relation between elements of a given set (a relation that is reflexive, symmetric, and transitive) is not immediately concerned with whether such a relation exists - it does -. Moreover, this definition is not direct, but is made by defining its three essential properties. In the worst case, the

†The author learned about the human idealization / apotheosis meaning of Mérimée's "La Vénus d'Ille" in 1992 from Michel Serres, then professor at Stanford University

definition of an object may lead to trivial sets, such as the empty set or the set of all objects. However, when chosen judiciously, definitions can create and name very useful objects, such as groups, rings, fields, integrals, derivatives, to name a few, and "compactify" notations by creating higher-order abstractions. Within the framework given by a set of axioms, theorems are then expressed as properties of the defined objects, and consist of the following essential elements: A set of assumptions, a property, and a logical reasoning leading the analyst from the assumptions to the property. For a candidate theorem to become a theorem, it is necessary that the assumptions be "correct", which means that they must be consistent with the initial set of axioms. Moreover, the logical path from the assumptions to the final statement must also be correct. Some mathematical theorems are quite famous for their value and their simplicity. For example, consider groups and the notion of right-inverse.

Theorem [38]: *When elements of a group have a right inverse, then they also have a left inverse.*

B. Scientific hypotheses

Scientific theories are mathematical theories based on a collection of axioms and complemented with "scientific laws" resulting from the observation of physical phenomena. For example, such scientific laws include Newton's universal gravitation law and classical dynamics, both of which are useful representations of universal phenomena. Newton law of dynamics casually implies

$$F = ma,$$

where F is the sum of the forces applied to a system of mass m, resulting in the acceleration a. Now, Newton's law, when it is defined this way, is far from complete and, perhaps, quite dangerous when put in the hands of untrained individuals. A more appropriate enunciation of Newton's law requires the definition of reference frames and a domain of validity, which looks far more like a definition than "a law". Once the notion of reference frame (a purely mathematical object per se) is defined, a more correct statement might look like

Definition: *An inertial reference frame is a reference frame where the identity*

$$F = ma$$

holds for any given set of particules. F is the sum of the external forces applied to the particules, m is the total mass of the particules (assumed constant), and a is the acceleration of the center of mass of the particules.

The "law" that follows can be expressed as a scientific finding:

Scientific finding: *There exist inertial reference frames, including the Earth (over short time periods), the Sun-Centered Celestial Equatorial Frame (SCCEF) (over much longer time periods), the International space station (over very short time period), and many others.*

Of course, we now since the late 19th century that Newton's scientific finding "works" when speeds do not exceed a small fraction of the speed of light, which is still quite acceptable for most engineering applications. One most notable exception is the GPS global positioning system. In essence, the syntax of a scientific finding is not very different from that of a theorem or an axiom, discussed thereafter.

1. Un-invalidated scientific hypotheses

Un-invalidated scientific hypotheses are the codification of experimental observations, where the codification, which looks like a theorem because it contains (i) assumptions that usually capture the range over which some kind of (ii) formalization or mathematical abstraction consistent with the experimental observations remains un-invalidated. If an experimental observation is inconsistent with the scientific hypothesis (or *model*) then the model is not valid anymore, and it must be modified according to standard procedures of either changing the assumptions or changing the "statement". For example, the Newtonian model has been changed in two different ways. First, the statement of $F = ma$ has been informally changed to $F = m\frac{d}{dt}v/\sqrt{1 - v^T v/c^2}$, where c is the speed of light. The second option has been to limit the domain of validity of Newton's law as described in the foregoing. There are very few scientific hypotheses that resits any form of invalidation attempts. Among the best known, general relativity offers a spectacular view of an un-invalidated theory, despite numerous attempts. Such attempts went beyond simple observation of nature to include sophisticated physics experiments, such as Gravity Probe B [39], on the one hand, and the later research by Maurice Allais ‡.

‡See `http://www.fondationmauriceallais.org/the-physicist/?lang=en`

C. Elements of mathematical and scientific theories

The elements of mathematical theories are many, and they include the introduction of the axioms, the definition of the appropriate quantities on an as-needed basis, and the constructions of theorems.

1. Axioms

The definition of mathematical axioms is a recurrent theme that keeps attracting the attention of mathematicians worldwide. A number of axioms have been postulated, and they constitute a field of great attention today, as axioms form the basis upon which all mathematical theories exist. One example of axioms is Peano's axioms [40], which are the axioms for the natural numbers and allow one to build the entire number theory, if they are completed by the axioms that allow one to manipulate sets, such as the axiom of choice, and possibly others. One of the essential characteristics of a set of axioms is *consistency*, that is, one axiom may not be contradicted by a logical combination of the axioms belonging to the set. The act of discovering axioms of practical interest is one that requires a significant level of creativity, and post people in the general population rely on axioms that they do not know are even there.

2. Definitions

Definitions are tools that help mathematical or scientific work work in several regards. One of them is to simply *name* objects that otherwise would take too long to write, or only for the sake of naming them. For example, considering a right triangle, it is useful to *name* the edges of the triangle. Indeed, the word "hypotenuse" takes many characters to write, and it is named as "h" or "c" in most right triangle diagrams (add picture here). The free-body diagram is one accepted way to perform accurate and relevant definitions and is especially useful for geometrical figures and engineering / physical diagrams. In other cases, the definition must be written explicitly according to the needs of the work to be performed.

3. Lemmas and Theorems

Lemmas and theorems are there to establish "results" based on available axiomatic theories. Theorems are essential to the development of important new elements within the framework offered by a given mathematical or scientific theory. In essence, theorems require three elements, denoted as (i) a claim, (ii) a set of hypotheses, and (iii) a logical set of arguments that is based solely on the hypotheses and establishes that the claim is true. There exists a complete taxonomy of theorems: Theorems, lemmas, propositions, and corollaries share the same structures, and provide further information about the function of the claim they contain: The claim may be intermediate for a lemma, a minor "result" for a proposition, the consequence of a theorem for a corollary. The name "Theorem" is usually reserved for important claims, such as Fermat's theorem, Pythagoras' theorem and others.

IV. Linking the system engineering process to mathematical and scientific theories

We now state the scientific hypothesis that is the central result of this paper, and we provide some elements that support this hypothesis. We then sketch possible possible invalidation strategies, although such an effort, which is most welcome and desirable, will require more focused and collective efforts for invalidation efforts to conclude about the validity of the hypothesis.

Hypothesis: *System engineering theory is a scientific theory and a system is a synthesized and un-invalidated, manufactured scientific hypothesis.*

A. General system design practices and processes

Many design methods come together with the celebrated Vee diagram shown in the foregoing. The Vee diagram recommends that all "electronic" and conceptual studies be completed before material goods be used and transformed according to the needs of the system. This no-nonsense approach is regularly challenged by alternate approaches, which include "spiral" and "agile" approaches. The spiral approach, which may also be called "do it, then do it right" [§], consists of repeatedly completing design cycles resulting in increasingly appropriate systems. This practice is common for small systems, such as those designed within universities and small contractors. However, they create risks as soon as significant hardware is involved, since hardware is considered expensive to purchase, test, and integrate. That said, there are also many possible benefits of a spiral approach, not the least being the early identification of scientific hypotheses

[§]The author would like to thank Professor David Miller at MIT for giving this convenient and illuminating description that must be answered before the system is completed. Such scientific questions and hypotheses are more prone to arise when dealing with demanding, one-of-a-kind projects, such as the De Haviland Comet (whose crashes launched much work on structural fatigue), the Handley Page O/400 aerial bomber during World War one (whose loss led to the study of flutter), etc. The early identification of such questions is essential to the development of fully mature, high-quality products, but finding them most often requires building mock-up versions of the system allowing researchers to quickly identified unanswered scientific questions. See [41] as an example. Answering these questions clearly is all the more important when the system is safety-critical or otherwise very valuable. This includes health care devices, weapon systems, advanced processors, and banking systems. Sometimes, the solution to the scientific problems may take years, or even decades. This is the case, for example, with composite materials, whose progressive introduction in aircraft has culminated when the Boeing 787 featured a fuselage mostly made with composite materials [42]. Often however, the identification of difficult scientific questions can lead the design team operating within a for-profit organization to set aside the chosen approach and opt for another, more predictable path. The current interest for and investment in self-driving cars has led to raising a multitude of very difficult engineering science questions and corresponding hypotheses, such as "can a fully automated car ever mix with human traffic", whose outcome, "yes" or "no" remains uncertain at best, and for which the necessity of a positive answer has already required modifying the hypothesis several times and delaying many deadlines, making the whole enterprise look increasingly like those associated with quantum computing and industrial power from atomic fusion. One consequence of an engineering system design program with many open scientific questions, such as *all* the foregoing enterprises, is the very high probability of cost overruns, perhaps exceeding initial budgets paid for by governments and investors by an order of magnitude or more. Historically, one of the best known expedients to support system developments involving the solution to many scientific problems, especially those with an experimental nature, has been the support of large organizations, such as governments, corporate entities with extra-ordinary financial means, or a mix of both. Bearing in mind that one of the core missions of governments is to ensure the well-being of its population, many of the programs supported by governments end up investing very significant amounts in defense systems. Such defense systems offer the additional advantage that they are usually more tolerant to failures and loss of personnel, so they offer the perspective for the faster development of operational systems, although these systems are mere prototypes for their use in commercial applications. That has certainly been the case for the introduction of jet engines, first used in military systems before being translated to commercial applications, and is initiated for autonomous ground vehicles [43].

B. Matching the Vee diagram with systems engineering science

Despite the limits of the Vee diagram discussed in the foregoing, it provides a useful support to our claimed systems engineering theory relative to a well-accepted and well-understood system engineering practice. An approximate correspondence diagram showing the relation of the various phases of Vee diagram to the essential elements of systems engineering science described in this report and its central hypothesis is shown in Fig. 4.

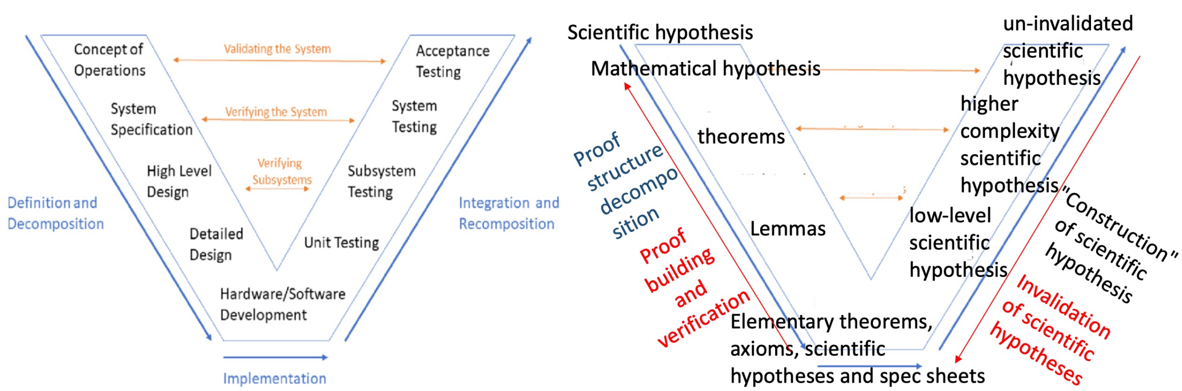

Fig. 4 System engineering theory as applied to the Vee diagram. Left: Vee diagram. Right: Projection of the systems engineering science hypothesis onto the Vee diagram.

In essence, the expression of the system's functional requirements is the formulation of a scientific hypothesis, "there exists an experimental construct that meets the client's needs". This hypothesis is immediately followed by the formulation of a mathematical hypothesis, which is similar in nature to "mathematical physics" and aims at proving that, in theory, the functional requirements/scientific hypothesis is true. Like any mathematical analysis, the mathematical hypothesis arising from the functional requirements might be proved, disproved, or remain unanswered ¶. Following the standard practice of mathematical physics, the proof, which is based on mathematical axioms, scientific laws, and their close cousins named parts specification sheets, must be extensively verified and often completed to avoid or clarify any errors and inconsistencies. They must also be optimized as quickly outlined below. The right side of the Vee, which is devoted to the construction of the physical system, matches on-to-one with the construction of the item that will confirm the scientific hypothesis "does there exist a system, either pre-existing or designed, that meets the functional requirements"? By alternating steps of verification and construction, the assembly process follows the proof of the mathematical hypothesis as it attempts to confirm experimentally the analytical prediction of the mathematical model. In that regard, the "right side of the Vee" is no less important than the "left side of the Vee". To illustrate the importance of that step, one might remember that no construct predicted by mathematical physics has resulted into the attribution of the Nobel Prize until experimental confirmation occurs [44].

C. Consequences of the hypothesis. "Left side of the Vee diagram"

Before embarking into arguments that validate or invalidate the foregoing hypothesis, it is interesting to go after its potential consequences and applications. Among such consequences, some are there to dispel, modernize, or formalize certain well-known practices. For example, there is considerable discussion about the value of the Vee diagram relative to other system engineering paradigms such as waterfall, spiral, and agile development methods. Bringing the discussion under the light of scientific evaluation criteria as introduced above can only help restore objectivity in what often turns into an intellectual brawl. Others open the possibility of system science to make, perhaps, fast-forward progress by using standard tools used in science for the purpose of supporting, or even improving, the system design process. Last, the insight acquired through a genuine science of system engineering is to possibly impact the lives of the many engineers and technicians involved in the process of designing systems by providing them with a clear map of the skills required to perform each phase of a complex system design in the context of a complete and clear framework.

1. System architecture

The phase that immediately follows (or runs concurrently with) the definition of the system functional requirements is the definition of a system architecture, which we define here as the activity whereby the functions are supported by a set of increasingly detailed requirements, and that ends where actual building blocks (computer programs, hardware elements) are identified. For "classical systems", where no new Engineering Science needs to be identified and performed, the identification of a system architecture is mathematically equivalent to establishing a mathematical proof based on assumptions (specification sheets for existing parts, theorems, and known science) and leading to a theorem, that is a mathematical representation of the system that meets a mathematical description of the requirements the functional requirements. Once this equivalence is identified, it opens the gates for a dialog between the definition of a system architecture and mathematics, applied mathematics, theoretical physics and computer science.

The establishment of system architectures can, for example, greatly benefit from the type of creativity exercised by good mathematicians, theoretical physicists, and computer scientists to establish proofs supporting their intended goals. The kind of constraints established by the functional requirements, and therefore the need to establish a specific theorem requires a high level of excellence in terms of desired creativity and rigor, compared with that required for "fooling around" with fantasy goals or, worse, when the goals are discovered at random during mathematical manipulations. The techniques used by pure mathematicians to establish theorems not unlike Fermat's theorem or Poincaré's conjectures, or those used by applied mathematicians to establish the necessary theorems to evaluate the dynamic behavior of flexible structures [45, 46] can be immensely useful in support of defining a good system architecture. A natural consequence of this observation is the need for system architects to have a solid mathematical culture, most notably a good familiarity with and appreciation for the construction of mathematical proofs, where a reasoning mistake, which is a normal part of the training of the mathematical mind, is usually far less costly than similar errors occurring during the design of a complex system.

In return, the kind of challenges presented by "proving" the functional requirements established by societal needs should not escape the attention of mathematicians, who might see interesting challenges to exercise their sagacity and, perhaps, the source of interesting new mathematical problems. In particular, the added value of proof simplification,

¶This last possibility is a natural consequence of K. Gödel's incompleteness theorems which is not always considered for what it is worth, becomes screaming in system architecture design. Indeed, given functional requirements, a simple architecture is far more valuable than a complex equivalent architecture, because it often leads to a reduction in the number of parts and complex interactions that all make a system less reliable and more expensive. In addition, one of the key aspects of engineered systems is the quest for the lowest possible cost given functional requirements. This demand brings a powerful and structuring component to the architecture exploration problem: At equivalent functionality level, an architecture that results into a system with lower development or operating costs is preferable. This powerful component gives a special status to mathematical techniques that aim at optimizing costs/benefit analyses in a diversity of frameworks. Generality of an optimization technique is therefore considered to be a key attribute, and explains the progressive and somewhat irreversible tendency of optimization techniques to belong to the essential toolset of systems engineering practice. While the generality and usefulness of optimization techniques cannot be denied, the process of optimizing, or simplifying, proofs runs far further and constitutes one area of its own right in mathematics and computer science. One of its possible impacts is how well such proofs can be welcome by the engineering community, and most notably the community whose job is to certify safety-critical systems.

From the "computer aided architecture design" perspective, many computer tools could be used to achieve the foregoing purposes. That includes many existing theorem provers, such as PVS [47], Coq [48], KeyMaera [49], Isabelle [**?**] and more recent efforts [50]. The key element for these tools to become productive is to "feed" them with the proper knowledge in terms of mathematical and physical science. While this task is necessarily tedious, it is also necessary and it requires bringing together mathematics, natural, and engineering scientists to cover the whole range of desired backgrounds. The net result will be a systematic codification of knowledge allowing system development users not to have the same level of knowledge as those currently in charge of large system development projects, and therefore reduce system development costs, including their duration. An additional benefit of emphasizing rigor (without, evidently, reducing creativity), is facilitating the necessary verification and validation of the resulting architecture - a pure theoretical product - by systematically linking low-level requirements, whether they are scientific facts or technical specifications, to their high-level counterparts, all the way up to the functional requirements themselves, thus bringing rigor and clarity to the entire "left side of the Vee" activities. In that regard, they represent a very significant way of the future for products such as Doors and Jama ‖. Another benefit of using such tools is the extraordinary analyses and decompositions that have been performed by the developers of these tools, in particular a systematic decomposition of the "process of proof", including automatic steps, such as skolemization, and highly creative steps, such as instantiation, and a clear definition of the logical environments used to establish these proofs, such as propositional logic, first-order logic, or higher-level, computationally more challenging logical frameworks [38]. Linear temporal logic and its close neighbor, signal-temporal logic, also come as a convenient logical environments to handle system temporal specifications. Most recently, the automated tools to establish proofs of theorems have been completed by new technologies, such as artificial intelligence. By looking over hundreds of proofs and training the appropriate artificial cognitive architecture, some researchers have been able to perform automated theorem discovery [51].

2. Open problems in system architecture designs

As in many system engineering problems, encountering open problems can be both a curse and a blessing, depending on how it is looked at. The open problems can be of a mathematical nature or of a scientific nature. These problems may appear as a "missing theorem or lemma", or it may also appear as a "missing scientific hypothesis", that is, a piece of un-invalidated scientific knowledge, during the architecture design effort. In both cases, the open problem prevents the architecture design task to be completed without extra effort. For example, many aircraft have been known to run into a problem when embedded software was found too big to fit within the available memory resulting into reduced capability compared with the proposed functional requirements. From an industrial systems engineering perspective, encountering open mathematical or scientific problems is a curse, because the event automatically increases the budget of the system development and, in extreme cases, it can put the entire system in jeopardy: What if the theorem to be proven is wrong or the scientific hypothesis cannot be validated? From a scientific perspective, and although the uncovered issues have genuine value, they may be judged of "limited interest" and not be addressed, perhaps unless appropriate funding is arranged to tackle these problems. Regardless, when a contract has been established around a given set of functional specifications, it is everyone's interest to identify open problems as early as possible and to solve them quickly if necessary. In that regard, some level of automation is also possible, based on the same theorem provers as already discussed in the foregoing. Unlike "closed" system design problems that may often rely on techniques such as combinatorial constraint satisfaction or operations research, the intrusion of "open" problems throws a monkey's

‖The author would like to thank Mr. Kemi Lewis for introducing the author to this requirements tracing tool.

wrench into the system design problem and corporations are naturally allergic to them. Discovering these issues early and therefore quickly is very important so as to evaluate their significance and take remedial action. Often, scientific or mathematical issues can be resolved by basic research. One major case was the development of Compcert, that was shown to provably produce safety-critical, optimized binary code. The memory savings offered by Compcert had immediate impact on relieving the pressure exercised on many aircraft due to limited available memory [26]. It is worthy to note that automatically establishing the theoretical *impossibility* of a given engineering design because it contradicts mathematical or scientific principles is extremely interesting for the same reasons, and such an exercise might be worth investigating in its own right.

D. Consequences of hypothesis: Right hand side of the "Vee diagram"

The right hand side of the "Vee diagram" shown in Fig. 1 can be seen as the experimental validation of the theoretical construct postulated by the "left side of the Vee" activity. While the "left hand side of the Vee" is essentially an analytical effort aimed at establishing the analytical plausibility of the system feasibility, the "right hand side of the Vee" is about demonstrating it experimentally, by building the system itself. If the system were a major scientific hypothesis, the Nobel prize could be obtained by its designers and builders by completing the "right hand side of the Vee", thus confirming the old saying "the proof in the [recipe for the] pudding is in the eating" **. Unlike mathematics and the activities pertaining to "left part of the Vee", where erroneous proofs do not result in significant damages other than possible contradictory debates, experimentally invalidated scientific hypotheses (that is, unverified or invalidated subsystems and system) can result in astronomical cost increases, the abandonment of the project, or human casualties if the system is safety-critical. In any case, much of the validation and testing activities are important elements of the trust that may be put into the system, and must be treated with the same level of care as any scientific experiment. That said, software often escapes standard scientific evaluation because it is a purely mathematical, man-made element of the system that often operates in closed-loop with the rest of the system. The mathematical nature of software places special demands on its verification, which in many regards is similar, if not identical to, the process of building mathematical proofs of proper behavior, and therefore lends itself to meeting the demands of an exhaustive investigation. Existing advisory documents, such as RTCA DO-178C [**?**] or RTCA DO-333 [**?**] recognize the existence of tools that perform exhaustive and mathematically consistent evaluations of software, such as Polyspace and Astrée [52]. Some of these tools, such as Polyspace [53], have now been widely accepted for low-level software analysis, looking for possible divide-by-zero and array index out-of-bound errors, for example. In addition, the recent introduction of qualified optimizing compilers [26] is making verifying the binary codes resulting from these compilers unnecessary in principle since all basic verification tasks may be made at the level of source code (mostly in C).

1. The human component

"Only" the human component of a system appears to be very resistant to the foregoing discussion. Indeed, the human being features astonishing adaptation capabilities supported by a very versatile body combined with a very adaptive neural structure. It has been determined, however, that the human is not as adaptive as unaware people might think. Many experiments, such as the "volley ball players" (cite experiment here) reveal the highly selective nature of human attention and the necessity for a human to abandon certain tasks, such as surveillance, to help him to focus his attention on one demanding task, such as tracking an object. It has also been observed that human judgement can be impaired by an over-tendency to prioritize recent observation over the cumulative information obtained from past measurements. Last, it is also well-known that human rote memory is highly culture-specific and it can be more limited than originally thought [54, 55]. Last, there is a great deal of unawareness of the tasks and requirements that may merge when a given function is allocated to the human operator. All these elements contribute to making it difficult to build a universal model of human sensing and decision making [56]. However, many reliable, validated task-specific models of the human operator [57] have been produced that can be used with confidence in new engineering system designs. Like any other element and function of a system, the ability to anticipate the need for establishing a new model of the human operator prior to "bending metal" can greatly reduce the risk of producing an unmanageable new system. One example of this is where one contributor to the 737 MAX deadly accidents was, among other reasons, an inappropriate model of modern human pilot capabilities and knowledge [58].

Another element of the "human component" that may be misunderstood is the legal aspects of human participation to the operation of a complex system. Unlike most components of the complex system, the personal liability of the

**The author would like to acknowledge Alan Epstein for uttering this sentence during an MIT meeting also attended by the author human operator might be engaged, should a faulty system behavior be noticed. In principle it is often the case that one of the human operators of a given system is responsible for it and also free to undertake any action aiming at preserving its safety. When accidents happen, his situation and the sharp departure of real life conditions from this ideal model can often result in significant legal problems whose may be produced several years later. For the sake of compactness and of the engineering focus of this paper, these legal aspects will not be pursued any further than stating they may be stipulated as part of the definition of the system's functional requirements and handled analytically. In aviation for example, it is often the case that mishaps be reported and handled with maximum confidentiality by trusted agencies, such as the US National Transportation Safety Board, to avoid possible criminalization and negative impact on the "learning from mistakes" mechanisms in place to improve system safety [59].

E. Validation and invalidation of hypothesis: Cross-validation with existing programs in Systems Engineering

The (in)validation of the proposed hypothesis is a long a tedious task that cannot be comprehended in this report and probably represents several years of observations. In fact, it can only be the result of a contradictory debate that is by definition the fruit of many collaborations. That said, a first invalidation effort may be performed by examining the course offerings of academic units devoted to systems engineering. For that purpose, only the academic units explicitly featuring the name "systems engineering" are considered. The investigated institutions are not, by far, exhaustive.

1. Systems Engineering at University 1

Consider for example the undergraduate and graduate curricula of the department of Industrial and Systems Engineering at University 1 [††] A cursory look at the graduate program indicates a great diversity of courses related to mathematical or approximately mathematical analyses, such as simulation and computational tools in the latter case. However, the presence of fundamental mathematics and the related understanding of elementary math, such as theorems and proofs, is not explicitly covered by the curriculum. Instead, the curriculum "jumps" into offering more specialized courses which, to our opinion, may blur the overarching image of Systems Engineering that alone, can give a clear justification to each and every courses. Two additional points stand out in the curriculum, which include (i) a careful treatment of human presence in engineered systems, and (ii) an initiation to engineering law. The presence of these two areas was initially absent from the report and human presence has been restored, thus showing the usefulness of the validation/ invalidation process used in this paper. Engineering law is essentially a social activity and is mentioned only in passing because this report focuses strictly on the scientific and analytical aspects of system design. Two areas of systems engineering appear to have been neglected that should naturally be included in this curriculum: One is a precise description of the systems engineering process (in essence an expanded version of this paper), and the second on is a systematic introduction to the fundamental processes of math (axiomatic theories of mathematics, concept of theorem, including proof design). Whether these skills or vistas are actually necessary in practical system design remains to be validated belongs to the realm of future work. However, the author believes that the synthetic image of system engineering provided by this report already carries the abstract elements allowing the students to initiate their undergraduate or graduate studies with a clear and compact road map allowing them to map their system engineering needs onto their needs for additional knowledge. This map may come handy as the number of disciplines required to address any industrial and system engineering project grows very fast, and students may only spend a limited number of years at any college.

2. Systems Engineering at University 2

The overall department of Industrial and Systems Engineering at University 2 offers no elements that invalidate the central hypothesis of this paper. It is worthy of further study to determine whether the proposed courses related to Engineering Economy are relevant to Systems Engineering as defined in this report, therefore invalidating its central hypothesis, or whether they constitute a useful complement to a solid engineering education. Certain elements of the hypothesis proposed in this report are missing from University 2's program, most notably as they relate to the presence of the human in the system, which can hardly be dismissed given the evidence regularly provided by the news. A closer look at University 2's undergraduate program reveals that no program explicitly mentions systems engineering as a declared track. Likewise, no MS program explicitly states "systems engineering". The same comment applies to the PhD program.

While not claiming that systems engineering is not evaluated as worthy of consideration by the department of

[††]The names of the universities studied as part of this effort have been withheld for confidentiality.

Industrial and Systems Engineering at University 2, it might be useful for this department to resolve this apparent contradiction and make systems more visible.

3. Systems Engineering at University 3

Systems Engineering at University 3 can be found in a program dedicate to systems design and management, a graduate program. The program offers several features mentioned in the foregoing, and contains several elements that complement what is discussed in this paper. On the one hand, a course is found on multi-disciplinary design optimization that acknowledges the very broad range of technical disciplines required to execute a design project successfully. Theoretically speaking, such an activity belongs to the "architecture design" introduced in the foregoing, and mathematically belongs to the realm of (automatically) proving and optimizing a given theorem. On the other hand, a course is offered that focuses on real options for product and systems design, and the question arises as to whether the topics addressed by this course invalidate the hypothesis presented in this report. We argue that with our definition of systems engineering, the hypothesis is not invalidated. Rather, Real Options for Product and Systems Design constitutes a peripheral set of knowledge that certainly forms a very desirable complement to System design as it helps financial people manage a portfolio of such systems and, as such, is completely relevant to the "management" part of the program. Likewise, the courses offered on project management and teamwork are essential to the implementation of successful system engineering designs and completely fit inside the name of the program. However, they offer the means, but not a concise, analytic, and scientific foundation for the system design process. Surprisingly perhaps, but consistent with the foregoing educational programs, the program does not forcefully introduce mathematics and the essential mechanisms of proof building. Validating the usefulness of all basic mathematical arguments to support any architecture design effort sits beyond the intent of this report but could form the basis for further investigations.

4. Systems Engineering as per the Accreditation Board for Engineering and Technology (ABET) criteria

ABET writes the following criteria for undergraduate engineering programs that include “systems (without other modifiers)” in their title: "There are no program-specific criteria beyond the General Criteria." Without going into further details, the general engineering criteria include courses in mathematics and sciences, the use of engineering tools, a form of general education (foreign languages, writing, other humanities), and a "capstone project", focused on a significant system design effort. Again, the requirements do not depart from those described in this report. However, a synthetic view of what constitutes "system engineering" is not present. That view would be well-covered by the hypothesis proposed in this report, assuming its essence remains after more extensive invalidation efforts.

5. Theory weaknesses: The human presence in large-scale systems engineering

Several arguments may be brought forward to further invalidate the proposed theory. Most of them are related to the relation of the human to the design process [‡‡]. Indeed, systems engineering science as described above tends to forget the human-centric nature of the system engineering process. This fact is shared with ALL mathematical and scientific theories however. Indeed, the myth of the lone and successful researcher is rapidly decreasing in favor of teams of researchers with the increased complexity of the problems at stake, and exchanges through conferences presentations, journal papers, and new media that include Youtube, Arxiv, and even Twitter at the time this paper is written. It will therefore not be studied further.

Another argument may criticize mathematics as a central tool for the design of high-performance systems. Indeed, many systems have been built and successfully operated without explicit reference to mathematics as it is known today. We contend, however, that projects that do not rely on rigorous mathematics at one point or another fail to perform optimally or are so unique, such as the Lockheed C-130 system currently, that the methods used are hardly transferrable to new projects. It is worthy to note that there are efforts in developing a formalism named soft computing to support the informal engineering and other activities that dominate everyday life. [60]

[‡‡]The author would like to thank Vincent Martinot-Lagarde for indicating his view that complex systems engineering is, first and foremost, a human adventure driven by $\alpha\gamma\alpha\pi\eta$, a form of love by humans for humans that appears in ancient Greek.

F. Initial recommendations for systems engineering education

1. General recommendations

Perhaps the greatest impact of the paper is to dismantle the commonly accepted idea that systems engineering and engineering science must compete for the attention of the researcher and the student. In particular, a close examination of systems engineering science reveals that system design includes *both* solid mathematical and experimental demands. As such, systems engineering projects (often called system design projects) more than fulfill the requirements for a demonstrated practice of engineering science. Given their strong attractiveness to motivated engineering students, "design-oriented projects" should definitely be encouraged: Why deprive the student from a pleasant and motivating way to discover science, and to learn the fundamental tenets of mathematics and creativity as added, implicit benefits? That said, it is clear that poorly understood design projects, such as designing an n-th version of Lockheed's C-130 "Hercules", is likely not to stress any specific aspects of engineering science or design, and can easily provide a very incomplete system engineering education ersatz via computer-aided design tools and approximate mathematics.

2. The importance of feedback

One key element of any engineered system is the ubiquitous presence of feedback. Indeed, bringing two physical elements together instantly creates a feedback relation between these two elements. Consider for example a pen and its cap of the kind shown in Fig. 6.

Fig. 5 A pen and its cap.

The pen and its cap form a system. As shown in Fig. 6, the pen and its cap are independent. Once the cap is put on the pen (the design of the pen allows the cap to be placed on either end of the pen), then they mutually interact in such a way that they remain together over most normal use conditions. More precisely, according to Newton's laws used in the appropriate reference frame, the two elements apply forces and moments equal in magnitude and opposite in moments in such a way that the system may be written as the traditional negative feedback loop shown in Fig. **??**. 6.

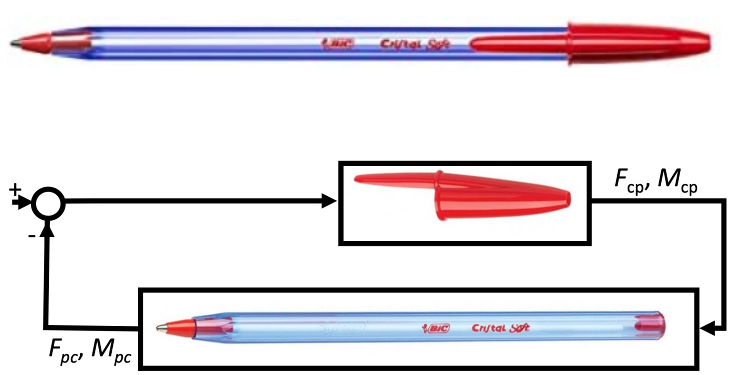

Fig. 6 A pen and its cap attached together, expressed as negative feedback loop. The forces and moments satisfy $F_{pc} = F_{cp}$ and $M_{pc} = M_{cp}$.

Note the dry friction and moments F and M are effectively acting as feedbacks as they allow the cap and the pen to stay bonded together despite a range of disturbing forces and moments. From thereon, multiple feedbacks loops can be added to the system that tend to make it increasingly "smart" as more sophisticated elements, especially computing elements, are included.

Conclusion

The author has formulated the possible existence of an authentic science of complex systems engineering. Inspired by the earlier works of Morin and Teilhard de Chardin, the discussion that followed brings initial elements that define this science. It then gives a preview of the practical applications of this science for the purpose of accelerating the design of large complex systems, on the one hand, and improving or perfecting the education of systems engineers, on the other hand. An initial attempt at invalidating the hypothesis is presented, although further invalidation efforts and the establishment of a healthy debate involving more players are necessary.

Dedication

The author dedicates this report to the late Professor Joseph Homer Saleh, whose input was essential to the genesis of the ideas expressed in this document.

References

[1] Silvestre de Sacy, A., *Exposé de la religion des Druzes*, Imprimerie Royale, Paris, 1838.

[2] Hergé, *Le Secret de la Licorne*, Casterman, 1943.

[3] Hergé, *Le Trésor de la Rackham le Rouge*, Casterman, 1945.

[4] Hergé, *La Vallée des Cobras*, Casterman, 1957.

[5] Hergé, *Le Testament de Mr Pump*, Casterman, 1951.

[6] Hergé, *Destination New York*, Casterman, 1951.

[7] Hergé, *Le Manitoba ne Répond plus*, Casterman, 1952.

[8] Anonymous, "North Korea's Kim Jong Un fetes nuclear scientists, holds celebration bash," `https://www.reuters.com/article/uk-northkorea-missiles-anniversary-idUKKCN1BL01J`, September 2017. Accessed: 2022-12-3.

[9] Kwon, K., "China Reaches New Milestone in Space-Based Quantum Communications," *Scientific American*, 2020.

[10] Selva, D., Cameron, B., and Crawley, E., *System Architecture: Strategy and Product Development for Complex Systems*, Pearson, 2015.

[11] Teilhard de Chardin, P., *The Phenomenon of Man*, Harper and Brothers, 1955.

[12] Morin, E., *La Méthode*, 1977-2004.

[13] von Bertalanffy, L., *General System Theory*, George Braziller, 1968.

[14] Ashby, W., "General Systems as a New Discipline," *General Systems Yearbook 3*, 1958, pp. 3–6.

[15] Crawley, E., Malmqvist, J., Östlund, S., and Brodeur, D. R., *Rethinking Engineering Education: The CDIO Approach*, Springer, 2007.

[16] Mellow, C., "The Rise and Fall and Rise of Iridium," *Smithsonian Magazine*, 2004.

[17] Mui, C., "How Kodak Failed," *Forbes*, 2012.

[18] Kuchar, J., "Methodology for Alerting-System Performance Evaluation," *AIAA J. Guidance, Control, and Dynamics*, Vol. 19, No. 2, 1996.

[19] Anonymous, "Summary of Koch Industries' industry facts," `https://www.sec.gov/Archives/edgar/data/41077/000119312505225697/dex996.htm`, 2005. Accessed: 2022-12-3.

[20] Silvestre de Sacy, J., *Alexandre-Théodore Brongniart 1739-1813. Sa vie, son oeuvre.*, Plon, 1940.

[21] Traverse, P., Lacaze, I., and Souyris, J., "Airbus Fly-by-Wire: A total approach to dependability," https://link.springer.com/content/pdf/10.1007/978-1-4020-8157-6_18.pdf, ???? Accessed: 2022-12-3.

[22] Anonymous, "Boeing 777 wing test," https://www.youtube.com/watch?v=Ai2HmvAXcU0, 1995. Accessed: 2022-12-3.

[23] Anonymous, "PU-3000 Achieves Multicore Processor Industry First for Commercial Avionics Systems," https://interactive.aviationtoday.com/pu-3000-achieves-multicore-processor-industry-first-for-commercial-avionics-systems/, March 2021. Accessed: 2022-12-3.

[24] Leveson, N. G., *Safeware: system safety and computers*, Addison-Wesley, 1995.

[25] Flottau, J., "What Went Wrong With The Airbus A380?" https://aviationweek.com/shownews/dubai-airshow/what-went-wrong-airbus-a380#:~:text=As_it_turned_out%2C_different,and_engineers_at_the_problem., November 2021. Accessed: 2022-12-3.

[26] Leroy, X., "Formal verification of a realistic compiler," *Communications of the ACM*, Vol. 52, No. 7, 2009, pp. 107–115.

[27] Anonymous, "Thales selected by Airservices Australia to carry out drone survaillance trials at Sydney airport," https://www.thalesgroup.com/en/worldwide/aerospace/press_release/thales-selected-airservices-australia-carry-out-drone, June 2021. Accessed: 2022-12-3.

[28] Hawkins, A., "Boeing-backed Wisk Aero reveals a four-seater autonomous air taxi," https://www.theverge.com/2022/10/3/23380518/wisk-aero-air-taxi-electric-autonomous-boeing-faa, October 2022. Accessed: 2022-12-3.

[29] Nader, R., *Unsafe at Any Speed*, Grossman Publishers, 1965.

[30] Anonymous, "UK to dismantle French asbestos-laden aircraft carrier," https://www.france24.com/en/20080701-uk-dismantle-french-asbestos-laden-aircraft-carrier-navy-shipbuilding, July 2008. Accessed: 2022-12-3.

[31] Moore, J., "Turkey finally refuses to welcome the former aircraft carrier Foch for its dismantling," https://ukdaily.news/turkey-finally-refuses-to-welcome-the-former-aircraft-carrier-foch-for-its-dismantling-64882.htmlg, August 2022. Accessed: 2022-12-3.

[32] Anonymous, "EMBRAER KC-390," http://www.military-today.com/aircraft/embraer_kc390.htm, ????. Accessed: 2022-12-3.

[33] Carpenter, S., "Metaphor and Madness in Mérimée's "La Vénus d'Ille," *Romance Notes*, Vol. 27, No. 1, 1986, pp. 75–80.

[34] Black, J., "Gustave Eiffel - pioneer of experimental aerodynamics," *The Aeronautical Journal (1968)*, Vol. 94, No. 937, 1990, p. 231–244. https://doi.org/10.1017/S0001924000022971.

[35] Bisplinghoff, R., Ashley, H., and Halfman, R., *Aeroelasticity*, Addison-Wesley Publishing Company, 1955.

[36] Whittingham, M., "Electrical Energy Storage and Intercalation Chemistry," *Science*, , No. 4244, 1976, pp. 1126–1127.

[37] Watson, J., and Crick, F., "Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid," *Nature*, , No. 171, 1953, pp. 737–738.

[38] Peled, D. A., *Software Reliability Methods*, Springer, 2001.

[39] Buchman, S., Everitt, C., Parkinson, B., Turneaure, J., DeBra, D., Bardas, D., Bencze, W., Brumley, R., Gill, D., Gutt, G., Gwo, D., Keiser, G., Lipa, J., Lockhart, J., Mester, J., Muhlfelder, B., Taber, M., Wang, S., Xiao, Y., and Zhou, P., "The Gravity Probe B Relativity Mission," *Advances in Space Research*, Vol. 25, No. 6, 2000, pp. 1177–1180. https://doi.org/https://doi.org/10.1016/S0273-1177(99)00982-5, URL https://www.sciencedirect.com/science/article/pii/S0273117799009825, fundamental Physics in Space.

[40] Skolem, T., "Peano's Axioms and Models of Arithmetic," *Studies in Logic and the Foundations of Mathematics*, Vol. 16, 1955, pp. 1–14.

[41] Howell, E., "SpaceX's Grasshopper: Reusable Rocket Prototype," *Space.com*, 2016, pp. 1–14.

[42] Guizzo, E., "Carbon takeoff [carbon fiber composites]," *IEEE Spectrum*, Vol. 43, No. 1, 2006, pp. 26–29. https://doi.org/10.1109/MSPEC.2006.1572347.

[43] Hope, G., "US Military Tests Off-Road Autonomous Vehicles," *IoT World Today*, 2022. URL https://www.iotworldtoday.com/2022/09/27/us-military-tests-off-road-autonomous-vehicles/#.

[44] Carr, F., “Here’s why Stephen Hawking Never Won the Nobel Prize in Physics,” *Time USA*, 2018. URL https://time.com/5198933/why-stephen-hawking-never-won-nobel/.

[45] Doyle, J., “Analysis of Feedback Systems with Structured Uncertainties,” Vol. 129-D, No. 6, 1982, pp. 242–250.

[46] Balas, G. J., and Doyle, J. C., “Control of Lightly Damped, Flexible Modes in the Controller Crossover Region,” *AIAA J. of Guidance, Control, and Dynamic*, Vol. 17, No. 2, 1994, pp. 370–377.

[47] Owre, S., Rushby, J. M., and Shankar, N., “PVS: A prototype verification system,” Springer, Berlin / Heidelberg, 1992, pp. 748–752.

[48] “The Coq proof assistant: reference manual, version 6.1,” Tech. Rep. 203, Grenoble, FRANCE, 1997.

[49] Platzer, A., and Quesel, J.-D., “KeYmaera: A Hybrid Theorem Prover for Hybrid Systems (System Description),” *Automated Reasoning*, edited by A. Armando, P. Baumgartner, and G. Dowek, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 171–178.

[50] Ganesalingam, M., and Gowers, W. T., “A Fully Automatic Theorem Prover with Human-Style Output,” *Journal of Automated Reasoning*, Vol. 58, No. 2, 2017, pp. 253–291. URL https://doi.org/10.1007/s10817-016-9377-1.

[51] Kaliszyk, C., and Urban, J., “Learning-Assisted Automated Reasoning with Flyspeck,” *Journal of Automated Reasoning*, Vol. 53, 2014, pp. 173–213. URL https://doi.org/10.1007/s10817-014-9303-3.

[52] Cousot, P., Cousot, R., Feret, J., Mauborgne, L., Miné, A., Monniaux, D., and Rival, X., “The ASTREÉ Analyzer,” *Programming Languages and Systems*, edited by M. Sagiv, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 21–30.

[53] Anonymous, “Polyspace - MATLAB & Simulink,” , ????. URL https://www.mathworks.com/products/polyspace.html, accessed: 2022-12-3.

[54] Parasuraman, R., Sheridan, T., and Wickens, C., “A model for types and levels of human interaction with automation,” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, Vol. 30, No. 3, 2000, pp. 286–297. https://doi.org/10.1109/3468.844354.

[55] E. Salas, E., *Situational Awareness*, Routledge, 2011. URL https://doi.org/10.4324/9781315087924.

[56] Corker, K., and Smith, B., “An architecture and model for cognitive engineering simulation analysis-Application to advanced aviation automation,” *9th Computing in Aerospace Conference*, 1993, p. 4660.

[57] McRuer, D., and Jex, H., “A Review of Quasi-Linear Pilot Models,” *IEEE Transactions on Human Factors in Electronics*, Vol. HFE-8, No. 3, 1967, pp. 231–249. https://doi.org/10.1109/THFE.1967.234304.

[58] Endsley, M. R., “Human Factors & Aviation Safety - Testimony to the United States House of Representatives - Hearing on Boeing 737-Max 8 Crashes,” , 2019. URL https://www.hfes.org/Portals/0/Documents/Human_Factors_and_the_Boeing_737-Max8-FINAL.pdf, accessed: 2022-12-3.

[59] Beamish, R., “NASA releases data on pilot survey. Format makes analysis difficult,” , January 2008. URL http://archive.boston.com/news/nation/articles/2008/01/01/nasa_releases_data_on_pilot_survey/?page=1, accessed: 2022-12-3.

[60] Zadeh, L. A., “Computing with Words and perceptions—A paradigm shift,” *2009 IEEE International Conference on Information Reuse & Integration*, 2009, pp. viii–x. https://doi.org/10.1109/IRI.2009.5211627.