A Comparison of STPA and the ARP 4761 Safety Assessment Process[1]

MIT Technical Report[2]

Nancy Leveson, MIT
Chris Wilkinson, Honeywell
Cody Fleming, MIT
John Thomas, MIT
Ian Tracy, MIT

June, 2014

[1] This research was supported in part by the SSAT (System-Wide Safety and Assurance Technology) component of the NASA Aviation Safety Program under contract NNL10AA13C. The views and conclusions in this report are those of the authors alone. Approval by NASA as a NASA Technical Report is still in process.

[2] Electronic versions of this technical report can be obtained from http://sunnyday.mit.edu/papers/ARP4761-Comparison-Report-final-1.pdf

Table of Contents

1. Introduction

The goal of this report is to compare the approach widely used to assess and certify aircraft with a new, systems-theoretic hazard analysis technique called STPA and to determine whether there are important factors missing from the commonly used approach.

First a little background is needed for those not in this industry. 14CFR (Code of Federal Regulations) specifies the airworthiness regulations applicable to transport category aircraft. 14CFR/CS 25.1309 is the subchapter of 14 CFR/CS describing the rules applicable to equipment, systems, and installations for the FAA (Federal Aviation Administration) and EASA (European Aviation Safety Agency). The FAA/EASA does not require specific practices for certification but issues advisory circulars that recognize acceptable means for developing and certifying an aircraft. One advisory circular, AC20-174, recognizes SAE Aerospace Recommended Practice (ARP) 4754A as an acceptable means for establishing a development assurance process. ARP 4754A documents a process that can be used throughout the requirements development cycle (top half of Figure 1.1). SAE ARP 4761, describing a safety assessment process, is a supporting part of the larger development process described by ARP 4754A. Equivalent and harmonized European regulations and guidance are provided by EASA.

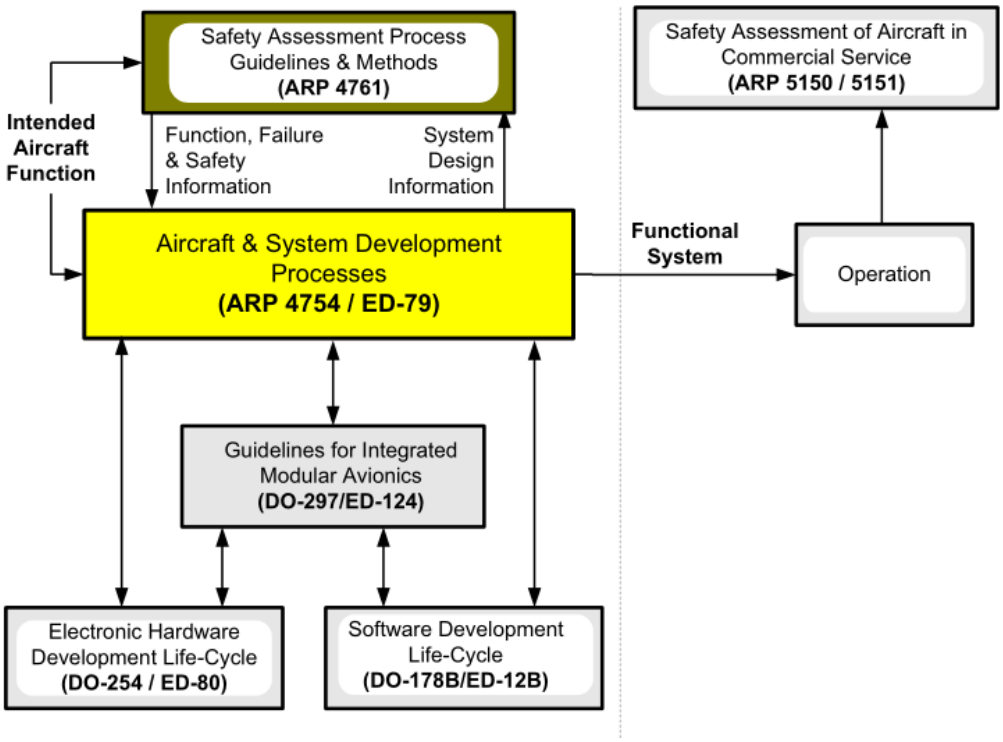

Figure 1.1: Guideline documents covering aircraft system development [SAE ARP 4754A]. (ARP 4754A and DO-178C have since been published and are now used.)

There is no advisory circular that specifically recognizes ARP 4761 as an industry standard for conducting a safety assessment process. Instead, AC 25.1309 describes various acceptable means for showing compliance with FAA safety requirements and the airworthiness regulations. Nonetheless, ARP 4761 is a supporting part of the larger systems development process described by ARP 4754A, is widely used, and may be invoked by the regulators on a project by project basis through Issue Papers (FAA) or Certification Review Items (EASA).

Although it can be argued that ARP 4754A has been effective on the relatively simple electromechanical aircraft designs that have prevailed in the industry, the use of software as well as complexity are increasing. The traditional hazard analysis methods described in ARP 4761 are no longer as effective on software-intensive systems where accidents may result from unsafe interactions among the components and not just component failures.

STPA (System-Theoretic Process Analysis) is a new hazard analysis method based on systems theory rather than reliability theory [Leveson, 2012; Leveson, 2013]. STPA has as its foundation a new accident causality model that extends the prevailing view of accidents as caused by component failures to include additional causes such as system design errors (including software and system requirements errors), human error considered as more than just a random "failure," and various types of systemic accident causes. As such, STPA is potentially more powerful than the traditional hazard analysis methods and approach used in ARP 4761. A goal of this paper is to provide evidence to support this hypothesis by comparing the approach and results of using the process described in ARP 4761 with the results of STPA.

First the approach outlined in ARP 4761 is described using the Wheel Brake System (WBS) example in the standard. Then an STPA analysis is shown for the same system. Finally, the two approaches are compared. While the WBS example is relatively simple and primarily electromechanical and thus does not demonstrate the power of STPA on more complex, software-intensive aircraft components, interesting comparisons still are possible. In addition, use of the example in the standard eliminates the potential for claims that we misunderstood or did not do an adequate job using the ARP 4761 approach.

2. The ARP 4761 Generic Commercial Aircraft Wheel Brake System Example

As ARP 4761 explains, the wheel braking system is installed on the two main landing gears to provide safe retardation of the aircraft during park, pushback, taxi, takeoff (and rejected takeoff (RTO)) and landing phase. Figure 2.1 shows the phases of flight including some in which the wheel braking system is used. The wheel braking system also provides differential braking for directional control, stops the wheel rotation upon gear retraction after take-off, and prevents aircraft motion when parked.

Most of the analysis in ARP 4761 is performed with a reduced scope for the purpose of demonstration. The analysis considers a single wheel system with anti-skid functionality. The controls required for differential braking are not described and not analyzed. The physical braking system and the functionality of the Brake System Control Unit (BSCU) are also analyzed, but only at the basic level of detail presented in ARP 4761. This report uses the same scope to demonstrate STPA. In addition, to show how human and software behavior can be included in the same analysis, STPA is also applied to the basic automation and pilot controls described in ARP 4761.

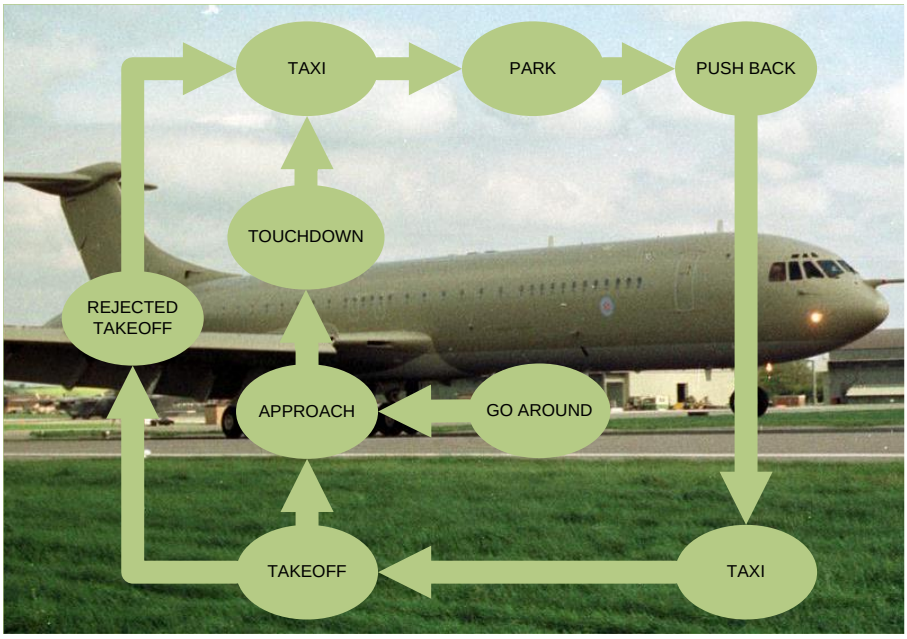

Figure 2.1: WBS Operational Phases

Figure 2.2 shows the WBS diagram from ARP 4761. The following is a summarized description of the system directly from Appendix L3 in ARP 4761:

"The Wheel Brake System is installed on the two main landing gears. Braking the main gear wheels is used to provide safe retardation of the aircraft during taxi and landing phase, and in the event of a rejected take-off. The wheel brake system is shown in Figure 3.0-1. The wheel brakes also prevent unintended aircraft motion when parked, and may be used to provide differential braking for aircraft directional control. A secondary function of the wheel brake system is to stop main gear wheel rotation upon gear retraction.

Braking on the ground is commanded either manually, via brake pedals, or automatically (Autobrake) without the need for pedal application. The Autobrake function allows the pilot to pre-arm the deceleration rate prior to takeoff or landing. [One feature of the Autobrake system typically engages pressurized wheel braking upon touchdown to a landing surface. During rollout deceleration, depression of the brake pedals will transfer braking control back to the pilot.] Autobrake is only available with the NORMAL braking system.

The eight main gear wheels have multi-disc carbon brakes. Based on the requirement that loss of all wheel braking is less probable than 5E-7 per flight, a design decision was made that each wheel has a brake assembly operated by two independent sets of hydraulic pistons. One set is operated from the GREEN hydraulic supply and is used in the NORMAL braking mode. The Alternate Mode is on standby and is selected automatically when the NORMAL system fails. It is operated independently using the BLUE hydraulic power supply" [SAE ARP 4761, p.190-191].

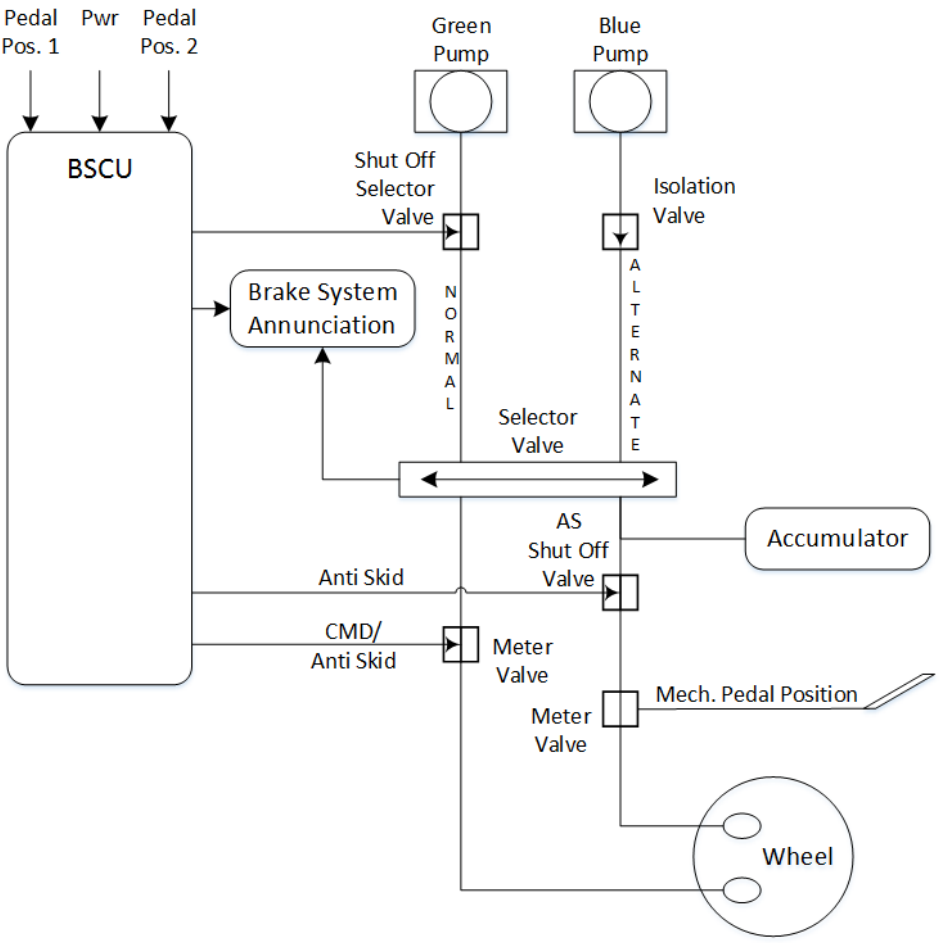

Figure 2.2: Preliminary Wheel Brake System Diagram [SAE ARP 4761 App L, Fig 3.0-1 p. 192]

3. The ARP 4761 Safety Assessment Process

The approach described in ARP 4761 focuses on failures and probabilistic risk assessment (PRA). It illustrates the use of failure-based hazard analysis techniques such as fault tree analysis (FTA), failure modes and effects analysis (FMEA), and others. The goal of the safety assessment process is primarily quantitative. There may be qualitative aspects, such as considering the assignment of functional and item Development Assurance Levels (FDAL/IDAL), HIRF, lightning strike, etc., but they are in addition to the quantitative analysis of failures. In addition, DALs are used to justify the probabilistic values assumed for software in the overall quantitative analysis.

The ARP 4761 process has three parts—the Functional Hazard Analysis, the Preliminary System Safety Analysis, and the System Safety Analysis—which are performed at each relevant level of abstraction (or hierarchical level) for the system under study.

Functional Hazard Analysis (FHA): The FHA is conducted at the beginning of the aircraft development cycle. There are two levels of FHA: the aircraft level FHA and the system level FHA. The aircraft-level FHA identifies and classifies the failure conditions associated with the aircraft level functions. The classification of these failure conditions establishes the safety requirements that an aircraft must meet. The goal is to identify each failure condition along with the rationale for its severity classification. A

standard risk assessment matrix is used, shown in Table 3.1. Both the failure of single and combinations of aircraft functions are considered. The failure condition severity determines the development assurance level allocated to the subsystem.

TABLE 3.1: Failure Condition Severity as Related to Probability Objectives and Assurance Levels [SAE ARP 4761, p. 14]

Probability (Quantitative)	Per flight hour 1.0 1.0E-3 1.0E-5 1.0E-7 1.0E-9					
Probability (Descriptive)	FAA	Probable		Improbable		Extremely Improbable
	JAA	Frequent	Reasonably Probable	Remote	Extremely Remote	Extremely Improbable
Failure Condition Severity Classification	FAA	Minor		Major	Severe Major	Catastrophic
	JAA	Minor		Major	Hazardous	Catastrophic
Failure Condition Effect	FAA & JAA	• slight reduction in safety margins • slight increase in crew workload • some inconvenience to occupants		• significant reduction in safety margins or functional capabilities • significant increase In crew workload or in conditions impairing crew efficiency • some discomfort to occupants	• large reduction in safety margins or functional capabilities • higher workload or physical distress such that the crew could not be relied upon to perform tasks accurately or completely • adverse effects upon occupants	• all failure conditions which prevent continued safe flight and landing
Development Assurance Level	ARP 4754	Level D		Level C	Level B	Level A

Note: A " No Safety Effect" Development Assurance Level E exists which may span any probability range.

Later in the development process, the architectural design process allocates the aircraft-level functions to particular subsystems. A system-level [3] FHA considers the failures or combination of system or subsystem failures that affect the aircraft-level functions.

The same procedure is used at both the aircraft-level and the system level:

1. Identification of all the functions associated with the level under study
2. Identification and description of failure conditions associated with these functions, considering single and multiple failures in normal and degraded environments.

[3] In ARP 4761, the "system-level" is the aircraft component or subcomponent level.

3. Determination of the effects of the failure conditions.
4. Classification of failure condition effects on the aircraft (catastrophic, severe-major/hazardous, major, minor, and no safety effects.
5. Assignment of requirements to the failure conditions to be considered at the lower level classification.
6. Identification of the method used to verify compliance with the failure condition requirements.

The aircraft-level probabilistic risk analysis is derived from the FHA and used to budget (i.e. allocate) failure rates to the various sub-systems such that the aircraft as a whole meets the 14CFR/CS requirement in terms of the probabilities given in the guidance (e.g. AC 25.1309). This process is iterative as the objective is to allocate achievable failure rate allocations to the subsystems and hence may require aircraft architectural mitigations (essentially additional AND gates in the fault trees). The end result of this FHA is a list of functions grouped by subsystem and the failure rate allocated to the group. Severity determines the development assurance level allocated to the subsystem.

A fault tree analysis (FTA), Dependence Diagram (DD), Markov Analysis (MA), or other analysis methods can be used to derive lower level requirements from those identified in the FHA. Figure 3.1 shows the overall relationship between the FHA, FTA, and FMEA. The FHAs generate the top level events in the FTA. The quantitative results from FMEA and FTA feed back into aircraft level FHAs to show compliance with numerical safety requirements for the identified severity classification. In the same way, FHA is traced down into preliminary design and detailed design, with fault tree and other analyses used to provide quantitative results.

The FHA also establishes derived safety requirements needed to limit the effects of function failure. These derived requirements may affect the failure condition classification and may include such things as design constraints, annunciation of failure conditions, recommended flight crew or maintenance action, etc.

Once the high-level requirements have been identified, they may be used to generate lower-level requirements as part of the PSSA process. The process is continued, with reiteration, until the design process is complete.

Preliminary System Safety Assessment (PSSA): The PSSA is used to complete the failure conditions list and the corresponding safety requirements. It involves a "systematic examination of a proposed system architecture to determine how failures can lead to the functional hazards identified by the FHA and how the FHA requirements can be met" [SAE APR 4761, p. 40]. Note the assumption made that only failures can lead to hazards and the converse that the absence of failures implies the absence of hazards. We will show that this assumption is incorrect.

For each of the system functions, the FHA identifies functional failure conditions that are potentially hazardous when considered in the context of possible environmental conditions (wet, icy, crosswind etc.) and flight phase (takeoff, land, taxi etc.) taken in combination. Again probabilistic risk analysis is performed at the sub-system(s) level(s) to show that the failure probability meets the requirement passed down from the aircraft level FHA.

Because a subsystem may consist of further subsystems, the process is continued to decompose the failure rate allocations to the component subsystems. There may thus be multiple PSSA's that have to be integrated into one "master" PSSA for the complete subsystem.

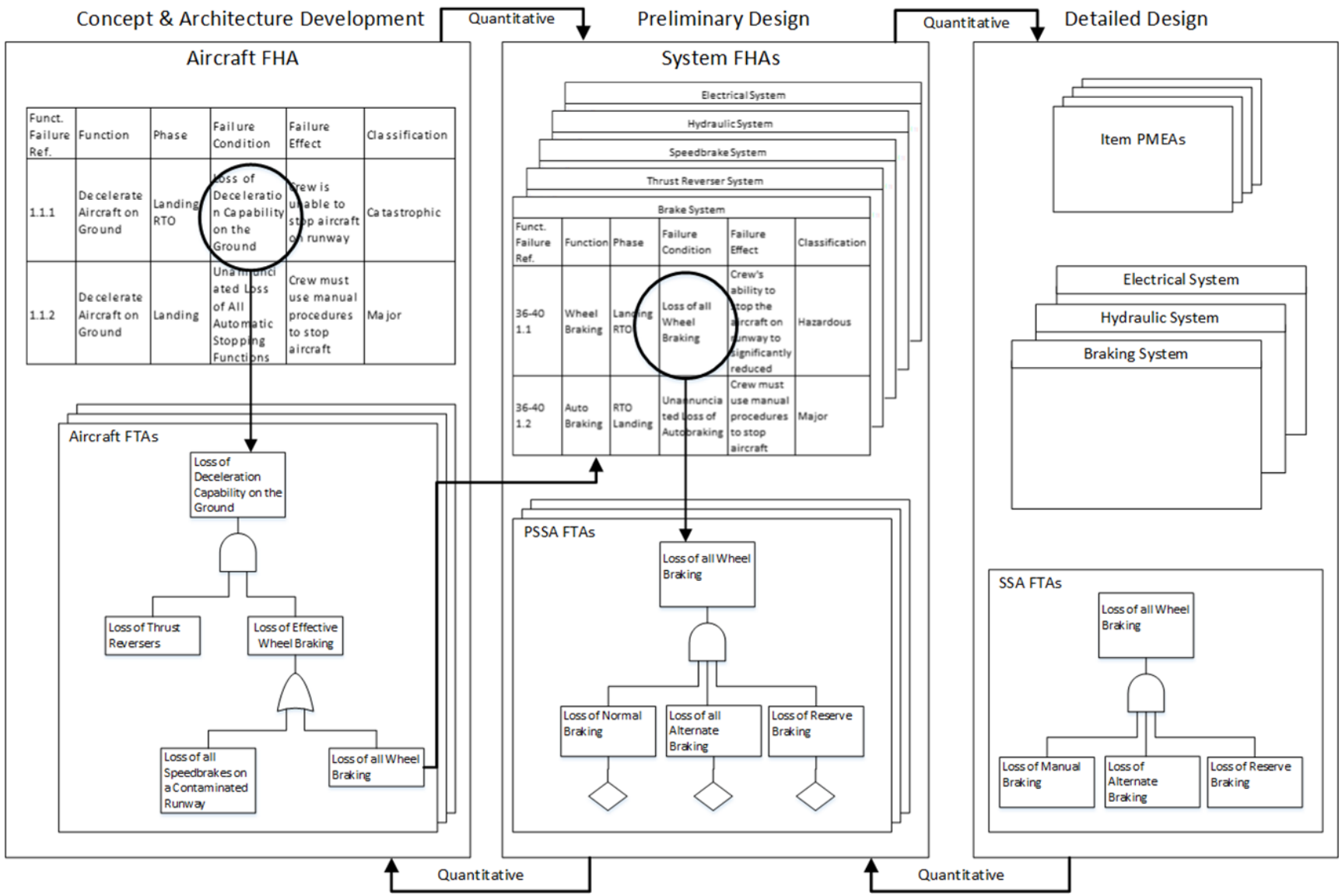

Figure 3.1: The relationship between FHA, FTA, and FMEA [SAE ARP 4761, p. 18]

While either quantitative or qualitative analysis methods are allowed in theory by ARP 4761, in practice and in the examples included, the PSSA uses quantitative FTA or similar PRA techniques to show that, by combining system failure probabilities using Boolean logic, each top-level functional hazard is less probable than its budgeted probability. It is customary (using Markov models) to assume a constant failure rate (exponential distribution), and therefore failure rates are specified in terms of a lambda value. The outputs of higher-level PSSAs form the basis for lower-level PSSAs. Common Cause Analyses (CCAs) are also conducted in order to substantiate independence claims made in the FHA.

System Safety Assessment (SSA): Once the PSSA is completed at multiple levels of abstraction for the system, the final step of System Safety Assessment (SSA) begins. The SSA is a "[bottom-up] verification that the implemented design meets both the qualitative and quantitative safety requirements…defined in [both] the FHA and PSSA" [SAE ARP 4761, p. 21]. This step is performed concurrently with the detailed design of the subsystem [SAE ARP 4761, p. 21]. For each PSSA carried out at any level, there should be a corresponding SSA

As with the PSSA, the SSA consists of failure-based, probabilistic risk analysis methods. Fault trees are used to ensure that both "qualitative and quantitative requirements and objectives associated with the failure condition can be met by the proposed system architecture and budgeted failure probabilities" [SAE ARP 4761, p. 43]. Common Cause Analyses (CCAs) are reviewed to ensure that the independence requirements generated during the PSSA have been satisfied.

It is possible that at any of the three steps, the need for additional requirements will be identified in order to achieve the aircraft level probability necessitating a redesign at some level of the hierarchy.

Development Assurance Levels: Because software and other components of an aircraft may not lend themselves to probabilistic assessment, an "item development assurance level" or IDAL is assigned to each component, depending on the safety-criticality of the component. The primary development assurance level standards published by RTCA[4] are:

- DO-178C: Software Considerations In Airborne Systems And Equipment Certification [RTCA, 2012]
- DO-278: Guidelines For Communication, Navigation, Surveillance, And Air Traffic Management (CNS/ATM) Systems Software Integrity Assurance
- DO-254: Design Assurance Guidance for Airborne Electronic Hardware [RTCA, 2000]. This guidance is applicable only to programmable devices. There is currently no guidance available for hardware in general.
- DO-297: Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations

These documents, together with the system design and safety assessment documents developed by the SAE (ARP-4754A and ARP-4761), provides the primary guidance material used by certification applicants. These RTCA documents are not regulations as such but are often invoked by reference in advisory circulars, policy memos, orders etc.

[4] RTCA, Inc. is a private, not-for-profit corporation that develops consensus-based recommendations regarding communications, navigation, surveillance, and air traffic management (CNS/ATM) system issues. RTCA functions as a Federal Advisory Committee. Its recommendations are used by the Federal Aviation Administration (FAA) as the basis for policy, program, and regulatory decisions and by the private sector as the basis for development, investment and other business decisions. RTCA has facilitated the development of design assurance documents for software and hardware that form the basis for the current certification framework.

DO-178C and DO-254 provide guidance to achieve design assurance of flight software and hardware respectively from Level A (flight critical) to Level E (no safety impact) inclusive. Note that both assume that there is an existing set of requirements and that these requirements are consistent and complete. The design process then is characterized as implementing the requirements (as given) and generating derived requirements as necessary to pass on to lower level design activities. This process proceeds tree-like down to the lowest level of design, typically a software module or circuit card. Verification proceeds in the reverse direction, verifying that each requirement (and derived requirement) is satisfied using test, analysis or review as appropriate to the verification object. There are inherent feedback loops because problems can be discovered and corrected at the hardware, software or integration level. The process completes when all open problem reports are closed.

The reality is much more complicated, however. Many sub-elements undergo design, verification and integration simultaneously and are brought together into increasingly large sub-elements until the final product is obtained. Requirements deficiencies can become apparent at any time, or indeed following an accident, which must be fed back to the systems process for resolution, allocation and redesign.

4. The ARP 4761Wheel Brake System Analysis

This section presents the analysis of the WBS in ARP 4761 Appendix L. Readers familiar with this analysis can skip to Section 5.

4.1 Functional Hazard Analysis (FHA)

The process starts with the identification of aircraft level functions, as shown in Figure 4.1.

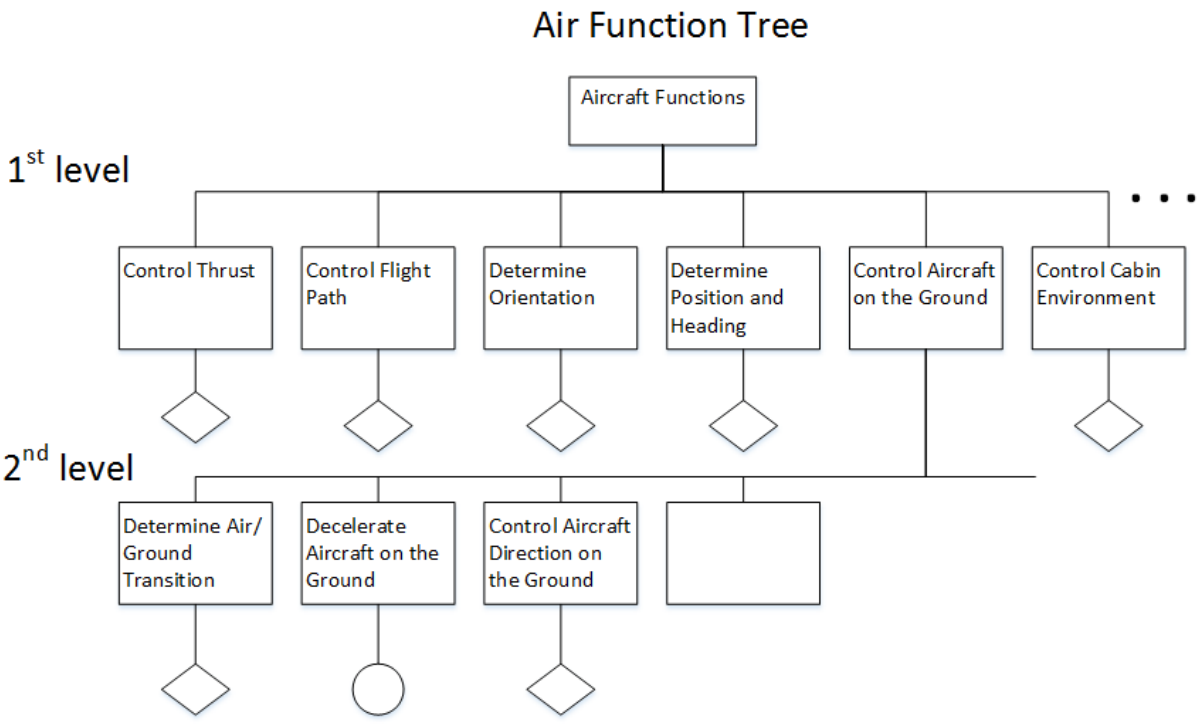

Figure 4.1: Aircraft Function Tree Aircraft High Level Functions and Associated Failure Conditions [SAE ARP 4761, p. 175]

Failure conditions for first-level functions are identified in Table 4.1.

TABLE 4.1: Example Failure Conditions [SAE ARP 4761, p. 31, Table A1]

Function	Failure Condition
Control Flight Path	Inability to control flight path
Control Touchdown and Roll Out	Inability to control Touchdown and Roll Out
Control Thrust	Inability to control Thrust
Control Cabin Environment	Inability to control Cabin Environment
Provide Spatial Orientation	Inability to provide Spatial Orientation
Fire Protection	Loss of Fire Protection

The example in ARP 4761 analyzes the function “Decelerate aircraft on the ground” (stopping on the runway) from a set of system functions that must be maintained throughout system operation [SAE ARP 4761, pp. 176-177]:

Functional Failure Conditions:

a. Loss of all deceleration capability
b. Reduced deceleration capability
c. Inadvertent activation
d. Loss of all auto stopping features
e. Asymmetrical deceleration

Environmental and Emergency Configurations and Conditions

a. Runway conditions (wet, icy, etc.)
b. Runway length
c. Tail/Cross wind
d. Engine out
e. Hydraulic System Loss
f. Electrical system loss

Applicable Phases:

a. Taxi
b. Takeoff to rotation
c. Landing Roll
d. Rejected takeoff (RTO)

Interfacing Functions:

a. Air/Ground determinations
b. Crew alerting (crew warnings, alerts, messages)

For each failure condition, the effects of the failure condition on the aircraft and crew are determined, as shown in Table 4.2 (only part of the original table is shown for space reasons).

TABLE 4.2: Aircraft FHA *(Partial Only)* [SAE ARP 4761, p. 178]

Function	**Failure Condition (Hazard Description)**	**Phase**	**Effect of Failure Condition on Aircraft/Crew**	**Classification**	**Reference to Supporting Material**	**Verification**
Decelerate Aircraft on the Ground	Loss of Deceleration Capability	Landing/ RTO/ Taxi	See Below			
	a. Unannunciated loss of deceleration capability	Landing/ RTO	Crew is unable to decelerate the aircraft resulting in a high speed overrun	Catastrophic		S18 Aircraft Fault Tree
	b. Annunciated loss of deceleration capability	Landing	Crew selects a more suitable airport, notifies emergency ground support and prepares occupants for landing overrun.	Hazardous	Emergency landing procedures in case of loss of stopping capability	S18 Aircraft Fault Tree
	c. Unannunciated loss of deceleration capability	Taxi	Crew is unable to stop the aircraft on the taxi way or gate resulting In low speed contact with terminal, aircraft, or vehicles	Major		
	d. Annunciated loss of deceleration capability	Taxi	Crew steers the aircraft clear of any obstacles and calls for a tug or portable stairs	No Safety Effect		
	Inadvertent Deceleration after Vl (Takeoff/RTO decision speed)	Takeoff	Crew is unable to takeoff due to application of brakes at the same time as high thrust settings resulting in a high speed overrun	Catastrophic		S18 Aircraft Fault Tree

Notice that in failure condition “d” (annunciated loss of deceleration capability), the crew is assumed to be able to steer the aircraft clear of any obstacles so the classification is “No safety effect.” In STPA, crew errors are integrated into the WBS hazard analysis.

Based on the FHA objectives, architectural decisions are made during the conceptual design phase. These decisions are the basis for the preliminary aircraft fault tree analysis shown in Figure 4.2. The aircraft fault tree in Figure 4.2 shows the results of the FHA in terms of probabilistic failure objectives based on the Table 4.2 classification of failure conditions.

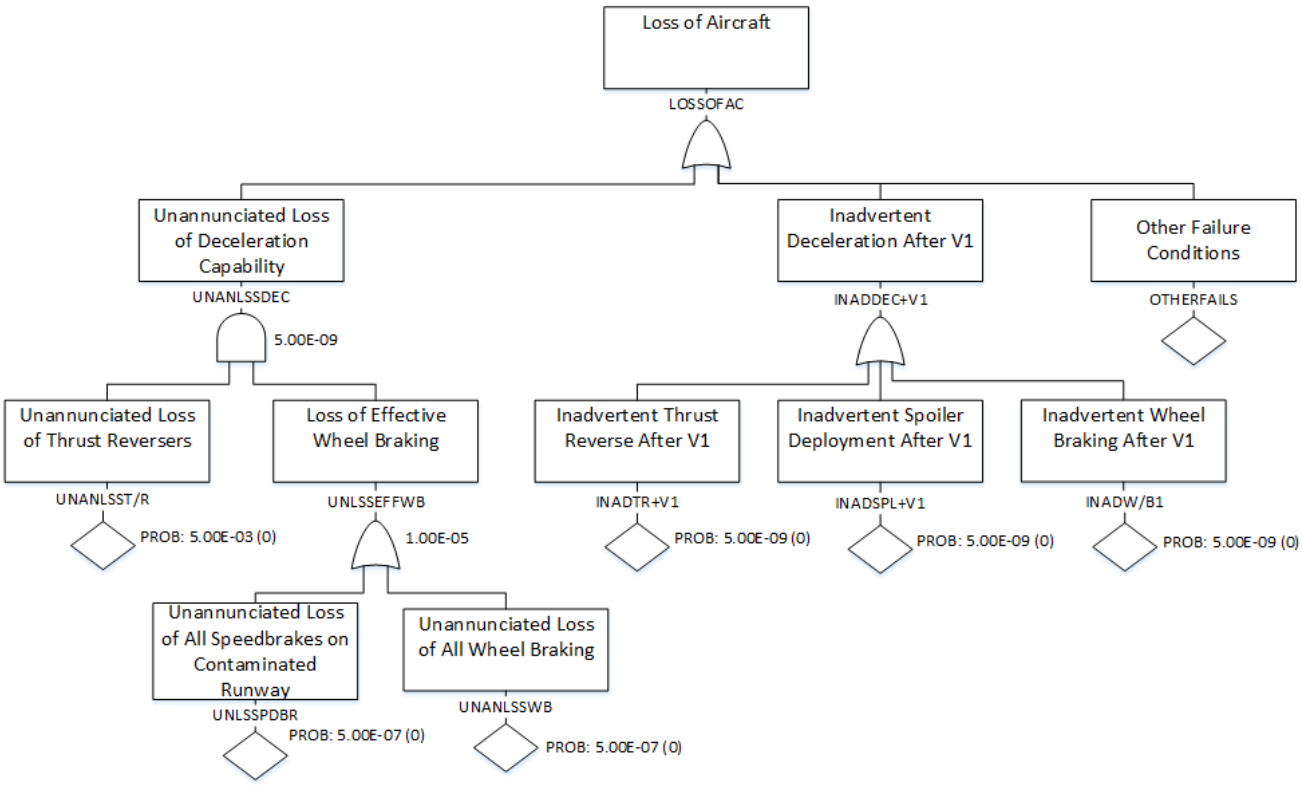

Figure 4.2: Aircraft FHA Preliminary Fault Tree [SAE ARP 4761, p. 182]

For the WBS example, in a similar way as at the aircraft level, the system level FHA begins by defining the functions needed. The general system description is: The primary purpose of the wheel braking system is to decelerate the aircraft on the ground without skidding the tires. The wheel braking system performs this function automatically upon landing or manually upon pilot activation. In addition to decelerating the aircraft, the wheel braking system is used for directional control on the ground through differential braking, stopping the main landing gear wheel rotation upon gear retraction, and preventing an aircraft motion when parked.

Not all of the aircraft-level functions are relevant to the WBS. Those that are relevant are listed below, further decomposed into sub-functions that the WBS must provide. Note that ARP 4761 Appendix L example is limited to the WBS. It does not consider the landing gear as a whole and therefore hazards relating to gear extension/retraction are not analyzed and we adopt the same limitation in our STPA analysis to maintain comparability.

The WBS functions are [from SAE ARP 4761, pp. 184-185]:

a. Decelerate the wheels on the ground
    (1) Manual activation
    (2) Automatic activation
    (3) Anti-skid
b. Decelerate the wheels on gear retraction
c. Differential braking for directional control
d. Prevent aircraft from moving when parked

Finally, the plan for the verification of safety objectives is created in a table similar to Table 4.2. The WBS FHA is then used to generate requirements, which are provided to the PSSA.

1) Loss of all wheel braking during landing or RTO shall be less than 5E-7 per flight
2) Asymmetrical loss of wheel braking coupled with loss of rudder or nose wheel steering during landing or RTO shall be less than 5E-7 per flight
3) Inadvertent wheel braking with all wheels locked during takeoff roll before V1 shall be less than 5E-7 per flight.
4) Inadvertent wheel braking of all wheels during takeoff roll after V1 shall be less than 5E-9 per flight.
5) Undetected inadvertent wheel braking on one wheel w/o locking during takeoff shall be less than 5E-9 per flight.

Note that all of these safety requirements are stated as probabilities.

4.2 Preliminary System Safety Analysis (PSSA)

The PSSA process has two main inputs: the aircraft and/or system FHA and the aircraft FTA. The system FHA yields failure conditions and classifications necessary for the next steps. The aircraft FTA determines the functional failures of concern and the budgeted failure rate. The aircraft FTA is supplemented by the Common Cause Analysis (CCA) to generate the top failure events for the system FTA. The CCA also establishes the system requirements such as redundancy, separation and independence of functions needed to be implemented by the design of the system.

The WBS design description gets more detailed at this point in the design process [SAE ARP 4761, p. 190-191] as specified in Section 2 and shown in Figure 2.2 and in [SAE ARP 4761, pp. 191-191].

The five safety requirements derived from the FHA are listed at the end of the previous section. An additional two requirement are generated from the CCA:

6) The wheel braking system and thrust reverser system shall be designed to preclude any common threats (tire burst, tire shred, flailing tread, structural deflection, etc.)
7) The wheel braking system and thrust reverser system shall be designed to preclude any common mode failures (hydraulic system, electrical system, maintenance, servicing, operations, design, manufacturing, etc.)

While these two added requirements are not probabilistic, they are generated solely to justify the probabilities for the first five requirements by requiring that the components represented by the boxes in the fault tree be independent

Table 4.3 shows the design decisions that result from the five safety requirements derived in the PSSA.

These design decisions lead to the decision to have a primary and backup system and thus a set of derived safety requirements associated with them (Table 4.4):

1. The primary and secondary system shall be designed to preclude any common threats (e.g., tire burst, tire shred, flailing tread, structural deflection).
2. The primary and secondary system shall be designed to preclude any common mode failures (hydraulic system, electrical system, maintenance, servicing, operations, design, manufacturing, etc.).

Table 4.3: **PSSA Wheel Brake System Safety Requirements and Design Decisions [ARP 761, p. 194)**

Safety Requirement	Design Decisions	Remarks
1. Loss of all wheel braking (unannunciated or annunciated) during landing or RTO shall be less than 5E- 7 per flight.	More than one hydraulic system required to achieve the objective (service experience). Dual channel BSCU and multimode brake operations.	The overall wheel brake system availability can reasonably satisfy this requirement. See PSSA FTA below.
2. Asymmetrical loss of wheel braking coupled with loss of rudder or nose wheel steering during landing shall be less than 5E-7 per flight.	Separate the rudder and nose wheel steering system from the wheel braking system. Balance hydraulic supply to each side of the wheel braking system.	The wheel braking system will be shown to be sufficiently independent from the rudder and nose wheel steering systems. System separation between these systems will be shown in the zonal safety analysis and particular risk analysis
3. Inadvertent wheel braking with all wheels locked during takeoff roll before V1 shall be less than 5E-7 per flight.	None	Requirement 4 is more stringent and hence drives the design.
4. Inadvertent wheel braking of all wheels during takeoff roll after V1 shall be less than 5E-9 per flight.	No single failure shall result in this condition.	None
5. Undetected inadvertent wheel braking on one wheel w/o locking during takeoff shall be less than 5E-9 per flight.	No single failure shall result in this condition.	None

Table 4.4: Design Decisions from Derived Safety Requirements (ARP 4761 p. 195)

Safety Requirement	Design Decisions	Remarks
1. The primary and secondary system shall be designed to preclude any common threats (tire burst, tire shred, flailing tread, structural deflection).	Install hydraulic supply to the brakes in front and behind the main gear leg.	Compliance will be shown by ZSA and PRA. *(Editor's Note: In this example only for the main gear bay zone and the tire burst particular risk.).*
2. The primary and secondary system shall be designed to preclude any common mode failures (hydraulic system, electrical system, maintenance, servicing, operations, design, manufacturing, etc.).	Choose two different hydraulic systems to supply the brakes, emergency braking without electrical power.	Compliance will be shown by CMA.

The PSSA provided in ARP 4761 includes analysis for only one failure condition: "Unannunciated loss of all wheel braking" [SAE ARP 4761, p. 196]. It is noted in ARP 4761 that the PSSA would "normally contain the fault trees for all significant failure conditions."

The PSSA fault tree is shown in Figure 4.3. Some design changes were required to satisfy the top level probabilistic failure requirement:

1) Two means of applying wheel brakes are used to effect a complete stop, i.e., the Normal and Alternate brake systems.
2) A parking brake is necessary for normal operation of the aircraft on the ground and a decision to allow it to act as an emergency brake is made.
3) Discussions with potential BSCU vendors revealed that a 6.6 E-6/hour failure rate is not feasible with a single item so two BSCUs are required.

The FTA in ARP 4761 Fig 4.2.1-2 shows that the top level functional failure requirement could not be met by a single and feasible BSCU so a second BSCU was added in the design resulting in the modified FTA shown in Figure 4.3 [SAE ARP 4761 Fig 4.2.1-3]. That analysis creates derived lower-level requirements, e.g., the installation requirement that the Primary and secondary hydraulic supply system shall be segregated. Note that in the example analysis, BSCU 1 and BSCU 2 failure are considered to be independent. A CCA would be done to ensure that the failures are independent, but indirect or other types of dependencies may not be detected (discussed further in Sections 5 and 6).

Item Level Requirements are generated from the fault trees and the design additions:

1) The probability of "BSCU Fault Causes Loss of Braking Commands" shall be less than 3.3E-5 per flight.
2) The probability of "Loss of a single BSCU shall be less than 5.75 per flight.
3) The probability of "Loss of Normal Brake System Hydraulic Components" shall be less than 3.3E-5 per flight.
4) The probability of "Inadvertent braking due to BSCU" shall be less than 2.5E-9 per flight.
5) No single failure of the BSCU shall lead to "inadvertent braking."

6) The BSCU shall be designed to Development Assurance Level A based on the catastrophic classification of "inadvertent braking due to BSCU."

Additional requirements on other systems are also generated such as: The probability of "Loss of Green Hydraulic Supply to the Normal brake system" shall be less than 3.3E-5 per flight. In addition, installation requirements and maintenance requirements are generated.

The same process can be repeated at a lower level of detail, for example, the BSCU. The resulting requirements from such a BSCU analysis given in SAE ARP 4761 [p. 227] are:

Installation Requirements:

1. Each BSCU System requires a source of power independent from the source supplied to the other system.

Hardware and Software Requirements:

1. Each BSCU system will have a target failure rate of less than 1E-4 per hour.
2. The targeted probabilities for the fault tree primary failure events have to be met or approval must be given by the system engineering group before proceeding with the design.
3. There must be no detectable BSCU failures that can cause inadvertent braking.
4. There must be no common mode failures of the command and monitor channels of a BSCU system that could cause them to provide the same incorrect braking command simultaneously.
5. The monitor channel of a BSCU system shall be designed to Development Assurance Level A.
6. The command channel of a BSCU system may be designed to Development Assurance Level B.[5]
7. Safety Maintenance Requirements: The switch that selects between system 1 and system 2 must be checked on an interval not to exceed 14,750 hours.

[5] The allocations in 5 and 6 could have been switched, designing the command channel to level A and the monitor channel to level B.

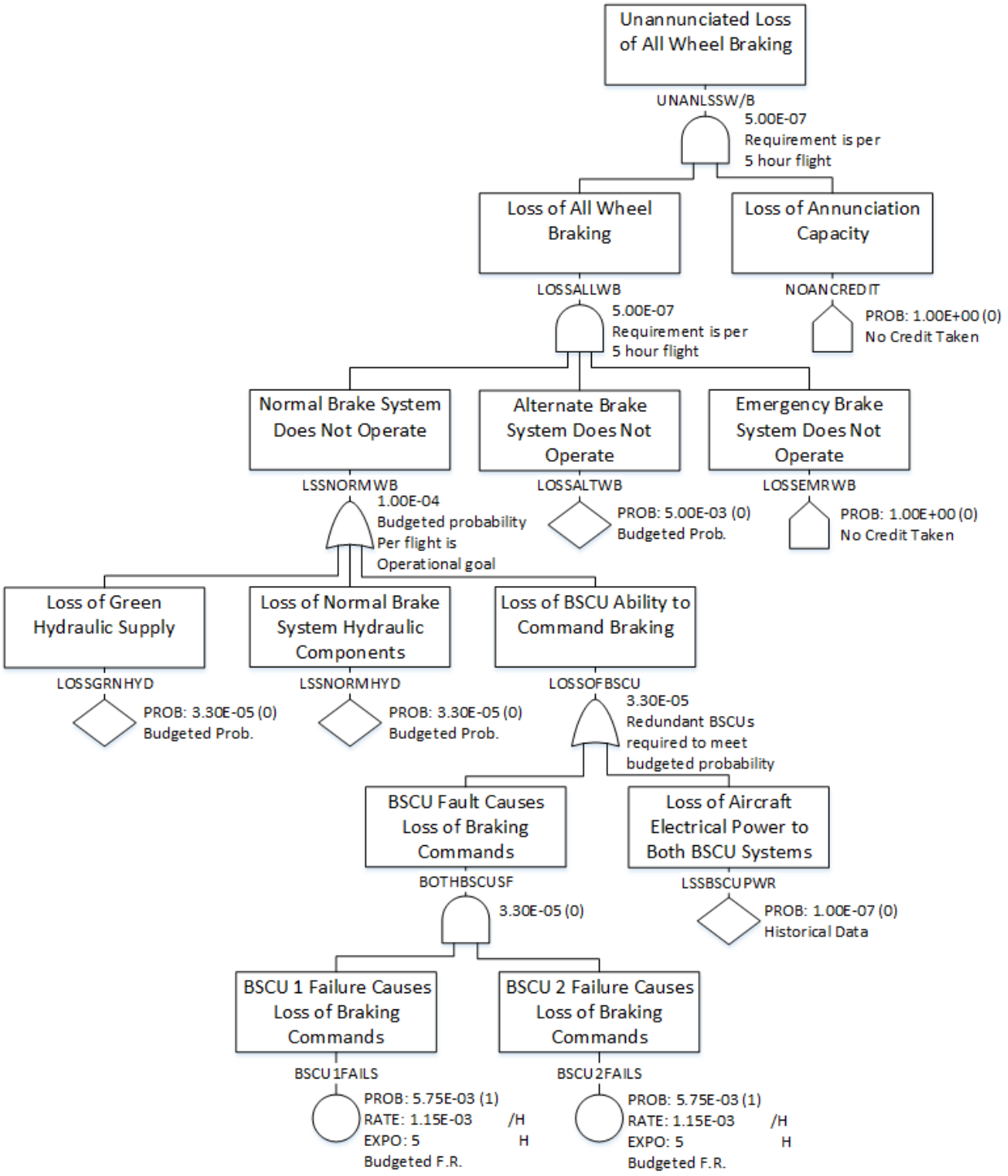

Figure 4.3: FTA - Unannunciated Loss of All Wheel Braking Fault Tree, Revision B [SAE ARP 4761, p. 200]

4.3 System Safety Analysis (SSA)

The SSA provides verification that the implemented design meets both the qualitative and quantitative safety requirements as defined in the FHA and PSSA. In the SSA, hardware reliability requirements, architectural requirements, and hardware and software DALs are verified against the safety requirements identified in the PSSA process. DO-178/DO-254 procedures are used to assure that the software/hardware implementation meets the required DAL. For example, in the WBS example in ARP 4761, there is a statement that "the computation and monitor channel software has been developed and rigorously verified to Development Assurance Level A and B respectively, using DO-178 compliant processes, thus precluding design flaws of concern" [SAE ARP 4761, page 251].

The SSA is usually based on the PSSA FTA (or Dependency Diagrams and Markov Analysis) and uses the quantitative failure values obtained from FMEA (Failure Modes and Effects Analysis). The results of FMEA are grouped together on the basis of their failure effects in a document called the FMES (Failure Modes and Effects Summary). We omit the details of this analysis and refer the interested reader to ARP 4761, Appendix L.

5. System-Theoretic Process Analysis (STPA) of the Wheel Brake System[6]

STPA is a hazard analysis technique that is based on system theory [Leveson, 2012; Leveson, 2013]. As will be seen, the process and results are very different than the approach specified in ARP 4761. STPA is a top-down, system engineering technique and can be used at the very beginning of the system's design to influence and guide design decisions. STPA is based on a model that assumes accidents are caused by inadequate enforcement of behavioral safety constraints on system component behavior and interactions. Rather than thinking of safety as a failure problem, it conceives of it as a control problem. Note that failures are still considered, but they are considered to be something that needs to be controlled, as are design errors, requirements flaws, component interactions, etc.

The underlying model of causality, called STAMP (System-Theoretical Accident Model and Processes), has been described elsewhere [Leveson, 2012] and that description is not repeated here.

The STPA process starts at the system level, as does FHA.[7] The rest of the STPA process can be decomposed into two main steps: (1) identifying unsafe control actions that can lead to system hazards and (2) identifying causal scenarios for the unsafe control actions. The scenarios include component failures but also additional factors such as direct and indirect interactions among system components (which may not have "failed"). The identified causal scenarios serve as the basis for developing system and component safety requirements and constraints.

---

[6] The authors of this paper have limited experience with wheel braking systems. There may be technical inaccuracies or missed hazards and causes that could be uncovered by STPA but are missed here because of our lack of knowledge. The results, however, should provide insight into the potential of STPA and allow a comparison with conventional methods.

[7] ARP 4761 labels the "aircraft level" what we would call the "system" level, where the system is the largest unit being considered. What the ARP labels the "system level" is, in more standard system engineering terminology, the components or subsystems.

5.1 System-Level Analysis

As in ARP 4761, STPA is an iterative process that starts at the aircraft level and continues to iterate until the hazards have been adequately analyzed and handled in the design.

The WBS has two modes of operation; manual and auto. ARP 4761 does not consider the hazards associated with the human factors (HF) of mode transitions, annunciation and potential mode confusion. STPA includes the human as part of the system that is analyzed and thus identifies these potential types of mode confusion as well as the hazards that could arise due to loss of synchronization between actual automation state and the crew's mental model of that state.

The goal of STPA is similar to that of other hazard analysis methods: it tries to determine how the system hazards could occur so the cause(s) can be eliminated or mitigated by modifying the system design. The goal, however, is not to derive probabilistic requirements, as in ARP 4761, but to identify hazardous scenarios that need to be eliminated or mitigated in the design or in operations. Hazards are defined as they are in System Safety engineering, that is, as system states or sets of conditions that, when combined with some set of environmental worst-case conditions, will lead to an accident or loss event [Leveson, 1995; Leveson, 2012]. Although ARP 4761 called the first analysis a "Functional Hazard Analysis," the term *hazard* seems to be defined very differently than in traditional system hazard analysis (as defined, for example, in MIL-STD-882)and is equated only with functional failures. This difference is discussed further in Section 6.

STPA uses the beginning products of a top-down system engineering approach, including the potential losses (accidents) and hazards leading to these losses. Unacceptable safety-related losses are:

A1. Loss of life or serious injury to aircraft passengers or people in the area of the aircraft
A2. Unacceptable damage to the aircraft or objects outside the aircraft

System hazards related to these losses include:

H1: Insufficient thrust to maintain controlled flight
H2: Loss of airframe integrity
H3: Controlled flight into terrain
H4: An aircraft on the ground comes too close to moving or stationary objects or inadvertently leaves the taxiway
H5: etc.

We are primarily concerned with system hazard **H4** in this report. The specific accidents related to **H4** occur when the aircraft operates on or near the ground and may involve the aircraft departing the runway or impacting object(s) on or near the runway. Such accidents may include hitting barriers, other aircraft, or other objects that lie on or beyond the end of the runway at a speed that causes unacceptable damage, injury or loss of life. **H4** can be refined into the following deceleration-related hazards:

Hazards:

H4-1: Inadequate aircraft deceleration upon landing, rejected takeoff, or taxiing
H4-2: Deceleration after the V1 point during takeoff
H4-3: Aircraft motion when the aircraft is parked
H4-4: Unintentional aircraft directional control (differential braking)
H4-5: Aircraft maneuvers out of safe regions (taxiways, runways, terminal gates, ramps, etc.)
H4-6: Main gear wheel rotation is not stopped when (continues after) the gear is retracted

The high-level system safety constraints (SCn) associated with these hazards are a simple restatement of the hazards in terms of requirements or constraints on the design.

SC1: Forward motion must be retarded within TBD seconds of a braking command upon landing, rejected takeoff, or taxiing.
SC2: The aircraft must not decelerate after V1.
SC3: Uncommanded movement must not occur when the aircraft is parked.
SC4: Differential braking must not lead to loss of or unintended aircraft directional control
SC5: Aircraft must not unintentionally maneuver out of safe regions (taxiways, runways, terminal gates and ramps, etc.)
SC6: Main gear rotation must stop when the gear is retracted

H4-4 and **H4-6**, although hazardous, are outside the scope of the ARP 4761 example analysis and therefore are also not considered in the STPA analysis that follows.

After identifying the accidents, the system safety hazards to be considered, and the system-level safety requirements (constraints), the next step in STPA is to create a model of the aircraft functional control structure. While a general control structure that includes the entire socio-technical system, including both development and operations, can be used, in this example we consider only the aircraft itself as that is the focus of the WBS example in ARP 4761. STPA analysis is performed using the functional control structure.

Figure 5.1 shows a very high-level model of the aircraft, with just three components: the pilot, the automated control system (which will probably consist of multiple computers), and the physical aircraft components. For complex systems, such as aircraft, levels of abstraction can be used to zoom in on the pieces of the control structure currently being considered. This type of top-down refinement is also helpful in understanding the overall operation of the aircraft and to identify interactions among the components.

The role of the pilot, as shown in the Figure 5.1 control structure, is to manage the automation and, depending on the design of the aircraft, directly or indirectly control takeoff, flight, landing, and maneuvering the aircraft on the ground. The pilot and the automated controllers contain a model of the system (for a human this is usually called the mental model) that they are controlling. The automation is controlling the aircraft so it must contain a model of the current aircraft state. The pilots also need a model of the aircraft state, but they also need a model of the state of the automation[8] and a model of the airport environment in which they are operating.

Pilots provide flight commands to the automation and receive feedback about the state of the automation and the aircraft. In some designs, the pilot can provide direct control actions to the aircraft hardware (i.e., not going through the automated system) and receive direct feedback.
The dotted lines represent this direct feedback. As the design is refined and more detailed design decisions are made, these dotted line links may be eliminated or instantiated with specific content. The pilot always has some direct sensory feedback about the state of the aircraft and the environment.

[8] Many pilot errors can be traced to flaws in their understanding of how the automation works or of the current state of the automation.

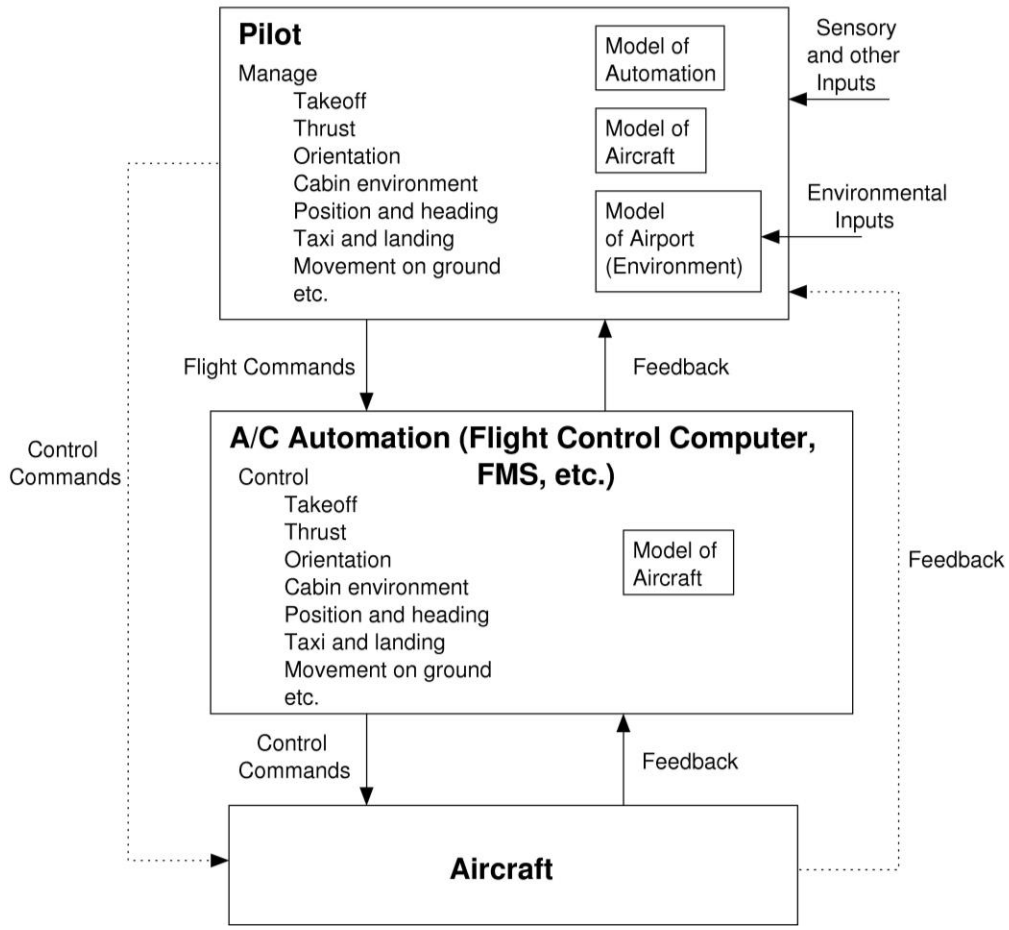

Figure 5.1: A High-Level Control Structure at the Aircraft Level

Figure 5.2 zooms in on the control model for the ground control function, which is the focus of the example in ARP 4761. There are three basic physical components being controlled, the reverse thrusters, the spoilers, and the wheel braking system. By including the larger functional control structure than simply the WBS, STPA can consider interactions (both intended and unintended) among the braking components related to the hazard being analyzed. Again, to promote comparison between the ARP 4761 safety assessment process results and STPA results, we focus only on the WBS in this paper.

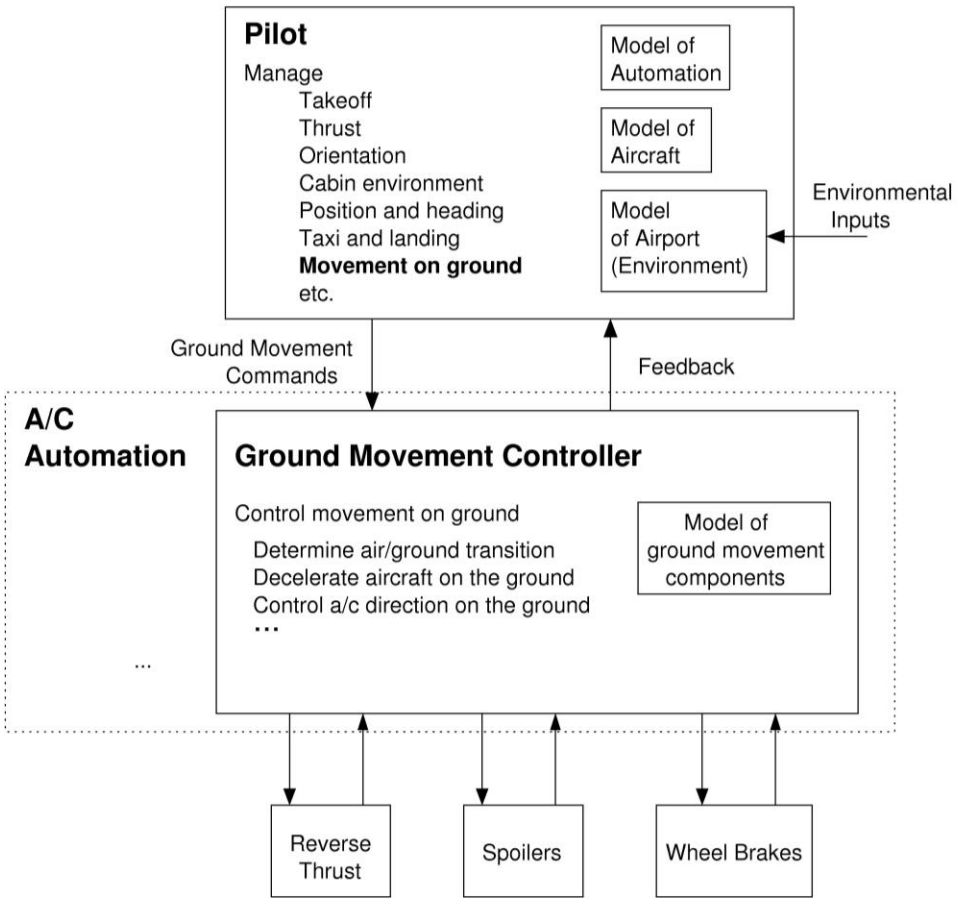

Figure 5.2: Control Structure for Ground Movement Control

Figure 5.3 shows only the parts of the control structure related to the WBS at a high level of abstraction. We had to make some assumptions because although ARP 4761 has technical detail down to every valve, it does not contain even basic information about pilot interaction with these systems. Specifically, ARP 4761 does not say anything about any visual feedback or any normal feedback to the crew, it only seems to specify annunciation for two abnormal conditions (i.e. BSCU fault or Alternate mode). We assumed very basic Autobrake feedback, which seems to be common practice.

The same is true for the power on/off command. ARP 4761 says pilots can force the WBS into alternate braking mode (i.e., mechanical instead of electronic braking), but it does not say how. The ARP 4761 BSCU design does not contain or allow a separate pilot command to make this switch. The BSCU design goes to alternate braking if power is lost or it detects a fault. We made the simplest assumption that pilots can turn off the BSCU power.

There is also mention in ARP 4761 about alerting the crew of faults, but no mention of what the crew can do after faults are annunciated. ARP 4761 design seems to only consider whether the fault is annunciated and assumes accidents will be prevented in that case (e.g., see [SAE ARP 4761, p. 178]). We include the crew in our hazard analysis, however. In practice, when a fault is annunciated, the standard way to reset the fault seems to be for the crew to turn the power off and on, which is what we have assumed. In fact, this practice has contributed to at least one accident.

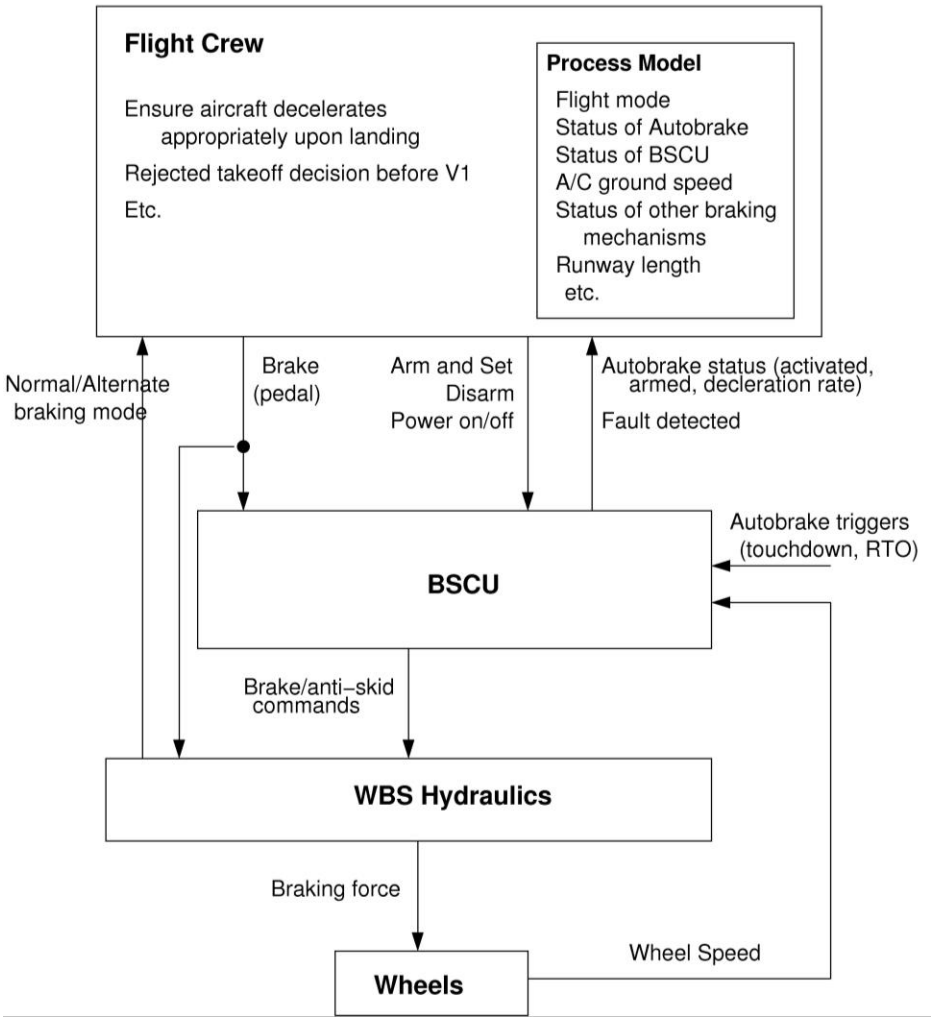

Figure 5.3: The Control Structure for the Wheel Braking System

Figure 5.4 shows a more detailed model of the functional control structure for the WBS. This model of the functional structure differs from the model of the physical structure of the WBS found in ARP 4761. The intent of the model in Figure 5.4 is to show the "functional" structure without any assumptions about the implementation. We needed to add functional details not in the ARP 4761 figure (such as Autobrake commands and status) in order to more fully specify the system function and, therefore, more fully analyze the hazard scenarios. We omitted physical details and functional implementation details that were added in ARP 4761 as a result of the safety assessment process. The biggest difference is the omission of a BSCU redundant channel in Figure 3.0-1 [p. 211] of ARP 4761, which was added to achieve the ARP 461 FTA reliability (failure probability) goals. This addition is not necessary for our purposes and is not shown in the functional control structure. STPA starts without a specific design solution to potential problems. Instead, it starts from the basic required functional behavior and identifies the ways that that behavior can be hazardous. Designers can later decide on particular design solutions, such as redundancy, if that turns out to be necessary to satisfy the safety requirements derived through this analysis.

Perhaps the easiest way to explain this difference is that the main goal of the safety assessment process in ARP 4761 is to show that the system is safe, defined as achieving a failure probability goal of less than 10E-9. Because of the nature of the ARP 4761 process, which is to identify probability goals for component failure, this approach is reasonable. The approach does, however, bias the design solutions toward redundancy and the addition of monitors, which can be easily added to achieve the fault tree probability goals. The use of a probabilistic risk analysis methodology tends to drive design solutions towards redundancy rather than considering higher level solutions that may obviate that necessity.

The goal of STPA, in contrast, is to identify hazardous behaviors so they can be eliminated or controlled in the system design, which results in identifying behavioral (functional but not necessarily probabilistic) safety requirements for the various system components, including the software and human operators.

For example, one problem we identified is that the BSCU receives brake pedal commands from both pilots, but the pilots never receive any feedback about what the other pilot is doing. This feedback is important not only for manual braking (pilots may both assume the other is controlling the pedal), but also because if either pilot touches the pedal when Autobrake is active, it will automatically disarm the Autobrake system. Because this behavior is not considered a fault, ARP 4761 does not require that it is annunciated or require alerting the crew that Autobrake has been deactivated.

Although ARP 4761 includes a proposed WBS design, several assumptions were not explicitly documented in the WBS description making it difficult to understand its operation. For example, the preliminary wheel brake system diagram on page 192 refers to “Pedal Pos. 1” and “Pedal Pos. 2” but it is not clear what these refer to or why there are two. They could refer to separate inputs from two pilots, a redundancy input from one pilot, differential (directional) inputs from one pilot, etc. As another example, the system diagram and text includes an automatic selector valve but does not document what “automatic” means in this context, the assumptions about what will control the behavior of the valve, which four ports can be connected or blocked simultaneously, etc. Although not documented, the ARP 4761 analysis seems to assume several answers to these and other questions.

There are also parts of the ARP 4761 WBS description that seemed to be in conflict with each other. For example the proposed architecture shows that the green shutoff valve can only be closed if the BSCU internal monitors detect a fault, however the text description on page 191 states that the system must also allow the pilots to manually command the switch-over. We have documented our understanding of the WBS system in ARP 4761, including these and other assumptions, in the Appendix to this report. If STPA were being done before the physical system design, then the physical design would be done after the STPA hazard analysis and STPA would create the behavioral safety requirements that must be implemented in the component design in order to control the hazards. While this approach is desirable, it makes the comparison with the example in ARP 4761 more difficult and therefore we start from the ARP 4761 design augmented with more details to allow a general hazard analysis and not just a component failure analysis. We do show how to generate the requirements for the Autobrake because no architecture is included for that component in ARP 4761. Note also that we have included feedback that appears to be necessary in the control diagram. The STPA analysis identifies what feedback is necessary to enforce the safety constraints, including additional feedback that is not already in the candidate control structure in Figure 5.4.

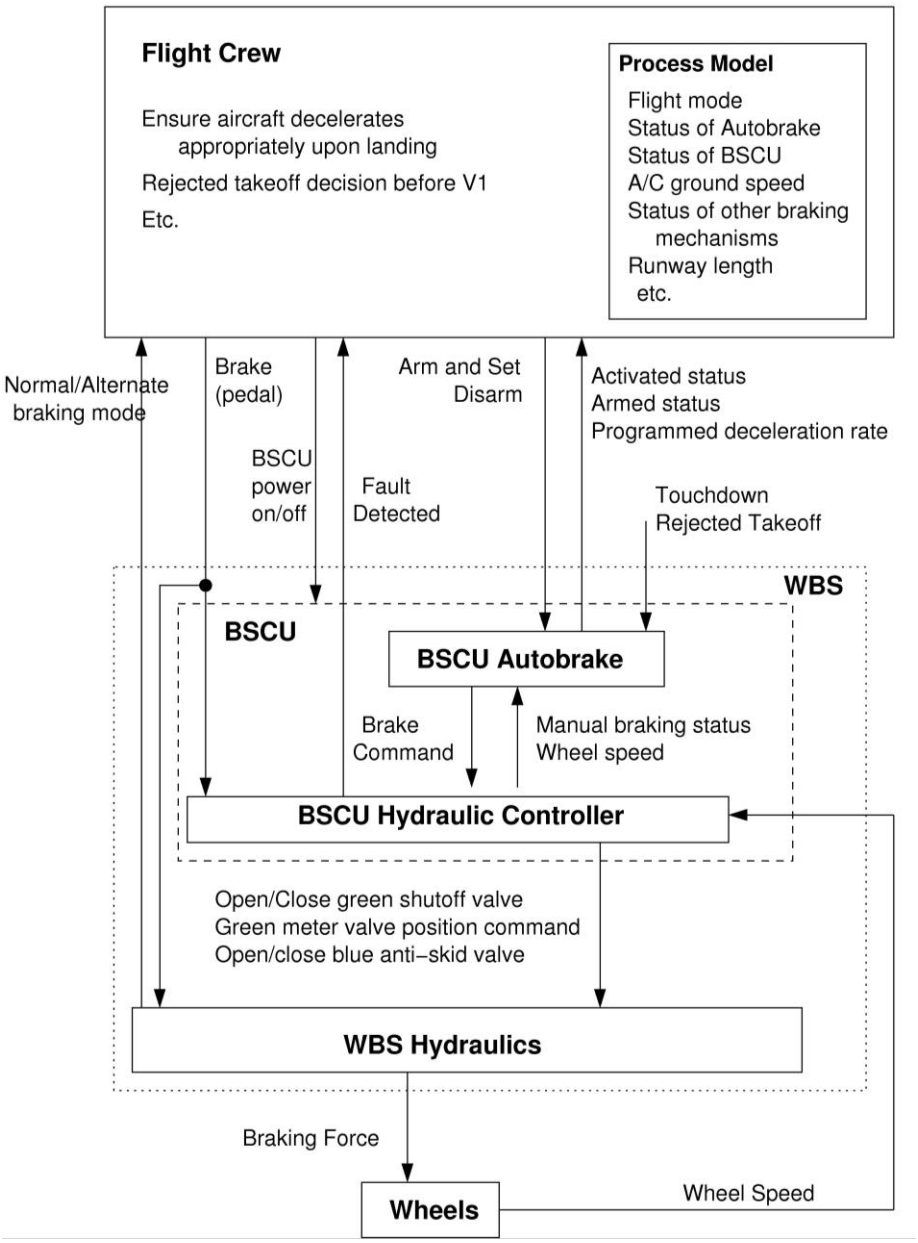

Figure 5.4: Functional Control Structure for WBS

The aircraft physical system, which includes the wheels and other physical components, is controlled by the WBS hydraulics. The WBS hydraulics include the hydraulic lines, the accumulator, the valves, and the pistons that apply braking force to the wheels. The WBS hydraulics will also detect when the system has switched to alternate braking mode and can alert the flight crew.

The WBS hydraulics are controlled either manually by the flight crew or electrically by the BSCU. The BSCU is comprised of two controllers. The hydraulic controller, which is analyzed in ARP 4761 based on the detailed architecture shown in Figure A1 (in Appendix A), outputs various hydraulic commands to achieve the desired braking force commanded by either the pilots or the Autobrake controller. The

hydraulic controller also performs anti-skid functionality based on aircraft signals such as wheel speed information.

The Autobrake controller, which is described functionally in ARP 4761 but has no proposed architecture, is configured by the pilots and automatically commands the necessary braking when triggered. The Autobrake controller receives external triggers from the aircraft that indicate when a touchdown or rejected takeoff has occurred. The Autobrake controller also receives an indication of the manual braking status and will automatically deactivate in the event that it is active while the pilots perform manual braking.

The flight crew can configure the Autobrake controller or provide manual braking commands using brake pedals. The manual braking commands are mechanically connected to the WBS hydraulics for alternate mode braking. The same manual braking commands are electrically sensed by the BSCU logic for normal mode braking.

The flight crew commands for Autobraking include:

- **Arm & set deceleration rate**: This command puts the Autobrake controller in "armed" mode. When armed, the Autobrake waits for a trigger such as the input indicating the aircraft just landed. When a trigger is received, Autobrake automatically issues brake commands to achieve the configured deceleration rate. When Autobrake is automatically applying the brakes, it is considered activated. The arm and set command is provided with a single action although the exact implementation may vary. For example, the crew may push a button corresponding to Low, Medium, or High deceleration that simultaneously arms the system. If this command is provided when Autobrake is already armed, the controller will remain armed but with the new deceleration rate. If this command is provided when Autobrake is already applying the brakes, the controller will immediately use the new deceleration rate.
- **Disarm:** When the Autobrake controller is armed, this command will put the Autobrake controller in "not armed" mode. In this mode, the controller will not automatically apply the brakes when a trigger is received. If the brakes are already being automatically applied, the Autobrake controller will immediately stop sending brake commands. The disarm command is provided with a single action, for example by using a button.

When Autobrake is triggered, it will continue to send braking commands until the aircraft comes to a stop, the crew disarms Autobrake, or the Autobrake controller detects manual braking commands from the crew. In all of these cases, Autobrake will become disarmed.

The crew is notified when Autobrake is active (i.e., sending automated brake commands), armed (i.e., configured and waiting for a trigger), and what the currently configured deceleration rate is (i.e., if Autobrake is armed or activated).

In addition to Autobrake commands, the crew can power off the overall BSCU to force the WBS hydraulics into alternate mode or to reset BSCU internal fault monitors. In addition to Autobrake feedback, the crew is notified when the WBS hydraulics are in normal or alternate braking mode and whether the BSCU has flagged an internal fault.

We now have enough basic information to perform the STPA hazard analysis. The analysis itself, for convenience, can be separated into two steps, although this division is not strictly necessary.

5.2 Identifying Potentially Unsafe Control Actions (Step 1)

The first step in STPA identifies potential hazardous control actions. At this stage in the analysis, it is immaterial whether control actions are provided manually or automatically. Our purpose is to define the

hazardous control actions from any source. We have developed tools to assist in the Step 1 analysis [Thomas, 2013], but they are beyond the scope of this report. The results of Step 1 are used to guide the generation of scenarios in Step 2 and can also be used to create requirements and safety constraints on the system design and implementation. For example, a safety constraint on the pilot might be that manual braking commands must be provided to override Autobrake in the event of insufficient Autobraking. Such constraints on humans clearly are not enforceable in the same way as constraints on physical components, but they can be reflected in the design of required pilot operational procedures, in training, and in performance audits. Some requirements that are considered to be error-prone or unachievable by human factors experts might result in changes in the braking system design.

We have found it convenient to document these unsafe control actions in a tabular form. Table 5.1 shows the control actions that can be provided by the crew, Table 5.2 shows those that can be given by the BSCU Autobrake controller, and Table 5.3 shows those that can be given by the BSCU Hydraulic Controller. The entries in the tables include both the control action (found in the control structures) and the conditions under which it will be hazardous The first column lists control actions that can be given by the controller and the four following columns list how those control actions could be hazardous in four general categories. These hazardous control actions are referred to as unsafe control actions (UCA). The four categories are;

- *Not providing causes hazard*: if the control action is not provided under certain conditions, then a hazard will ensue
- *Providing causes h*azard: if the control action is given under certain conditions, then a hazard will ensue
- *Too soon, too late, out of sequence causes hazard*: if the timing of the control action is critical (before/after/within a time window) relative to another control action, then a hazard will ensue
- *Stopped too soon, applied too long causes hazar*d: applicable only to continuous control actions and not discrete ones

Unsafe control may depend on the operational phase, so the applicable phase is noted in the table. For example, not providing braking input in cruise is not hazardous whereas it is in the landing phase. We labeled the UCAs with a reference code (e.g. CREW.1a1), some of which we will use as examples in the causal analysis step (Step 2). Where we did not know enough about braking system design to write specific requirements, we used “TBD” to indicate the need for more information by aircraft designers.

Step 1 analysis only identifies unsafe control actions. Hazards that result when a safe control action is provided but not followed or executed—which is the major focus of the ARP 4761 process—are identified in Step 2.

Table 5.1: Unsafe Control Actions for Flight Crew

Control Action By Flight Crew:	Not providing causes hazard	Providing causes hazard	Too soon, too late, out of sequence	Stopped too soon, applied too long
CREW.1 Manual braking via brake pedals	CREW.1a1 Crew does not provide manual braking during landing, RTO, or taxiing when Autobrake is not providing braking (or insufficient braking), leading to overshoot [H4-1, H4-5]	CREW.1b1 Manual braking provided with insufficient pedal pressure, resulting inadequate deceleration during landing [H4-1, H4-5	CREW.1c1 Manual braking applied before touchdown causes wheel lockup, loss of control, tire burst [H4-1, H4-5]	CREW.1d1 Manual braking command is stopped before safe taxi speed (TBD) is reached, resulting in overspeed or overshoot [H4-1, H4-5]
		CREW.1b2 Manual braking provided with excessive pedal pressure, resulting in loss of control, passenger/crew injury, brake overheating, brake fade or tire burst during landing [H4-1, H4-5	CREW.1.c2 Delayed manual braking applied too late (TBD) to avoid collision or conflict with another object and overloads braking capability given aircraft weight, speed, distance to object (conflict), and tarmac conditions [H4-1, H4-5]	CREW.1d2 Manual braking applied too long, resulting in stopped aircraft on runway or active taxiway [H4-1]
		CREW.1b3 Manual braking provided during normal takeoff [H4-2, H4-5]		
CREW.2 Arm autobrake	CREW.2a1 Autobrake not armed before landing causes loss of automatic brake operation when spoilers deploy. Crew reaction time may lead to overshoot. [H4-1, H4-5]	CREW.2b1 Autobrake not armed to maximum level during takeoff. This assumes that maximum braking force is necessary for rejected takeoff [H4-2]	CREW.2c1 Arm command provided too late (TBD), resulting in insufficient time for BSCU to apply brakes. [H4-1, H4-5]	

	Crew.2a2 Autobrake not armed prior to takeoff, resulting in insufficient braking during rejected takeoff (assumes that Autobrake is responsible for braking during RTO after crew throttle down) [H4-2]	CREW.2b2 Armed with too high of a deceleration rate for runway conditions, resulting in loss of control and passenger or crew injury. [H4-1, H4-5]		
		CREW.2b3 Autobrake is activated during takeoff [H4-1]		
CREW.3 Disarm Autobrake	CREW.3a1 Disarm Autobrake not provided during TOGA, resulting in loss of acceleration during (re)takeoff. [H4-1, H4-2, H4-5]	CREW.3b1 Autobrake disarm during landing or RTO causes loss of automatic brake operation when spoilers deploy. Crew reaction time may lead to overshoot. [H4-1, H4-5]	CREW.3c1 Disarm Autobrake provided more than TBD seconds after (a) aircraft descent exceeds TBD fps, (b) visibility is less than TBD ft, (c) etc…, resulting in either loss of control of aircraft or loss of acceleration during (re)takeoff [H4-1, H4-2, H4-5]	
CREW.4 Power off BSCU	CREW.4a1 Crew does not power off BSCU to enable alternate braking mode in the event of abnormal WBS behavior [H4-1, H4-2, H4-5]	CREW.4b1 Crew inadvertently powers off BSCU while Autobraking is being performed [H4-1, H4-5] CREW.4b2 Crew powers off BSCU when Autobrake is needed and is about to be used [H4-1, H4-5] CREW.4b3 Crew powers off BSCU when Anti-Skid functionality is needed (or will be needed) and WBS is functioning normally [H4-1, H4-5]	CREW.4c1 Crew powers off BSCU too late (TBD) to enable alternate braking mode in the event of abnormal WBS behavior [H4-1, H4-5] CREW.4c2 Crew powers off BSCU too early before Autobrake or Anti-Skid behavior is completed when it is needed [H4-1, H4-5]	N/A

CREW.5 Power on BSCU	CREW.5a1 Crew does not power on BSCU when Normal braking mode, Autobrake, or Anti-Skid is to be used [H4-1, H4-5]		CREW.5c1 Crew powers on BSCU too late after Normal braking mode, Autobrake, or Anti-Skid is needed [H4-1, H4-5]	N/A

Table 5.2: Unsafe Control Actions (BSCU Autobrake Controller)

Control Action BSCU:	Not providing causes hazard	Providing causes hazard	Too soon, too late, out of sequence	Stopped too soon, applied too long
BSCU.1 Brake command	BSCU.1a1 Brake command not provided during RTO (to V1), resulting in inability to stop within available runway length [H4-1, H4-5]	BSCU.1b1 Braking commanded excessively during landing roll, resulting in rapid deceleration, loss of control, occupant injury [H4-1, H4-5]	BSCU.1c1 Braking commanded before touchdown, resulting in tire burst, loss of control, injury, other damage [H4-1, H4-5]	BSCU.1d1 Brake command stops during landing roll before TBD taxi speed attained, causing reduced deceleration [H4-1, H4-5]
	BSCU.1a2 Brake command not provided during landing roll, resulting in insufficient deceleration and potential overshoot [H4-1, H4-5]	BSCU.1b2 Braking command provided inappropriately during takeoff, resulting in inadequate acceleration [H4-1, H4-2, H4-5]	BSCU.1c2 Brake command applied more than TBD seconds after touchdown, resulting in insufficient deceleration and potential loss of control, overshoot [H4-1, H4-5]	BSCU.1d2 Brake command applied too long (more than TBD seconds) during landing roll, causing stop on runway [H4-1]
	BSCU.1a3 Brake command not provided during taxi, resulting in excessive speed, inability to stop, or inability to control speed [H4-1, H4-5]	BSCU1b3 Brake command applied with insufficient level, resulting in insufficient deceleration during landing roll [H4-1, H4-5]	BSCU1c3 Brake command applied at any time before wheels have left ground and RTO has not been requested (brake might be applied to stop wheels before gear retraction) [H4-1, H4-2, H4-5]	BSCU.1d3 Brake command applied for tire lock until less than TBD seconds before touchdown (during approach), resulting in loss of control, equipment damage [H4-1, H4-5]
	BCSU.1a4 Brake command not provided after takeoff to lock wheels, resulting in potential equipment damage during landing gear retraction or wheel rotation in flight [H4-6]		BSCU.1c4 Brake command applied more than TBD seconds after V1 during rejected takeoff (assumes that Autobrake is responsible for braking during RTO after crew throttle down) [H4-2]	

Table 5.3: Unsafe Control Actions (BSCU Hydraulic Controller)

Control Action Hydraulic Controller:	Not providing causes hazard	Providing causes hazard	Too soon, too late, out of sequence	Stopped too soon, applied too long
HC.1 Open green shut-off valve (i.e. allow normal braking mode)	HC.1a1 HC does not open the valve to enable normal braking mode when there is no fault requiring alternate braking and Autobrake is used [H4-1, H4-5]	HC.1b1 HC opens the valve to disable alternate braking mode when there is a fault requiring alternate braking [H4-1, H4-2, H4-5] HC.1b2 HC opens the valve to disable alternate braking when crew has disabled the BSCU [H4-1, H4-2, H4-5]	HC.1c1 HC opens the valve too late (TBD) after normal braking is possible and needed (e.g. for Autobrake functionality) [H4-1, H4-2, H4-5] HC.1c2 HC opens the valve too late (TBD) after the crew has enabled the BSCU [H4-1, H4-2, H4-5]	HC.1d1 HC holds the valve open too long (TBD time) preventing alternate braking when normal braking is not operating properly [H4-1, H4-2, H4-5] HC.1d2 HC stops holding the valve open too soon (TBD) preventing normal braking when it is possible and needed (e.g. for Autobrake functionality) [H4-1, H4-2, H4-5]
HC.2 Pulse green meter valve and blue anti-skid valve	HC.2a1 HC does not pulse valves in the event of a skid [H4-1, H4-5]	HC.2b1 HC pulses valves when wheels are not skidding [H4-1, H4-2, H4-5] HC.2b2 HC incorrectly pulses valves with a frequency too high (TBD) or too low (TBD) to be effective [H4-1, H4-5] HC.2b3 HC pulses valves with a duty cycle too high (TBD) or too low (including keeping valve continuously open or closed) [H4-1, H4-5]	HC.2c1 HC pulses the valves too late after a skid has started [H4-1, H4-5] HC begins pulses more than TBD seconds after skid has started, resulting in loss of control [H4-1, H4-5]	HC.2d1 HC stops pulsing valves at any time before wheels regain traction, resulting in loss of control [H4-1, H4-5] HC.2d2 HC pulses the valves for more than TBD seconds after the wheels have stopped skidding, resulting in unnecessary loss of braking force [H4-1, H4-2, H4-5]

		HC.2b4 HC pulses the valves when aircraft speed is below 2 meters per second preventing a complete stop [H4-1, H4-5] HC.2b5 HC actuates blue anti-skid valve in any way when wheels are not skidding [H4-1, H4-5]		
HC.3 Green meter valve position command	HC.3a1 HC does not provide a position command to the valve when brake commands are received [H4-1, H4-2, H4-5]	HC.3b1 HC provides a position command that opens the valve when no brake commands are received. [H4-1, H4-2, H4-5] HC.3b2 HC provides a position command that closes the valve when brake commands are received [H4-1, H4-5] HC.3b3 HC provides a position command that is too low or too high (TBD) to achieve the commanded braking [H4-1, H4-2, H4-5]	HC.3c1 HC provides a position command too late (TBD) after braking is commanded by the crew or the Autobrake controller [H4-1, H4-2, H4-5]	HC.3d1 HC stops providing a position command (to keep valve open) too soon (TBD) while braking is still being commanded [H4-1, H4-5] HC.3d2 HC provides a position command (to keep valve open) too long (TBD) after braking was commanded [H4-1, H4-2, H4-5]

The results of this process can be used to produce general safety requirements for subsystems, training, etc. They will be refined into more detailed requirements in Step 2 when the causes of the unsafe control actions are identified. Some example requirements for the flight crew derived from the unsafe control actions are:

FC-R1: Crew must not provide manual braking before touchdown [CREW.1c1]

Rationale: Could cause wheel lockup, loss of control, or tire burst.

FC-R2: Crew must not stop manual braking more than TBD seconds before safe taxi speed reached [CREW.1d1]

Rationale: Could result in overspeed or runway overshoot.

FC-R3: The crew must not power off the BSCU during autobraking [CREW.4b1]

Rationale: Autobraking will be disarmed.

etc.

Example requirements that can be generated for the BSCU:

BSCU-R1: A brake command must always be provided during RTO [BSCU.1a1]

Rationale: Could result in not stopping within the available runway length

BSCU-R2: Braking must never be commanded before touchdown [BSCU.1c1]

Rationale: Could result in tire burst, loss of control, injury, or other damage

BSCU-R3: Wheels must be locked after takeoff and before landing gear retraction [BSCU.1a4]

Rationale: Could result in reduced handling margins from wheel rotation in flight.

Finally, some examples of requirements for the BSCU hydraulic controller commands to the three individual valves:

HC-R1: The HC must not open the green hydraulics shutoff valve when there is a fault requiring alternate braking [HC.1b1]

Rationale: Both normal and alternate braking would be disabled.

HC-R2: The HC must pulse the anti-skid valve in the event of a skid [HC.2a1]

Rationale: Anti-skid capability is needed to avoid skidding and to achieve full stop in wet or icy conditions.

HC-R3: The HC must not provide a position command that opens the green meter valve when no brake command has been received [HC.3b1]

Rationale: Crew would be unaware that uncommanded braking was being applied.

5.3 Identifying the Causes of Unsafe Control Actions (Step 2)

Step 2 involves identifying causes for the instances of unsafe (hazardous) control identified in Step 1. It also identifies the causes for a hazard where safe control was provided but that control was improperly executed or not executed by the controlled process. Figure 5.5 shows some of the factors that should be considered in this process. Notice that the unsafe control actions (upper left hand arrow from the controller to the actuator) have already been identified in Step 1.

This process differs from a FMEA in that not all failures are considered, but only causes of the identified unsafe control actions. It is similar to the scenarios leading to a hazard that are identified in fault tree analysis, but more than just component failure is identified and indirect relationships are considered. The use of a model (the functional control structure) on which the analysis is performed and a defined process that the analyst follows are less likely to lead to missing scenarios and allows the analysis to be revised quickly following design modifications arising from the hazard analysis.

The following sections demonstrate how STPA Step 2 can be applied to identify scenarios related to human, software, and hardware controllers. Notice that these scenarios can involve unsafe control actions and process model flaws across multiple controllers. Complete scenarios might not necessarily

be limited to any single controller. In fact, a UCA by one controller might indirectly cause UCAs by another controller.

One thing to note is that several scenarios shown here involve multiple controllers. Scenarios might not necessarily be limited to any single controller. For length reasons, the analysis for only one unsafe control action is included here for the flight crew, the Autobrake controller, and the hydraulic controller.

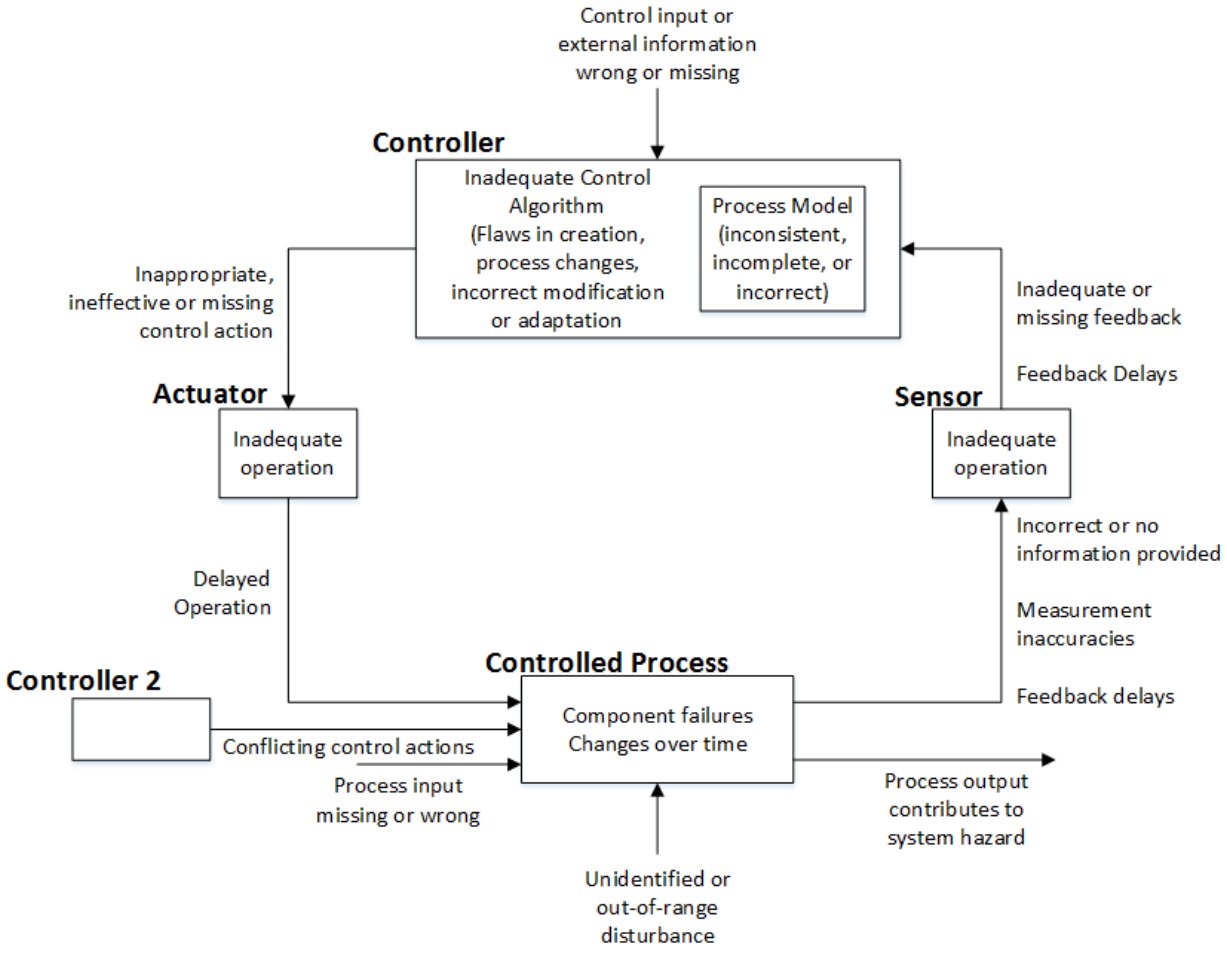

Figure 5.5: Generic Control Loop Flaws

Flight Crew

This section analyzes CREW.1a1, which is a flight crew control action that can be hazardous if not provided:

> **CREW.1a1**: Crew does not provide manual braking when there is no Autobraking and braking is necessary to prevent **H4-1** and **H4-5**.

The causes of this unsafe control action are considered by analyzing the parts of the control loop highlighted in Figure 5.6.

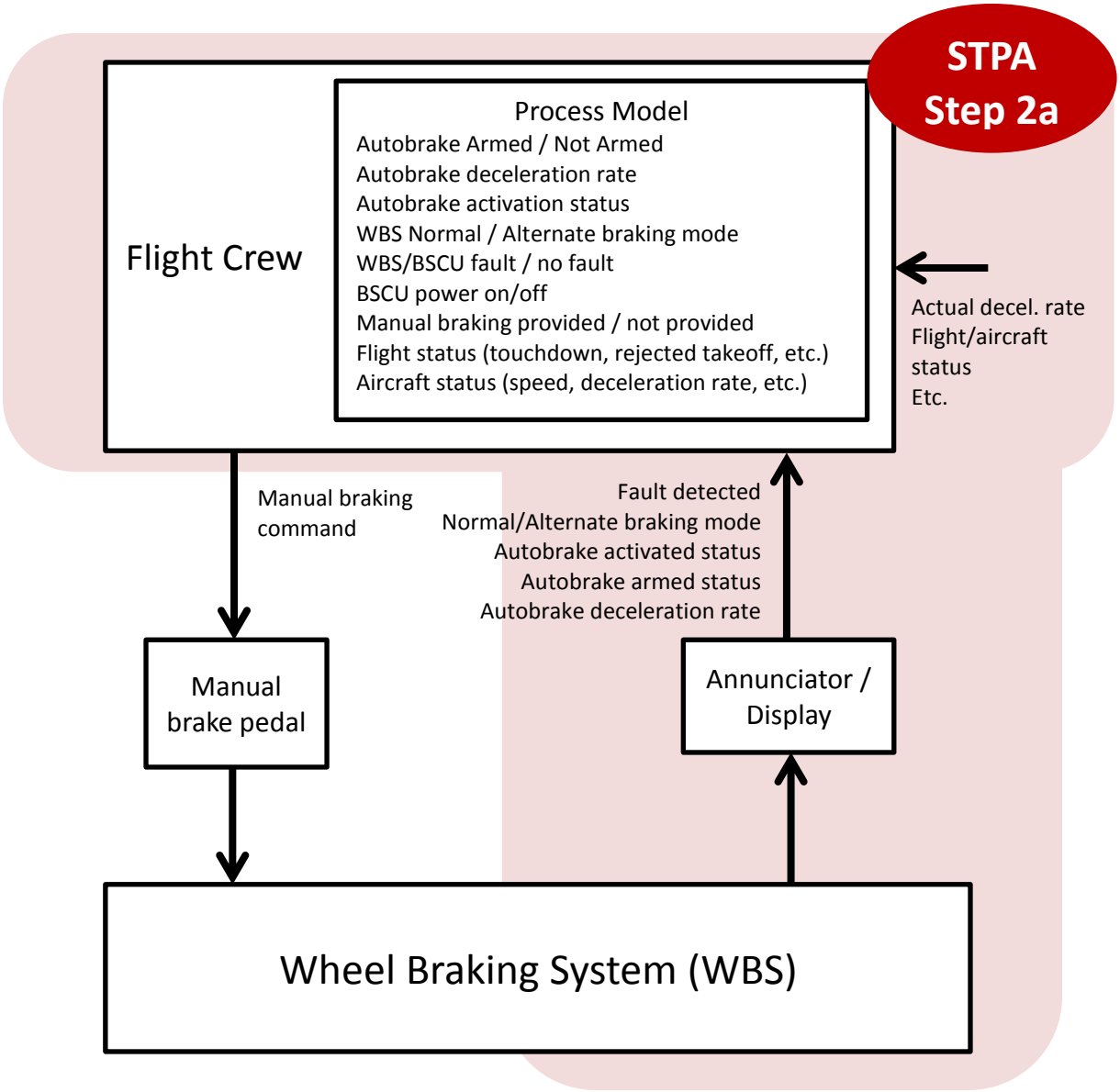

Figure 5.6: Control structure analyzed in STPA Step 2a to identify causes of unsafe control

Beginning with the unsafe control action and working backwards, scenarios can be identified by developing explanations for each causal factor in succession, for example:

UNSAFE CONTROL ACTION – CREW.1a1: Crew does not provide manual braking when there is no Autobraking and braking is necessary to prevent **H4-1** and **H4-5**.

Scenario 1: Crew incorrectly believes that the Autobrake is armed and expect the Autobrake to engage (process model flaw). Reasons that their process model could be flawed include:

a) The crew previously armed Autobrake and did not know it subsequently became unavailable, AND/OR

b) The feedback received may not have been adequate if the BSCU Hydraulic Controller detected a fault. The crew would be notified of a generic BSCU fault but they would also be notified that Autobrake is still armed (even though Autobraking is no longer available), AND/OR

c) The crew would be notified that the Autobrake controller is still armed and ready, because the Autobrake controller is not designed detect when the BSCU has detected a fault. When the BSCU detects a fault it closes the green shut-off valve (making Autobrake commands ineffective), but the Autobrake system itself will not notify the crew.

d) The crew could not process feedback due to multiple messages, conflicting messages, alarm fatigue, etc.

Possible new requirements for S1: The BSCU hydraulic controller must provide feedback to the Autobrake when it is faulted and the Autobrake must disengage (and provide feedback to crew)

Scenario 2: Crew does not provide manual braking upon landing because each pilot believed the other pilot was providing manual braking commands (process model incorrect). Reasons the process model could be incorrect include:

a) The crew feel initial deceleration from other systems (spoilers, reverse thrust, etc.) and may have different understandings about who currently has responsibility for braking, AND/OR

b) Change in crew roles due to anomalous aircraft behavior. Note that manual braking is often used because of something else going wrong or because the crew cannot solve a problem in the avionics or mechanical systems. These activities could also change how crew resources are managed and thus cause confusion

Possible new requirements for S2:

1. Feedback must be provided to the crew about the status of manual braking and whether it is being provided by any crew member.
2. Feedback must be provided to the crew when landing, manual braking is needed, and it is not being provided.

<u>Control action provided but not followed</u>

STPA Step 2b identifies causal factors that can lead to a hazard when safe control actions are provided but not followed. This section analyzes the case where CREW.1a1 is provided but not followed:

Crew provides manual braking when needed but manual brakes are not applied. Figure 5.7 shows the parts of the control structure analyzed for STPA Step 2b.

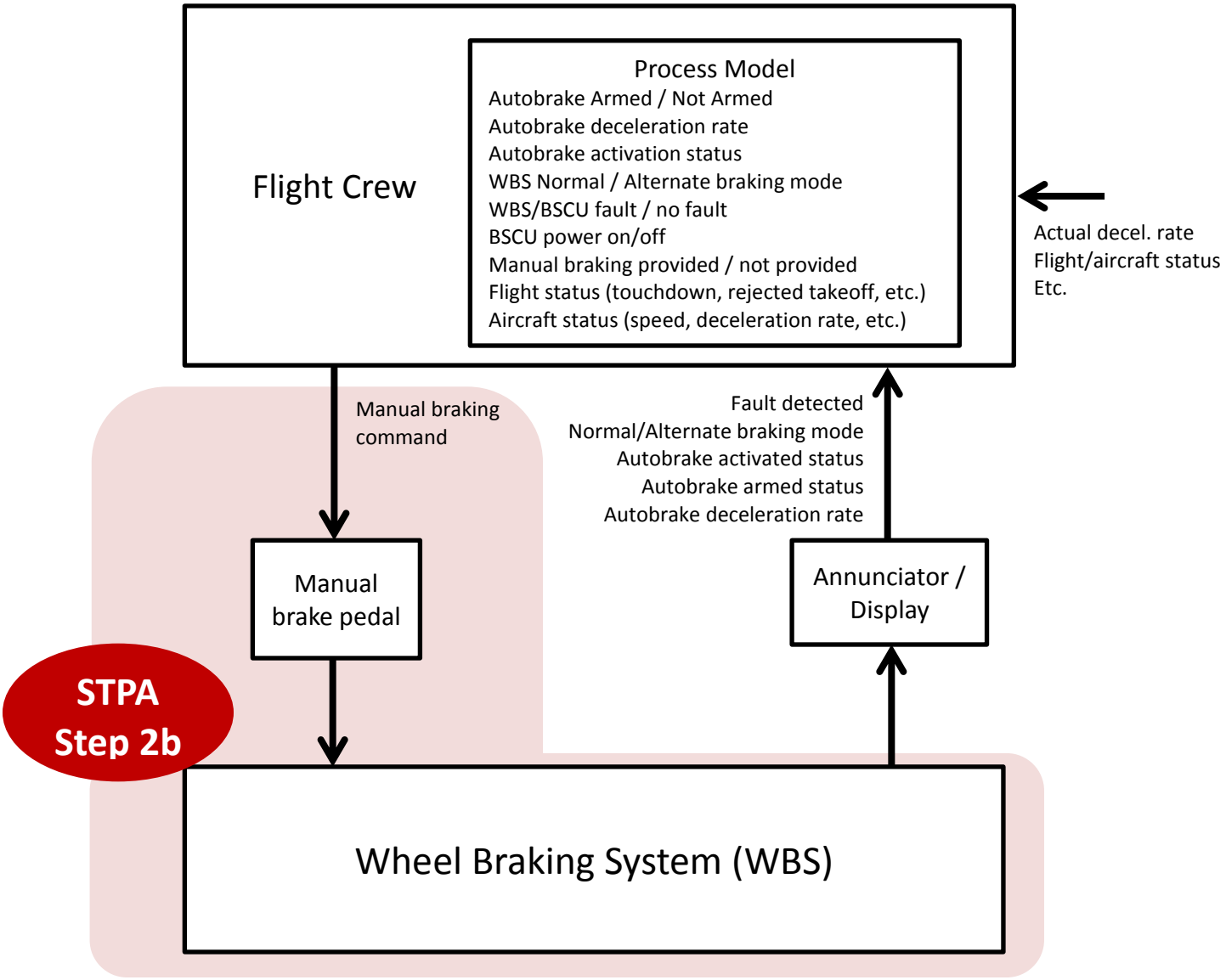

Figure 5.7: Control structure analyzed in STPA Step 2b

The following scenarios are based on the description of the WBS in ARP 4761 and the derived assumptions explained in the WBS description in Appendix A.

Scenario 3: Crew's manual braking commands are not be effective because the WBS is in alternate braking mode and the blue anti-skid valve is closed. Reasons for the lack of implemented action include:

a) The WBS would be in alternate braking mode could be because the BSCU has detected an internal fault in both channels and closed the green shut-off valve, AND/OR

b) The blue anti-skid valve would be closed could be because the BSCU internal fault was such that the CMD2 output incorrectly commanded all valves closed. This situation would erroneously close the blue-anti-skid valve because the BSCU logic depends on each CMD disabling its own outputs in the event of a fault inside the CMD.

Possible Requirement for S3: The BSCU fault protection must not rely on faulted components correctly disabling themselves

Scenario 4: The crew's manual braking commands may not be effective because the WBS is in alternate braking mode and the blue anti-skid valve is pulsed excessively. Reasons for the lack of implemented action include:

a) The WBS would be in alternate braking mode because the pilots disabled the green hydraulic system (perhaps for another issue), AND

b) The blue anti-skid valve would be pulsed excessively because the BSCU incorrectly believes the wheels are continuously skidding (incorrect process model, a BSCU analysis treats this scenario in more detail).

c) The BSCU might incorrectly believe the wheels are continuously skidding because the wheel speed feedback incorrectly indicates a sudden deceleration. This could be due to:

    a. A faulty wheel speed sensor,

    b. Incorrect design assumption about what deceleration characteristics indicate wheel slipping

Possible Requirements for S4:

1. The BSCU must be provided with other means of detecting wheel slipping than relying on wheel speed feedback;
2. The BSCU must provide additional feedback to the pilots so they can detect excessive anti-skid pulsing and disable it (e.g. by powering off the BSCU)

Autobrake Controller

This section describes causes of **AC.1a1**, which is an Autobrake control action that can be hazardous if not provided:

AC.1a1: Autobrake does not provide brake command when Autobrake has been armed and touchdown or rejected takeoff occurs

The causes of this unsafe control action are considered by analyzing the parts of the control structure highlighted in Figure 5.8.

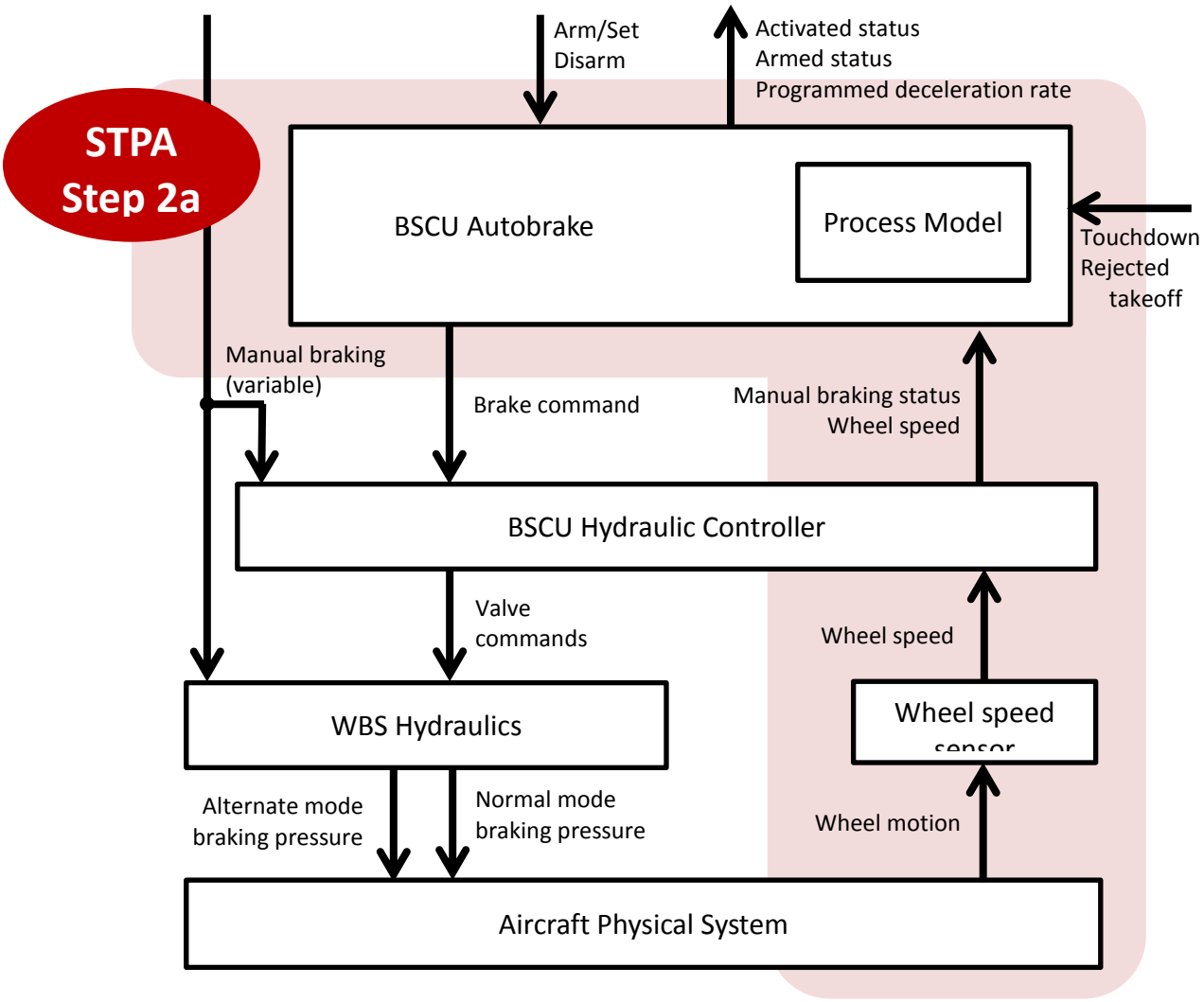

Figure 5.8: Control structure analyzed in STPA Step 2a to identify causes of unsafe control

Beginning with the unsafe control action and working backwards, scenarios can be identified by developing explanations for each causal factor in succession. The following scenarios are based on the description of the WBS in ARP 4761 and the derived assumptions explained in the WBS description.

UNSAFE CONTROL ACTION – AC.1a1: Autobrake does not provide brake command when Autobrake has been armed and touchdown or rejected takeoff occurs

Scenario 1: Autobrake believes the desired deceleration rate has already been achieved or exceeded (incorrect process model). The reasons Autobrake may have this process model flaw include:

a) The wheel speed feedback contains rapid pulses (inadequate feedback). This could make the actual aircraft speed difficult to detect and an incorrect aircraft speed might be assumed.

b) The feedback may contain rapid pulses could be because the runway is wet and anti-skid functionality is pulsing the wheel brakes

Possible Requirement for S1: Provide additional feedback to Autobrake to detect aircraft deceleration rate in the event of wheel slipping (e.g. inertial sensors)

Scenario 2: Autobrake has already brought the aircraft to a stop.

a) As soon as Autobrake detects the aircraft has stopped, it immediately releases the brakes and disarms itself (design flaw), AND

b) The aircraft may move as it is subjected to external wind, thrust, or other forces.

Possible requirements added for S2:

1. Autobrake must hold the aircraft stopped until the crew manually deactivate Autobrake;
2. Crew must be alerted when Autobrake is being used while other systems are providing thrust

Scenario 3: Autobrake believes touchdown or rejected takeoff has not occurred (incorrect process model). Reasons Autobrake may have this process model flaw include:

a) The method used to detect touchdown is inadequate for the runway or landing conditions, e.g. requires detection from all weight on wheels sensors, but aircraft lands on one wheel first; requires minimum wheel speeds, but runway is wet and wheels hydroplane

b) The conditions used to detect rejected takeoff do not occur when a rejected takeoff occurs, e.g. Autobrake may only detect a rejected takeoff if thrust levers are returned to the idle position, but they were not

c) The sensors used to detect touchdown or rejected takeoff malfunction or fail

Possible requirements added for S3:

1. Provide alternate means for Autobrake to detect touchdown or rejected takeoff in off-nominal conditions
2. Provide a way for the crew to manually trigger Autobrake in the event that touchdown or rejected takeoff is not detected

<u>Control action provided but not followed</u>

STPA Step 2b identifies causal factors that can lead to a hazard when safe control actions are provided but not followed. This section analyzes the case where **AC.1a1** is provided but not followed:

Autobrake provides the correct brake command when Autobrake is armed and touchdown or rejected takeoff occurs, but aircraft does not achieve necessary deceleration. Figure 5.9 shows the parts of the control structure analyzed for STPA Step 2bg.

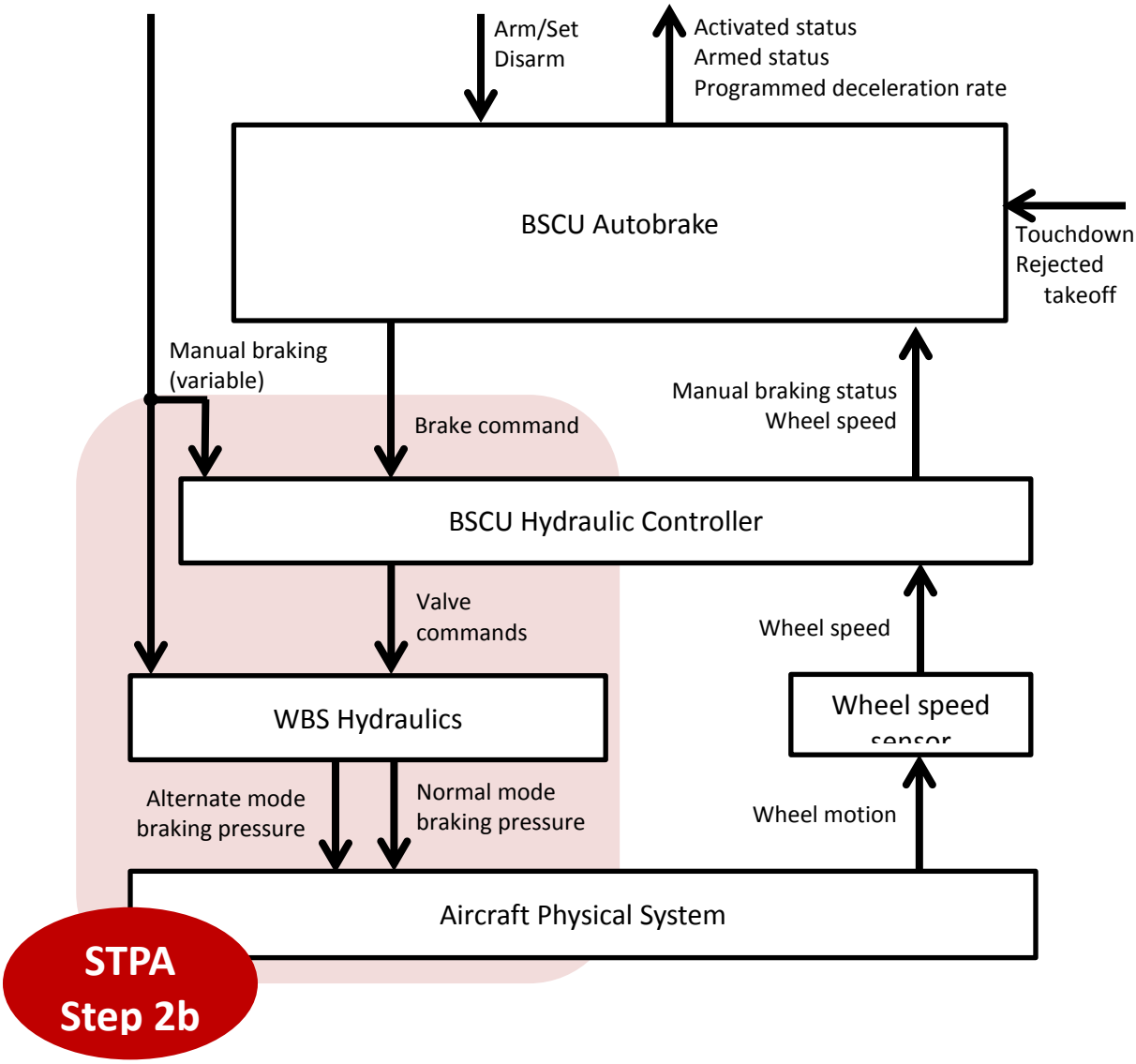

Figure 5.9: Control structure analyzed in STPA Step 2b

Scenario 3: Autobrake commands could be provided but not followed if the braking system is switched to alternate braking mode:

a) The Autobrake controller would be unaware of this switch and would continue to provide braking commands, AND

b) The Autobrake controller would also continue to provide feedback to the crew that Autobrake is activated (applying brakes), AND/OR

c) The system could be switched to alternate braking mode if a momentary fault is flagged in both channels of the BSCU Hydraulic Controller because faults are latched until the next power cycle

Possible Requirement for S3:

1. Additional feedback from the TBD (e.g. hydraulic system) must be provided to the Autobrake controller to detect when the system is in alternate braking mode and allow Autobrake to provide correct feedback to the crew regarding Autobrake availability. Although the crew will

likely be notified when the system is in alternate braking mode, Autobrake must not provide conflicting feedback to the crew

Scenario 4: Autobrake commands could be provided but not followed if a fault occurs in the green hydraulic system or the green hydraulic system is manually disabled by the crew.

a) The Autobrake controller would be unaware of this switch and would continue to provide braking commands, AND

b) The Autobrake controller would also continue to provide feedback to the crew that Autobrake is activated (applying brakes).

Possible requirements added for S4:

1. The TBD (e.g. hydraulic system) must provide additional feedback to the Autobrake controller to detect when the green hydraulic system can no longer provide sufficient pressure for effective Autobraking.
2. Although the crew will likely be notified of problems in the green hydraulic system, Autobrake must not provide conflicting feedback to the crew.

Hydraulic Controller

This section describes causes of **HC.3a1**, which is a BSCU Hydraulic Controller control action that can be hazardous if not provided:

> **HC.3a1**: HC does not provide a position command to the valve when brake commands are received

The causes of this unsafe control action are considered by analyzing the parts of the control structure highlighted in Figure 5.10.

Beginning with the unsafe control action and working backwards, scenarios can be identified by developing explanations for each causal factor in succession. The following scenarios are based on the description of the WBS in ARP 4761 and the derived assumptions explained in the WBS description.

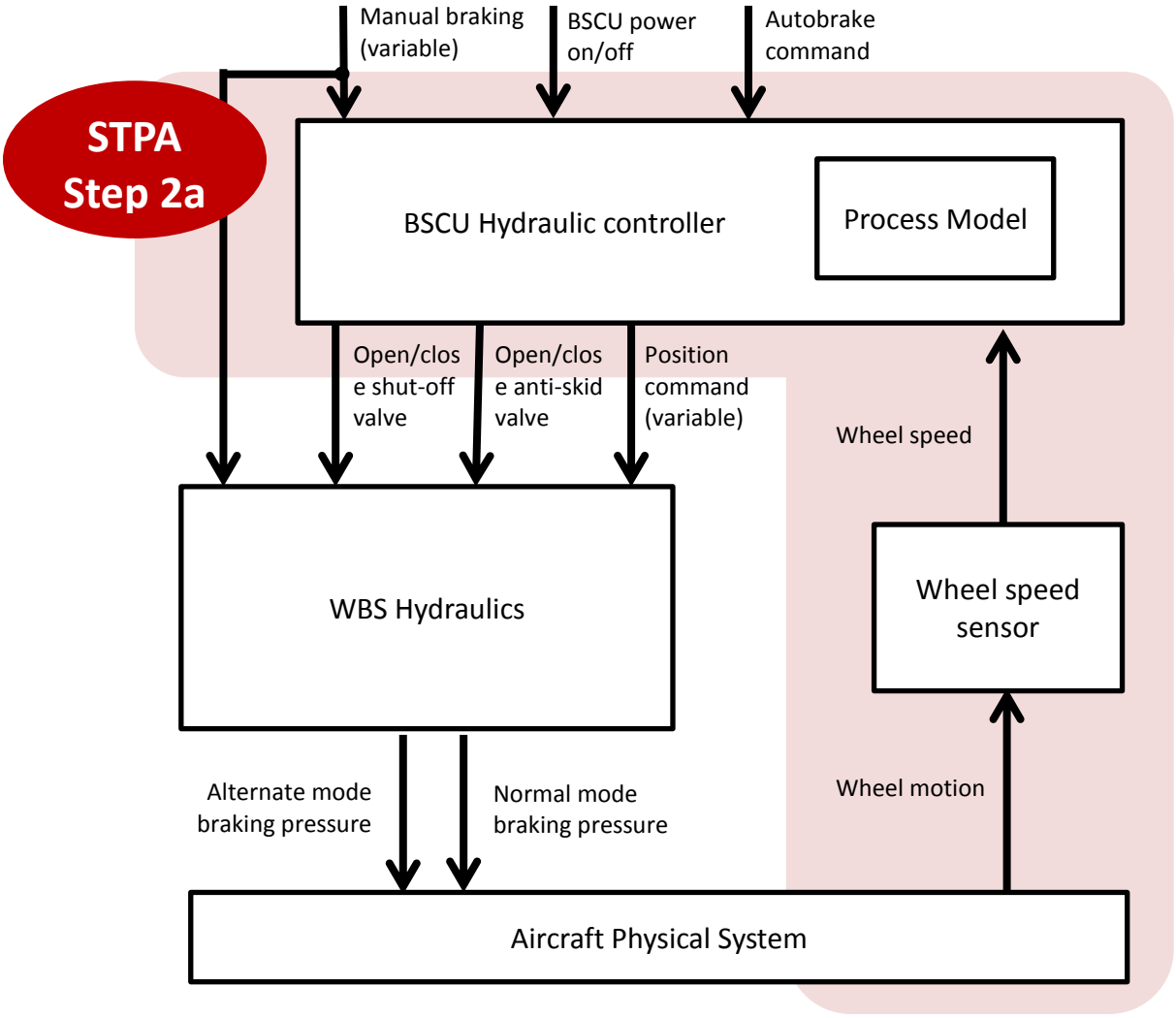

Figure 5.10: Control structure analyzed in STPA Step 2a to identify causes of unsafe control

UNSAFE CONTROL ACTION HC.3a1: HC does not provide a position command to the valve when brake commands are received

Scenario 1: HC does not provide a position command when brake commands are received might be because the brake commands are being sent by the Autobrake controller and a manual brake command was received before or during the Autobrake command. Possible contributions to this scenario include:

a) Manual braking command is received because it was intentionally provided by one of the pilots, OR

b) Manual braking command unintentionally provided by one of the pilots (e.g. foot on the pedal during landing or a bump), OR

c) Another disturbance such as a hard landing or sensor failure trips a manual braking command.

Possible Requirements for S1:

1. Provide feedback to the crew indicating when manual braking commands have been received from either pilot;
2. The Autobrake controller must disarm itself and notify the pilots when manual braking commands are received.
3. Provide additional feedback to Autobrake to detect aircraft deceleration rate in the event of wheel slipping (e.g. inertial sensors)

Scenario 2: Both MON1 and MON2 detect a momentary abnormality in their respective power supplies at any point since the initial power on. Causes of this scenario include:

a) MON1_valid and MON2_valid outputs are latched, AND

b) the BSCU HC will stop providing position commands until power off/on commands are received

Possible Requirements for S2: The Autobrake controller must disarm itself and notify the pilots when the BSCU HC has detected faults and cannot follow Autobrake commands

<u>Control action provided but not followed</u>

STPA Step 2b identifies causal factors that can lead to a hazard when safe control actions are provided but not followed. This section analyzes the case where **HC.3a1** is provided but not followed:

HC provides correct position command to CMD/AS valve when Autobrake commands are received, but aircraft does not achieve necessary deceleration. Figure 5.11 shows the parts of the control structure analyzed for STPA Step 2b, while Figure 5.12 shows the same control structure with additional detail regarding the WBS hydraulics as described in ARP 4761.

The following scenario is based on the description of the WBS in ARP 4761 and the derived assumptions explained in the WBS description in Appendix A.

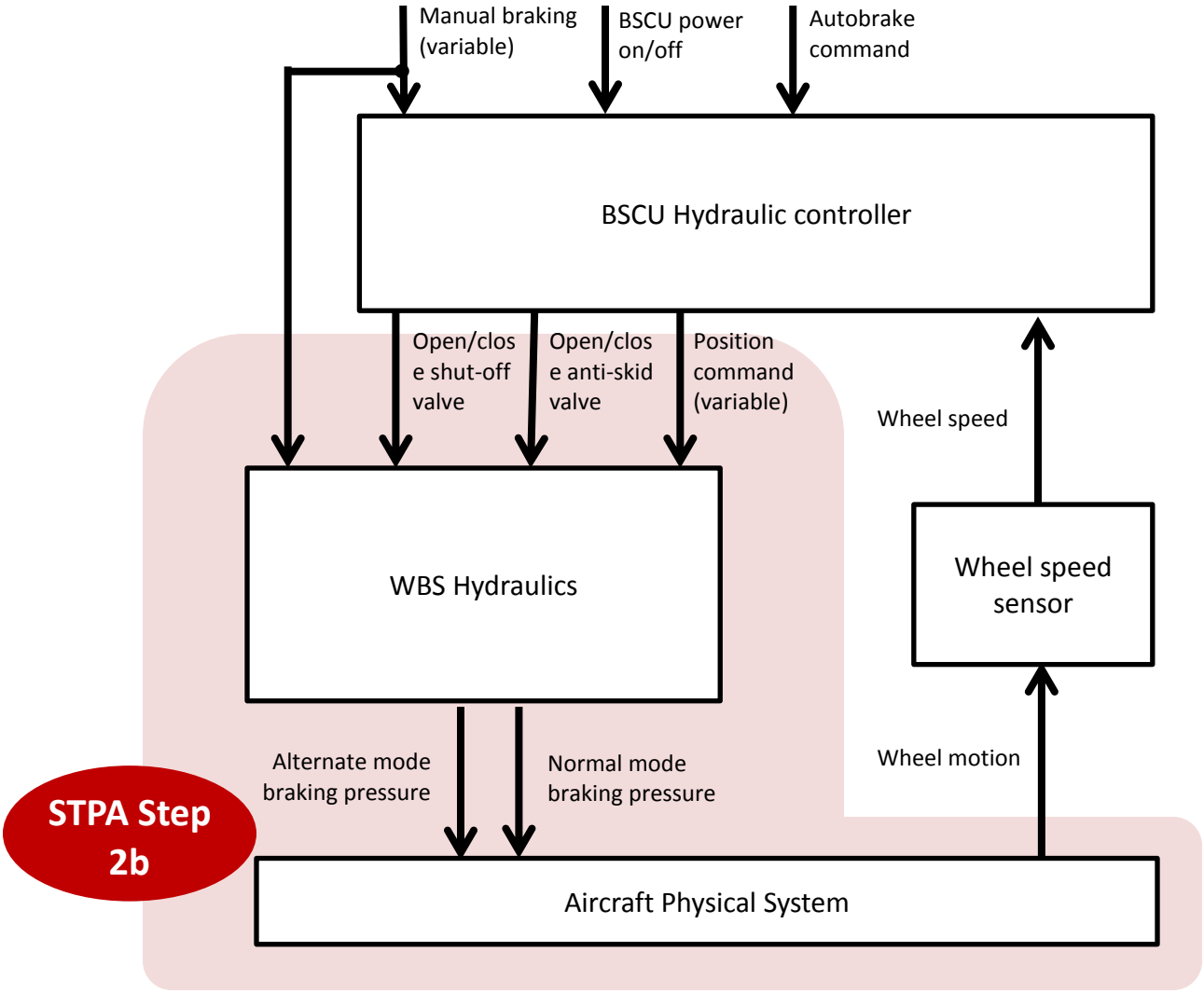

Figure 5.11: Control structure analyzed in STPA Step 2b

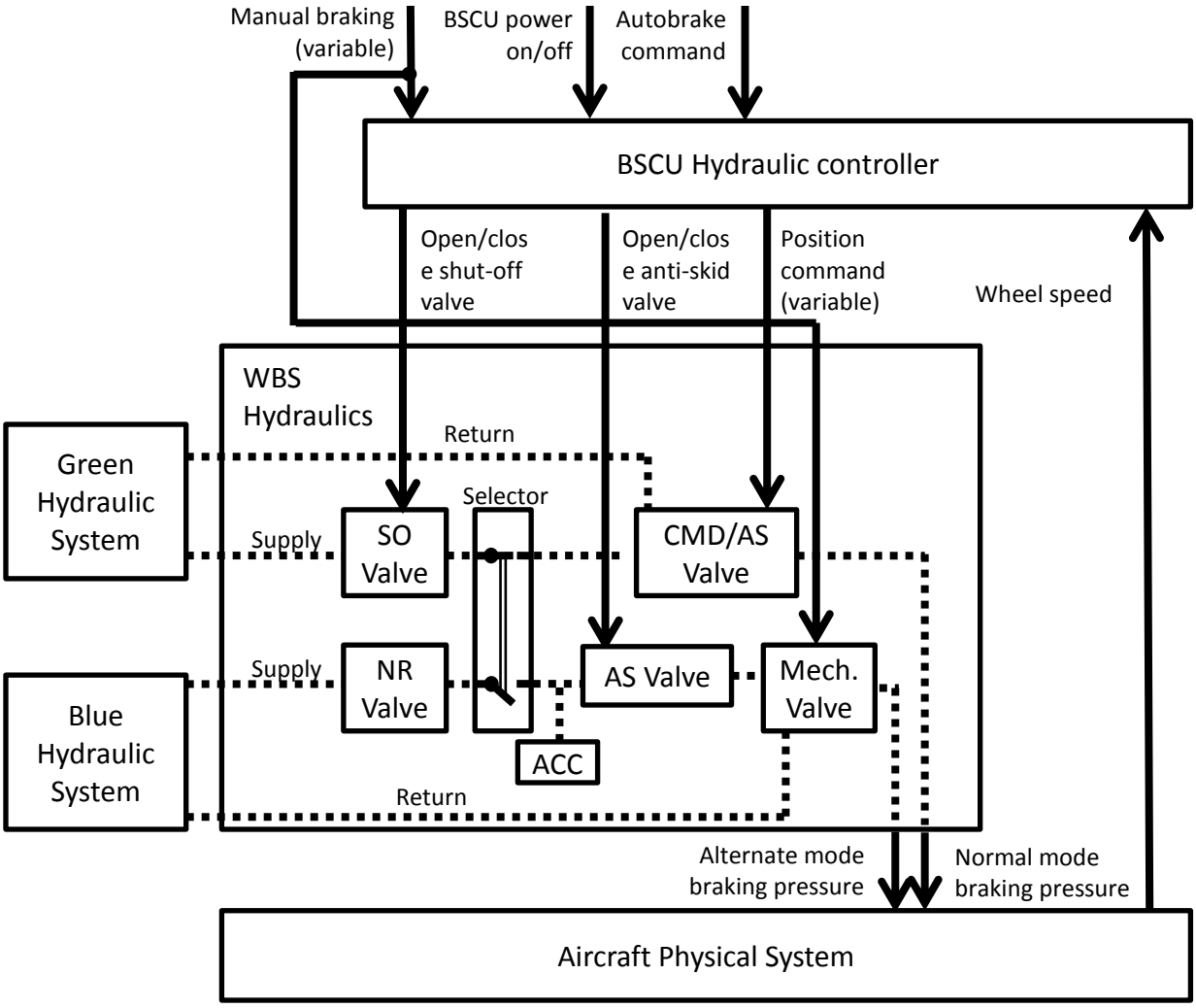

Figure 5.12: Detailed control structure used for STPA Step 2b

Scenario 3: Aircraft does not achieve the necessary deceleration despite correct position commands from the HC might be because the CMD/AS valve is sluggish. This condition is not (and cannot be) detected by WBS hydraulics and BSCU Hydraulic Controller. Causes of this scenario include:

a) Pilots and Autobrake actions are appropriate for certain conditions. That is, they have a 'correct' control algorithm (procedure), but an incorrect understanding of valve behavior
   - Pilots arm Autobrake before landing, THEN
   - The aircraft lands and Autobrake activates, THEN
   - Autobrake immediately begins to increase the commanded brake pressure to achieve the desired deceleration rate, but the deceleration rate cannot be achieved.
   - Autobrake commands soon request maximum brake pressure, but with little effect.

b) The CMD/AS valve continues to open slowly (Inadequate actuator behavior)

c) Pilots process model slowly degrades over time

- Upon landing, the pilots were informed that Autobrake has been triggered and is actively applying the brakes (Process model issue in both the crew and autobrake)
- The pilots notice some amount of deceleration, and the rate continues to increase.
- However Autobrake has no functionality to detect insufficient deceleration, and no mechanism to notify the pilots of such a condition (inappropriate or lack of feedback)

d) Crew process model is updated, but too slowly for the dynamics of situation:
- At some point after initial touchdown, the pilots realize that although the deceleration rate is improving, the amount of braking is insufficient to stop before the end of the runway.
- The pilots override Autobrake with the normal amount of brake pedal force, as they have no way of knowing Autobrake had already been requesting maximum braking.
- Autobrake becomes disarmed and disabled, and ceases to request maximum brake pressure.

e) Mode changes, resulting in both inadequate process model and incorrect algorithm (procedure)
- Instead of receiving Autobrake commands, the hydraulic controller begins receiving and obeying manual pilot commands for a lower amount of braking force.
- The pilots realize that additional brake pedal force is necessary, and eventually request maximum braking once more. Crew has correct model of necessary braking force and runway conditions, but incorrect model of the operating mode of braking system
- System evolves and by this time it is no longer possible to stop before the end of the runway and no longer possible to takeoff, and the aircraft overruns the runway.

6. Comparing STPA with ARP 4761

Because the goals are so different, it is difficult to compare the ARP 4761 process and STPA, particularly in terms of comparing the detailed results. This section tries to compare the approaches at a philosophical level by comparing the processes involved, with specific examples added where possible. While the types of results are clearly very different, it is important to compare them with respect to their use within the larger system engineering context. In what ways are the results useful? Are they complete?

Because a generic commercial aircraft WBS is relatively simple and primarily electromechanical, some of the differences are not so apparent in the particular example used. The advantages of STPA are more obvious in complex systems. It is with increased levels of complexity that factors other than component failures become particularly significant in accident causation. We were surprised, however, that we did find some safety-related design flaws that were not (and could not be?) identified in the ARP 4761 WBS example, some of which have been associated with actual aircraft accidents.

Another problem is that the WBS example in ARP 4761 is incomplete and was only intended as an example and not a complete analysis. Where possible, we try to compare the parts of the example that were analyzed using both approaches. But the focus of the comparison in this section will be on the philosophical and process differences between the two approaches.

Table 6.1 summarizes some of the differences in philosophy, goals, and analysis approach.

TABLE 6.1: Differences between the Two Processes

	ARP 4761 Safety Assessment Process	STPA Hazard Analysis Process
Underlying Accident Causality Model	Assumes accidents are caused by chains of component failures and malfunctions.	Assumes accidents are caused by inadequate enforcement of constraints on the behavior and interactions of system components.
	Focuses on component failures, common cause/mode failures.	Focuses on control and interactions among components, including interactions among components that have not failed as well as individual component failures.
	Consideration of limited (mostly direct) functional interactions among components.	Identifies indirect as well as direct unsafe functional relationships among components
	Safety is equated with reliability.	Safety is treated as a different (and sometimes conflicting) system property than reliability.
Goals	Safety assessment.	Hazard analysis.
	Primarily quantitative, i.e., to show compliance with FAR/JAR 25.1309. Qualitative analyses (e.g., CCA and DAL) are used where probabilities cannot be derived or are not appropriate.	Qualitative. Goal is to identify potential causes of hazards (perform a hazard analysis) rather than a safety assessment. Generates functional (behavioral) safety requirements and identifies system[9] and component design flaws leading to hazards.
Results	Generates probabilistic failure (reliability) requirements for the system and components. Likelihood analysis.	Generates functional safety requirements. Identifies design flaws leading to hazards.
	Likelihood (and severity) analysis	Worst case analysis.
Role of humans (operators) in the analysis	Crew and other operators are not included in analysis except as mitigators for the physical system component failures. Human factors is treated separately from and not addressed by ARP 4761.	Crew and operators are included as integral parts of the system and the analysis.
Role of software in the analysis	Does not assign probabilities to software. Instead identifies a design assurance level (DAL) and assumes rigor of assurance equals achieving that level.	Probabilistic software failure requirements are not generated or used. Instead treats software in same way as any controller, hardware or human. Impact of behavior on hazards analyzed directly and not indirectly through design assurance.

[9] Here we are using "system" in the general sense to denote the entire system being considered, such as the aircraft or even a larger transportation or airlines operations system in which the aircraft (and its subsystems) is only one component.

	Requirements assumed to be complete and consistent.	. Generates functional safety requirements for the software and other components to eliminate or control system hazards related to software behavior.
	Safety assessment considers only failures due to requirements implementation errors.	All software behavior is considered (not just "failures") to determine how it could potentially contribute to system hazards
Role of operations in the analysis	Operations generally not included except for generating installation and maintenance requirements	Operations and the overall safety management system can potentially be included (although it was not in this report).
Process	Starts with aircraft functions and loss of functions	Starts with hazards and losses.
	Iterative, system engineering process that can start in concept formation stage	Iterative, system engineering process that can start in concept formation stage
Cyber Security and other system properties	Not addressed by ARP 4761.	STPA-Sec integrates safety and security analysis (not shown in this report)

6.1 Underlying Accident Causality Assumptions

The difference between the two approaches starts with the underlying accident causality model. An accident causality model underlies all our efforts to engineer for safety. Basically it provides an explanation for why accidents occur and imposes patterns on accident causation. You may not be aware you are using one, but you are: Our mental models of how accidents occur determines how we investigate and prevent accidents.

Traditionally, accidents are assumed to be caused by chains of component failures and malfunctions. *Airworthiness* is defined in ARP 4761 as "the condition of an item (aircraft, aircraft system, or part) in which that item operates *in a safe manner* to accomplish its intended function" [SAE ARP 4761, p. 7]. Notice that "failure" is not mentioned. In contrast, the functional hazard assessment and the safety assessment process in ARP 4761 is defined in terms of identifying failure conditions:

> "A Functional Hazard Assessment (FHA) is conducted at the beginning of the aircraft/system development cycle. It should identify and classify the failure condition(s) associated with the aircraft functions and combinations of aircraft functions. These failure condition classifications establish the safety objectives" [SAE ARP 4761, p. 12].
>
> "The goal in conducting the FHA is to clearly identify each failure condition along with the rationale for its classification. After aircraft functions have been allocated to systems by the design process, each system which integrates multiple aircraft functions should be re-examined using the FHA process. The FHA is updated to consider the failure of single or combinations of aircraft functions allocated to a system. The output of the FHA is used as the starting point for conducting the Preliminary System Safety Assessment (PSSA)" [SAE ARP 4761, p. 15].
>
> "The PSSA is a systematic examination of the proposed system architecture(s) to determine how failures can cause the functional hazards identified by the FHA. The objective of the PSSA is to

establish the safety requirements of the system and to determine that the proposed architecture can reasonably be expected to meet the safety objectives identified by the FHA" [SAE ARP 4761, p. 15].

The ARP 4761 process, by definition, therefore equates the "safe manner" in the airworthiness definition with lack of failures and failure conditions. ARP 4761 defines a *failure* as a "loss of a function or a malfunction of a system or a part thereof" [SAE ARP 4761, p. 9]. A *malfunction* is defined as "operation outside specified limits" [SAE ARP 4761, p. 10] and thus does not include specification errors or behaviors within specific limits but where the specified behavior is unsafe.

STPA is based on a newer and more inclusive model of accident causation (STAMP) where accidents are assumed to be caused by inadequate enforcement of constraints on the behavior and interactions of system components. Examples of constraints related to ground movement are that the aircraft must not decelerate after V1 or that uncommanded movement must not occur when the aircraft is parked. The "safe manner" of the airworthiness definition is equated in STPA with the satisfaction (enforcement) of behavioral safety constraints. Safety constraints specify system behaviors that prevent accidents (losses) and hazards (system states that can lead to an accident). Safety constraints can be violated not only by failures and malfunctions but also by interactions among system components ("items").

The STAMP definition of accident causality includes all the things included in the ARP 4761 definition but also includes additional causes not included, such as software and system requirements errors, system design errors (perhaps caused by complexity that leads to misunderstanding all the possible interactions among the components), and human mismanagement of automation and other sophisticated human behavior leading to hazardous states. The goal of STPA is to include all these additional causes of accidents in the hazard analysis.

As just one example that involves aircraft braking, consider the Lufthansa A320 Flight 2904 overrun of the runway at Warsaw in 1993 in heavy rain and tail wind conditions [Warsaw]. The left and right landing gear did not touch down at the same time. By design, the ground spoilers and engine thruster deployment depend on compression of both landing gear struts, and therefore their activation was delayed. Touchdown was also designed to be indicated by wheel speed, but the water on the runway resulted in hydroplaning. Because the software did not believe the aircraft had landed, it disallowed pilot inputs to activate the thrust reversers until farther down the runway than normal. The aircraft overran the runway, resulting in two fatalities.

None of the aircraft components involved in the Warsaw accident, including the pilots, "failed" or even "malfunctioned." Each satisfied their requirements. The basic problem was a flaw in the system design combined with inaccurate wind speed/direction information that led to unsafe aircraft behavior. There have been other variants on this same braking accident cause including a T-1A Jayhawk aircraft accident [T-1A]. STPA identifies this type of design and requirements flaw and found something similar for the wheel brakes (versus the reverse thrusters in the actual accident). We did not identify the exact scenario because we did not analyze the reverse thrusters but only the wheel brake system.

ARP 4761 does consider some interactions, but limits them to those among items or functions that arise from common cause/mode failures, with the emphasis again on failures and malfunctions. STPA includes more types of unsafe interactions including indirect interactions and interactions among components that have not failed or even malfunctioned.

In essence, ARP 4761 and most current safety standards and engineering processes equate safety to reliability. The definition of *reliability* in ARP 4761 is the standard one: "the probability that an item will perform a required function under specified conditions, without failure, for a specified period of time" [SAE ARP 4761, p. 10]. Safety, however, in general is equated with a lack of accidents (losses) rather than failures. All failures do not lead to accidents (losses) and all accidents are not caused by failures.

The left circle in Figure 6.1 represents scenarios involving component and functional failures. The right circle represents scenarios leading to accidents. While there is overlap, many failure scenarios do not involve losses and, more important, many accident (loss) scenarios do not involve failures.

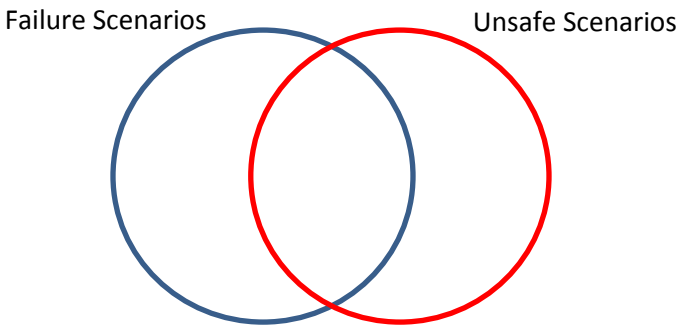

Figure 6.1: Failure scenarios vs. unsafe scenarios

Because accidents can occur without component failure or when the system operates outside the "specified conditions," reliability and safety are not equivalent nor is one a subset of the other.[10]

In fact, equating the two properties is not unique to ARP 4761 and has been going on for a long time. Until relatively recently, it was possible to thoroughly test the electromechanical systems being built and basically eliminate most design flaws. Software was used in only a limited and straightforward way (such as numerical calculations). Human roles in systems involved direct control following carefully designed and tested procedures. Equating safety and reliability under these conditions was a useful approximation.

But things are changing. Software is now playing much more sophisticated roles in systems, resulting in enormously more complex systems being built. Exhaustive testing is no longer possible. The role of human operators has also changed: To a much larger degree, humans are managing automation rather than providing direct control. The assumptions of the past, most notably that accidents are caused by failures or component malfunctions, are no longer true, and approaches based on those assumptions are becoming less and less effective [Leveson, 2012].

Fault trees and FMEA, which are the primary analysis methods used in the ARP 4761 WBS example and recommended in ARP 4761, are both failure based and were both created over 50 years ago. To effectively analyze today's systems, more inclusive analysis techniques are needed.[11]

---

[10] Safety and reliability may also conflict, where increasing safety can decrease reliability and increasing reliability can decrease safety. A discussion of these conflicts is beyond the scope of this report. The interested reader is referred to [Leveson, 2012].

[11] As another aside, "analysis" is defined in ARP 4761 as "an evaluation based on decomposition into simple elements." Systems theory, which underlies systems engineering, was created because in complex systems, analysis based on such decomposition is often not possible, e.g., the common description of complex systems as those where "the whole is greater than the sum of the parts." Analysis based on analytic reduction (as defined in ARP 4761) cannot handle emergent system properties, of which safety is one [Leveson, 2012].

Why has commercial aviation gotten away with equating reliability and safety for so long? Until relatively recently, new technology and especially software, has been introduced very slowly into aircraft, while depending on standard designs used over long periods of time, with most of the design errors removed during exhaustive testing or from use over time. Complexity was limited so engineers could depend on redundancy as a primary protection mechanism during operation and could use common cause analysis (CCA) to detect common failure channels.

When few major design changes occurred, this approach was reasonable. But with the increasing pace of introducing new technology and greater design complexity in aircraft (essentially implemented by the use of software), it is becoming less effective and simply blaming accidents with complex causes on pilot error does not solve the problem. New analysis tools are needed to help designers systematically identify the hazardous scenarios and create functional requirements that will eliminate or control them. STPA is based on systems theory and control theory rather than on reliability theory. As such, it is capable of identifying more paths to accidents than just those created by failure conditions. It also integrates software and humans directly into the analysis so that a larger set of causes can be considered.

One of the side effects of equating safety and reliability is that the use of reliability engineering tools, such as FTA and FMEA, leads designers to rely on redundancy and monitors as the most appropriate or best design solution. Adding redundancy is not necessarily the optimal solution to achieve a given level of safety because the underlying cause of a hazard goes unaddressed. Other solutions that might be cheaper or more effective for the types of hazard causes occurring (such as unintended interactions among system components and functions or software requirements errors) are less likely to be considered. By digging deep into the cause of the unsafe interactions, more effective solutions (than simply adding redundancy and monitoring) may be identified [Leveson, 1995; Leveson, 2012].

STPA accounts for subsystem interaction and safety-related design flaws and, in fact, assumes that not only can causal factors be dependent, but also that the behavior of (non-failed) components might be highly influential on other aspects of the system. As an example, in the WBS example, STPA identifies how normal anti-skid behavior could potentially interfere with Autobrake software behavior by affecting the wheel speed and the wheel speed sensor feedback and making the actual deceleration rate difficult to detect. Once identified, these influences can be controlled with functional safety requirements such as additional feedback to adequately detect deceleration or by defining the appropriate Autobrake behavior in these situations. As another example, the STPA analysis of the BSCU hydraulic controller identified potentially hazardous interactions between the Autobrake and the hydraulic control software where the hydraulic controller might start intentionally ignoring Autobrake commands if it believes manual braking commands have been received.

These types of interaction problems could not be found by CCA, Zonal Analysis, DD, or other techniques in the ARP 4761 process. In these cases, nothing failed and thus there were no common cause/mode failures involved.

The automotive industry is facing these same problems. An automobile today has about five times the amount of software as on the most sophisticated military aircraft and four to five times that of a new commercial aircraft. New software-supported features are common. Many recent automobile recalls have been related to unidentified interactions among these features and seemingly unrelated functions that were not identified during development [Suo, 2014]. For example, in 2014, General

Motors recalled 3.3 million vehicles due to potential interactions between key chains, ignition switches with inadequate torque requirements, and safety systems that could result in dangerous situations such as airbags being disabled in the event of a crash.[12] The authors have heard privately from automotive engineers about many feature interaction problems that have not been made public.

In general, automobile recalls involving software are becoming common. 89,000 Ford Fusion and Escape vehicles were recalled in 2013 because software programming could not handle certain operating conditions resulting in flammable liquids being released near hot surfaces and engine fires.[13] Chrysler recalled 141,000 vehicles in 2013 due to computer glitches that erroneously illuminated warning lights and caused instrument cluster blackouts.[14] STPA has been successfully used in an automotive research context to identify feature interactions and other potentially unsafe software behavior before they lead to accidents [Placke, 2014].

A recent paper by Bartley and Lingberg identified similar issues associated with increased coupling and complexity in aircraft design and integrated modular avionics (IMA) [Bartley and Lingberg, 2011]. That paper questioned the assumption that the current certification approach is sufficient to tackle the challenges inherent in tomorrow's avionics systems. Recent research has shown how STPA can be used to tackle these challenges that arise from functional coupling (vs. physical coupling) when multiple avionics applications are integrated onto the same platform [Fleming and Leveson, 2014]

6.2 Analysis Goals

The ultimate goal of ARP 4761 is to show compliance with the airworthiness standards specified in 14CFR/CS 25.1309 (for transport category aircraft). The goals are primarily quantitative although, for software, qualitative goals are included through the DAL (discussed further in Section 6.4).

In contrast, the goal of STPA is not assessing the safety of the system design, but hazard analysis. Hazard analysis identifies the causes of hazards in order to create functional safety requirements and to accumulate the knowledge necessary to eliminate or mitigate the hazards in the design. The results of STPA could be used to support a safety assessment, but that is not part of STPA.

ARP 4761 defines *hazard* [SAE ARP 4761, p. 9] as "a potentially unsafe condition resulting from failures, malfunctions, external events, errors, or a combination thereof." STPA starts from a definition of hazard (which is derived from MIL-STD-882) as "a system state or set of conditions that, together with a particular set of worst-case environment conditions, will lead to an accident (loss)" [Leveson, 1995; Leveson, 2012]

The two definitions are similar in that both define hazards as conditions, but the STAMP definition does not include the causes of the hazards. That difference is not terribly significant except for the fact that while ARP 4761 includes "errors" as a cause of hazards in the definition of "hazard," it does not effectively include "errors" in the safety assessment process.

---

[12] http://www.washingtonpost.com/cars/gm-recalls-35-million-buick-cadillac-chevrolet-gmc-vehicles-for-ignition-switch-and-other-flaws/2014/06/17/a60d09c2-f636-11e3-afdf-f7ffea8c744a_story.html

[13] http://corporate.ford.com/news-center/press-releases-detail/pr-ford-produces-fix-in-voluntary-37491

[14] http://www.autoblog.com/2013/10/02/2014-jeep-grand-cherokee-ram-truck-recall/

While there is a claim that errors are included as a cause of hazards, we found the word "error" in only a very few places in the general analysis descriptions in the ARP and in the WBS example. One of the problems may be that fault tree boxes, as defined in ARP 4761, represent "events" but errors are not events. On page 100, there is a fault tree with a box that says "Hardware error" as well as one that says "Software error." No information is given about what these errors might be and the boxes are not refined further. The accompanying text states that these boxes provide useful information about independence, which can be used to assign a DAL.

The only type of "error" that is included in the ARP 4761 WBS example is inadvertent operation of a system component. Even then, the fault tree analysis interprets inadvertent activation in terms of lower-level component failures for the purposes of obtaining a probability. In another fault tree (page 218), a box in the fault tree is labeled "BSCU 1 CPU Function Design Error Causes Bad Data" and assigned a probability of 0.00E+00 and Level A. Does this really mean that an assumption is made that Level A software has a zero probability of having an error?

Another problem in handing errors using the ARP 4761 process is that design errors cannot be associated with probabilities. If there were enough information to obtain a realistic probability, there would be enough information to simply fix the error. Instead, in ARP 4761, errors are supposedly handled through design assurance level (DAL) processes. Page 25 says:

> *"The occurrence of software errors are probabilistic but not in the same sense as hardware failures. Unlike hardware failures, these probabilities cannot be qualified. Therefore numerical and categorical probabilities should not be indicated for software errors in fault trees. Any software analysis in an FTA should be expressed in terms of development assurances to protect against software errors. The analysis should be evaluated for compliance on a purely qualitative basis."*

While there are some in the software engineering community that believe the occurrence of software errors is probabilistic, they are not in the majority. It is a moot point, however, whether it is probabilistic or not because everyone seems to agree that these probabilities cannot be specified and used in the safety assessment. The problem of software errors is discussed further in Section 6.4.

Table 6.2 contrasts the difference between failures as defined in ARP 4761 and hazards as defined by System Safety (MIL-STD-882) and STPA.

TABLE 6.2: ARP 4761 Wheel Brake System Failures vs. STPA Hazards

Examples of ARP 4761 Failures (p178)	**Examples of STPA Hazards**
Function: Decelerate aircraft on the ground Failure Conditions (p178): - Loss of deceleration capability - Inadvertent Deceleration after V1 (Takeoff/RTO decision speed) - Partial loss of Deceleration Capability - Loss of automatic stopping capability - Asymmetric Deceleration	System Hazard **H4**: An aircraft on the ground comes too close to moving or stationary objects or inadvertently leaves the taxiway Deceleration-related hazards: - **H4-1**: Inadequate aircraft deceleration upon landing, rejected takeoff, or taxiing - **H4-2**: Deceleration after the V1 point during takeoff - **H4-3**: Aircraft motion when the aircraft is parked - **H4-4**: Unintentional aircraft directional control (differential braking) - **H4-5**: Aircraft maneuvers out of safe regions (taxiways, runways, terminal gates, ramps, etc.) - **H4-6**: Main gear wheel rotation is not stopped

	when (continues after) the gear is retracted

6.3 Outputs (Results) of the Analysis

Given these different goals, it is not surprising that the outputs are different. While both identify safety requirements, the types of requirements generated are very different.

The ARP 4761 safety requirements are generated from a quantitative failure analysis, along with a few design assurance requirements for software components and design requirements to support the independent failure assumptions of the quantitative analysis. The resulting requirements are primarily quantitative. STPA safety requirements are generated from the system-level hazards, identified unsafe control actions (Step 1), and unsafe behavioral scenarios (Step 2). The resulting requirements are all functional design requirements. Table 6.3 contrasts the two in terms of the system-level safety requirements generated.

TABLE 6.3: System Level Requirements Generated

ARP 4761 Process	STPA
Requirements from the FHA: 1) Loss of all wheel braking during landing or RTO shall be less than 5E-7 per flight. 2) Asymmetrical loss of wheel braking coupled with loss of rudder or nose wheel steering during landing or RTO shall be less than 5E-7 per flight. 3) Inadvertent wheel braking with all wheels locked during takeoff roll before V1 shall be less than 5E-7 per flight. 4) Inadvertent wheel braking of all wheels during takeoff roll after V1 shall be less than 5E-9 per flight. 5) Undetected inadvertent wheel braking on one wheel w/o locking during takeoff shall be less than 5E-9 per flight Two requirements from the CCA: 6) The wheel braking system and thrust reverser system shall be designed to preclude any common threats (tire burst, tire shred, flailing tread, structural deflection, etc.) 7) The wheel braking system and thrust reverser system shall be designed to preclude any common mode failures (hydraulic system, electrical system, maintenance, servicing, operations, design, manufacturing, etc.)	**Requirements from the system-level hazards** **SC1**: Forward motion must be retarded within TBC seconds of a braking command upon landing, rejected takeoff, or taxiing. **SC2**: The aircraft must not decelerate after V1. **SC3**: Uncommanded movement must not occur when the aircraft is parked. **SC4**: Differential braking must not lead to loss of or unintended aircraft directional control **SC5**: Aircraft must not unintentionally maneuver out of safe regions (taxiways, runways, terminal gates and ramps, etc.) **SC6**: Main gear rotation must stop when the gear is retracted.

In the ARP 4761 safety assessment process and example, the PSSA analysis continues to refine the high-level probabilistic requirements and derived design requirements generated from the fault trees

into more specific (but still probabilistic) requirements. Software requirements are stated in terms of design assurance levels. Crew requirements (as well as probabilities for engineers making a system design or specification error) are not included.

For example, the BSCU requirements generated are:

1. The probability of "BSCU Fault Causes Loss of Braking Commands" shall be less than 3.3E-5 per flight.
2. The probability of "Loss of a single BSCU shall be less than 5.75E-3 per flight.
3. The probability of "Loss of Normal Brake System Hydraulic Components" shall be less than 3.3E-5 per flight.
4. The probability of "Inadvertent braking due to BSCU" shall be less than 2.5E-9 per flight.
5. No single failure of the BSCU shall lead to "inadvertent braking."
6. The BSCU shall be designed to Development Assurance Level A based on the catastrophic classification of "inadvertent braking due to BSCU."

Additional requirements on other systems are also generated such as

7. The probability of "loss of Green Hydraulic Supply to the Normal brake system shall be less than 3.3E-5 per flight."

Finally, installation requirements and maintenance requirements are generated as well as independence requirements such as:

8. Each BSCU System requires a source of power independent from the source supplied to the other system.

Hardware and software safety requirements generated look like:

1. Each BSCU system will have a target failure rate of less than 1E-4 per hour.
2. The targeted probabilities for the fault tree primary failure events have to be met or approval must be given by the system engineering group before proceeding with the design.
3. There must be no detectable BSCU failures that can cause inadvertent braking.
4. There must be no common mode failures of the command and monitor channels of a BSCU system that could cause them to provide the same incorrect braking command simultaneously.
5. The monitor channel of a BSCU system shall be designed to Development Assurance Level A.
6. The command channel of a BSCU system may be designed to Development Assurance Level B.[15]
7. Safety Maintenance Requirements: The switch that selects between system 1 and system 2 must be checked on an interval not to exceed 14,750 hours.

For STPA, after the first high-level hazards have been identified, the six safety constraints shown in Table 6.3 are refined into a more detailed set of functional safety requirements that are associated with specific system components, including the crew, the software, and the component interfaces. The safety requirements are generated to prevent the causal scenarios identified by STPA.

Some example requirements on the crew generated from the STPA analysis are:

[15] The allocations in 5 and 6 could have been switched, designing the command channel to level A and the monitor channel to level B.

FC-R1: Crew must not provide manual braking before touchdown [CREW.1c1]

Rationale: Could cause wheel lockup, loss of control, or tire burst.

FC-R2: Crew must not stop manual braking more than TBD seconds before safe taxi speed reached [CREW.1d1]

Rationale: Could result in overspeed or runway overshoot.

FC-R3: The crew must not power off the BSCU during autobraking [CREW.4b1]

Rationale: Autobraking will be disarmed.

etc.

Example requirements that can be generated for the BSCU:

BSCU-R1: A brake command must always be provided during RTO [BSCU.1a1]

Rationale: Could result in not stopping within the available runway length

BSCU-R2: Braking must never be commanded before touchdown [BSCU.1c1]

Rationale: Could result in tire burst, loss of control, injury, or other damage

BSCU-R3: Wheels must be locked after takeoff and before landing gear retraction [BSCU.1a4]

Rationale: Could result in reduced handling margins from wheel rotation in flight.

Finally, some examples of requirements for the BSCU hydraulic controller commands to the three individual valves:

HC-R1: The HC must not open the green hydraulics shutoff valve when there is a fault requiring alternate braking [HC.1b1]

Rationale: Both normal and alternate braking would be disabled.

HC-R2: The HC must pulse the anti-skid valve in the event of a skid [HC.2a1]

Rationale: Anti-skid capability is needed to avoid skidding and to achieve full stop in wet or icy conditions.

HC-R3: The HC must not provide a position command that opens the green meter valve when no brake command has been received [HC.3b1]

Rationale: Crew would be unaware that uncommanded braking was being applied.

HC-R4: the HC must always provide a valve position command when braking commands are received.

The next level of more detailed requirements is derived from the causal scenarios generated to identify how each of these requirements could be violated. The specific additional requirements generated will depend on how the design engineers decide to prevent these scenarios, that is, the controls added to the design. If the scenarios are to be prevented by crew actions, then additional crew requirements will be needed. If they are to be eliminated or controlled through software or hardware controls, then those controls must be added to the hardware or design requirements.

For example, consider HC-R4, which requires the hydraulic controller to provide a valve position command when brake commands are received. Why might it not do so? Several scenarios were identified in Section 5 of this report. One scenario might be that the brake commands were sent by the Autobrake controller and a manual braking command was received before or during the Autobrake command. Possible contributors to this scenario are that the manual braking command was intentionally provided by one of the pilots (perhaps because he did not know Autobraking was set or

thought that it was not working property), the manual braking command was unintentionally provided by one of the pilots (e.g., foot on pedal during landing or a bump), or a physical disturbance such as a hard landing or sensor failure trips a manual braking command.

The potential for this hazardous scenario occurring might be eliminated or controlled by changes to the hydraulic controller behavior, the Autobrake behavior, crew behavior, and/or physical system behavior. These design and requirements decisions must be made by the design engineers, but the STPA analysis will provide them with guidance on the decisions needed and why. If changes are made to try to prevent the scenario, there must be further analysis to determine that the changes do not introduce new hazardous scenarios.

Note that redundancy might be an option here to deal with the causal scenario involving the sensor failure. Physical failures are not ignored in the STPA causal analysis; it is just that the connection of the failure to a system hazard is first determined before trying to prevent it. And looking at the impact of a sensor failure in the larger scope of the WBS design might provide insight into a different way of solving the problem that does not involve costly redundancy or monitors.

We understand that aircraft functional requirements generation is performed separately from the safety assessment process (see ARP 4754) and therefore the functional requirements generated from the STPA analysis might result from that process. Without the assistance of a hazard analysis process, however, important safety requirements may be inadvertently omitted. For example, there is a provision in AC 25.1309 (paragraph 8g, p. 11) that says the crew must be alerted to unsafe operating conditions. This requirement takes its source as 14CFR §25.1309 (c). ARP 4761 does not provide any guidance on how to do that as far as we can determine, nor does the ARP 4761 WBS example generate any such requirements. The STPA analysis can identify critical crew alerting requirements and did so in the STPA WBS analysis.

Many recent aircraft accidents have involved erroneous functional requirements, particularly system, software and interface requirements, which may indicate a need for a more rigorous and safety-oriented requirements generation process. After doing the STPA WBS analysis, we looked for wheel braking accidents that had actually occurred to determine whether the scenarios we identified were plausible. We found several accident reports related to design errors identified by the STPA analysis. One example is the accident scenario common to the Warsaw and T1-A accidents described in Section 6.1.

Another example where STPA did not find the exact scenario but something very similar occurred in an Air UK Leisure G-UKLL A320 accident at Ibiza, 1998 [Ibiza]. This accident involved the flight crew pushing the Autobrake arm buttons too quickly (inside of the duty cycle of the Autobrake discrete sensor). The Autobrake then did not engage because one channel registered the input while the other did not. The monitor sent a BSCU fault warning. However, both channels were faulted due to the initial disagreement, and thus the BSCU disengaged the Normal hydraulic system. Ultimately, the manual braking system on the alternate system did not work because there was a latent problem of residual soap in the valve, which did not get checked very often due to its limited use.[16] The reason that the STPA analysis did not find this exact scenario is that the WBS design used in both the ARP 4761 and our STPA analyses did not exactly match the A320-212 design. STPA did, however, find several similar scenarios in

[16] The maintenance problem here may be related to the ARP 4761 practice of allowing a backup or monitor to be assigned a lower severity level than the primary system.

the physical design on which the STPA analysis was performed. These scenarios involve fault detection, switching to alternate mode, inadequate arming of the Autobrake system, and delayed or missing feedback regarding all of these factors.

A final example occurred during an A320 landing at Cardiff in 2003 [Cardiff]. On final approach, the Electronic Centralized Aircraft Monitoring (ECAM) display showed a STEERING caption and the crew cycled the A/SKID & N/W STRNG switch in an attempt to reset the Brake and Steering Control Unit (BSCU). The indications were that it was successfully reset but after touchdown the aircraft did not decelerate normally. The commander pressed the brake pedals to full deflection without effect. He then selected maximum reverse thrust and the co-pilot cycled the A/SKID & N/W STRNG switch. The commander again attempted pedal braking, without effect, and the crew selected the A/SKID & N/W STRNG switch to OFF. The commander then braked to bring the aircraft to a halt about 40 meters from the end of the runway, bursting three main wheel tires. There was no fire and the passengers were deplaned on the runway through the normal exit doors. Analysis showed that it took 10 to 13 seconds for the commander to recognize the lack of pedal braking and there was no overt warning from the ECAM of the malfunction of the BSCU. Two safety recommendations were made to the aircraft manufacturer regarding improved warnings and crew procedures.

Our STPA analysis included different assumptions about the warning systems than are in the A320 but found similar results. One of the causal scenarios identified for the unsafe control action "Crew does not provide manual braking when there is no Autobraking and braking is necessary" involves the crew's incorrect belief about the state of Autobrake and BSCU (p. 39). Causal factor 2.b) states that "The crew would be notified that the Autobrake controller is still armed and ready, because the Autobrake controller is not designed to know when the BSCU has detected a fault. When the BSCU detects a fault it closes the green shut-off valve (making Autobrake commands ineffective), but the Autobrake system itself will not notify the crew." The analysis also found that the crew may behave inadequately due to multiple or conflicting messages. STPA found several other scenarios and causal factors that were also present in the Cardiff A320 accident; see for example Scenario 2.b (page 39) and Scenario 3.a (page 40).

The A320 braking system is not the same as the ARP 4761 WBS example, and so it is inappropriate to speculate about what BSCU fault would cause the same incident. However, the recommendations coming out of the investigation are general enough to show that whatever caused this hazard would have been mitigated with those safety requirements (loss of braking warning and FCOM instructions on what to do if there is a warning) and STPA has found them in our limited analysis.

One major difference in the results of the ARP 4761 process and STPA is that STPA does not involve a probabilistic analysis. Is probabilistic risk assessment necessary to ensure safety? We will avoid a religious argument here, but note that one of the most successful safety programs in history does not allow the use of probabilistic risk assessment. SUBSAFE, the U.S. nuclear submarine safety program, was created in 1963 after the loss of the U.S.S. Thresher. Before SUBSAFE, the U.S. Navy lost on average one submarine every three years. After SUBSAFE was implemented 51 years ago, the U.S. has never lost a submarine certified under the SUBSAFE procedures. In SUBSAFE, certification is based on what they call *Objective Quality Evidence* (OQE). OQE is a statement of fact, either quantitative or qualitative, pertaining to the quality of a product or service based on observations, measurements, or tests that *can be verified*. OQE provides evidence that deliberate steps were taken to comply with the SUBSAFE requirements. Because probabilistic risk assessment cannot be verified without operating a system for years, it cannot be used for certification in the SUBSAFE program [Leveson, 2012].

6.4 Role of Software in the Analysis

Software represents a radical departure from the components used by engineers in the past. Basically, software is so useful and so powerful because it is essentially "design abstracted from its physical realization" [Leveson, 2012], that is, it is a pure abstraction without any physical embodiment until it is executed on a computer. While the computer hardware on which the software is executed and other digital system components can fail, the software itself does not "fail" any more than blueprints fail. Software simply does what the programmer programmed it to do. To its credit, ARP 4761 does recognize that assigning probabilities of "failure" to software in general is not possible.

Instead, ARP 4761 uses the concept of a design assurance level (or DAL), which specifies the level of rigor that applicants must use in assuring the implementation of the requirements is correct. RTCA DO-178C and DO-254, which specify the acceptable assurance practices for airborne software and electronic hardware, respectively, represent industry consensus on best assurance practices and are recognized by the certification authorities through advisory circulars as "an acceptable means of compliance." While the process required by DO-178B/C provides confidence that the requirements used are correctly implemented in the code, it does not ensure that the requirements themselves are correct or safe. In addition, there is no scientific analysis or evaluation we know of that shows any significant correlation between rigor of implementation assurance and system safety.

In fact, software by itself is not safe or unsafe. It cannot catch on fire or explode and, in fact, as noted above, it cannot even "fail" in the same way as hardware. Its contribution to safety lies in its behavior within the larger system context. Safety is an emergent system property, not a component property. Software that is perfectly safe in one system has contributed to accidents when it is reused in a different system [Leveson, 2004; Leveson 1995]. An example is the Ariane 5 loss, where software that was safe in the Ariane 4 was not safe when reused in the Ariane 5, which had a steeper trajectory [Ariane 5].The problem is not random failure but the software violating or not enforcing the system safety constraints for that particular system design. Virtually every serious incident or accident involving software has stemmed from incorrect or incomplete requirements [Leveson, 2004; Leveson 2012], for example, the omission of a requirement like "valve A must never be opened before valve B."

Increasing the DAL does lessen the risk that a requirements *implementation* verification step will be missed or be incorrect but cannot compensate for incorrect or missing requirements. The verification activities of RTCA DO-178C/DO-254 start from a presumption that the requirements (created by the ARPs processes) are correct, consistent, and complete. The link, therefore, between RTCA DO-178C/DO-254 and safety is at best tenuous. While the DAL for software is a convenient fantasy with respect to its relationship to aircraft safety, it is a fantasy nevertheless.

The problem is not necessarily what is in DO-178C but what is omitted. If an analysis of the software functional requirements is not included in the safety assessment process, then a major source of accident causation is being omitted and the results cannot claim to be very useful or predictive. Obviously, functional software requirements are generated at some point in the aircraft system engineering process. The point is that they are not generated from and integrated into (are not a part of) the safety assessment process. STPA generates the safety requirements (including those for software) and includes both hardware and software in the analysis process by analyzing the behavior of the system as a whole without treating the software as somehow different or oversimplifying its role in accidents. The interactions between software components can be subtle and indirect. Section 6.1, for example, described potentially hazardous interactions between the Autobrake and the hydraulic control software identified by STPA in the WBS example that could not be identified by the failure analysis process specified in ARP 4761: nothing failed in the scenario and software implementation errors were not involved.

As just one example, Section 5 described a hazardous scenario identified by STPA related to BSCU Autobrake behavior. Analysis of the BSCU hydraulic controller identified potentially hazardous interactions between the Autobrake and hydraulic control software.

Another major aspect of software-related safety issues that are not included in ARP 4761 is the impact of the software design on human operators who are trying to manage automation. Many of the major recent aircraft accidents have involved human-automation interaction problems. For example, software design is contributing to pilot mode confusion and inconsistent software behavior is leading to pilots doing the wrong thing because they are confused about the state of the automation or the aircraft [Billings, 1996; Sarter and Woods, 1995; Sarter, Woods and Billings, 1997; Leveson 2012]. While all four of the causes of the American Airlines B-757 Flight 965 accident near Cali Columbia in 1995 identified in the accident report involved errors on the part of the flight crew, perhaps the most puzzling is the fourth cause, which was described as the flight crew's failure to revert to basic radio navigation at the time when the FMS-assisted navigation became confusing and demanded an excessive workload in a critical phase of the flight [Cali]. Surely the fault should lie at least partly (if not in total) with the design of software that was confusing and required too high a work load and not with the pilot who was confused by it.

6.5 Role of Humans (Operators) in the Analysis

While software is at least treated as part of the system in ARP 4761, human operators are placed outside the system boundaries and considered primarily only as mitigators of hazardous physical system failures. In several places in the WBS example, an assumption is made that the pilots will be able to detect and respond appropriately to physical component failures. That assumption is used to lower the classification of the hardware or software. For example, in Table 4.2, under the failure condition of an annunciated loss of deceleration capability, the crew is assumed to be able to steer the aircraft clear of any obstacles so the classification is "No safety effect."

STPA includes pilots in the system analysis in the same way as any other component. Software is often automating what human pilots used to do, so it is strange that as soon as the human activities become automated, those activities are included in the ARP 4761 safety assessment process but excluded when they were performed by humans. In essence, ARP 4761 assumes the best case for pilot behavior (i.e., they will always do the right thing to respond to a hardware failure as long as the failure is annunciated) while STPA considers the worst case.

In STPA, crew errors are integrated into the WBS hazard analysis and considered even when faults are annunciated. For example, in the STPA WBS analysis, instead of just assuming a probability of valve failure, one scenario identifies how the BSCU hydraulic controller design (and green shut-off valve) might affect crew behavior adversely. This scenario is then used to identify potential solutions such as controls on the BSCU design or crew requirements. As another example, Table 5.1 explicitly considers flight crew unsafe control actions. The causal analysis of CREW.1a1 shows a scenario caused by inadequate feedback regarding the Autobrake.

The human factors analysis in STPA is still fairly limited. STPA analysis needs to account for the unique processes by which humans make decisions. We are working on extending STPA in this respect. Some first steps have been described by Thornberry [Thornberry, 2014].

The most important limitation of treating humans separately from system design is that human-system interaction is, as noted in Section 6.4, omitted. But such interaction of the pilot with the aircraft is an important factor in many of the aircraft accidents that have occurred in the last decade or so. Although the pilot is invariably found to be the cause of these accidents, it is interesting that after the

accident, the software is usually changed. Human behavior is always influenced by the design and operation of the overall system context and can thus be influenced to make unsafe decisions.

Again, we understand that as with functional and software requirements, human factors considerations are included in aircraft design today in a separate engineering process. The problem is that, like the functional requirements generation process, treating human factors separately means that there is little chance to look at the design of the system, including the automation, in an integrated way and from a safety standpoint so that the safety of the integrated system is assured. An independent human factors analysis also makes no contribution to identifying potential software design problems that can contribute to pilot error. Trying to fix poor automation design choices through human-machine interface design or by training is very likely to lead to avoidable accidents.

Including humans, however, makes the argument for doing a quantitative safety assessment even more difficult than for software. Although it might be possible to argue for "human reliability (probabilistic) analysis" in relatively simple systems where the steps are specified and trained (but even that is debatable), the activities of pilots today require complex cognitive processing. Cognitive engineers and system theorists argue that all human behavior is affected by the context in which it occurs [e.g., Dekker, 2006]. If this belief is true, then assigning general probabilities and separating errors from the specific context in which they occur is not possible.

It is interesting to compare the basic exclusion of pilots from the ARP 4761 aircraft safety assessment with the process used for safety assessment of air traffic control (ATC) systems, such as NextGen. Because these new ATC systems are so human intensive (and will be for quite some time), humans (the pilots and the controllers) are included in the safety assessments. However, they are treated as probabilistic devices and the same ARP 4761 reliability/failure approach is used [Fleming, et. al., 2013]. For example, the safety assessment in ITP (a new procedure for allowing aircraft to pass each other while temporarily violating minimum separation requirements) treats humans as leaves on a fault tree with assigned probabilities [RTCA DO-312]. Many of the most important potential human errors and accident causes are omitted from these fault trees [Fleming, et.al., 2013], but the real problem is treating humans as random devices, particularly when they are engaged in the complex decision making required to manage the automation on aircraft or to manage the airspace today.

If, as is often claimed, pilots really do cause the majority of aircraft accidents, then it seems that leaving them out of the safety assessment will only provide unrealistic safety assessment and a false sense of confidence in the aircraft design.

6.6 Role of Operations in the Analysis

Some maintainability requirements are included in ARP 4761, but little other information provided by the probabilistic analysis is very useful in operations. More about this is written elsewhere [Leveson, 2012], but improving safety in operations requires knowing the assumptions about operational conditions and pilot behavior that were made in the safety assessment/hazard analysis in order to be able to ensure that those assumptions are not being violated. These assumptions, particularly assumptions about the behavior of humans in the system, can be checked in performance audits and included as part of the incident analysis process.

In addition, while the STPA example in this paper included only aircraft design and not operations (to more closely match the WBS example in ARP 4761), STPA can include operations in the hazard analysis process.

6.7 Process Comparisons

Both the ARP 4761 process and STPA are iterative, system engineering processes that can start in the concept formation stage and guide more detailed design. A major difference is that ARP 4761 safety assessment starts from failures and failure conditions while STPA starts from hazards. Failure scenarios are identified by STPA in Step 2 when the causes of unsafe control (hazards) are identified.

Both also provide a means to use safety considerations to impact design decisions as they are made rather than assessing them after the design process is completed. The difference is in the type of guidance provided. ARP 4761 provides guidance in implementing fail-safe design, with an emphasis on redundancy and monitors. STPA, because of the more general safety requirements generated, has the potential for suggesting more general safe design features, including eliminating hazards completely.

6.8 Cyber Security and Other System Properties

While cyber security is not included in the example STPA analysis shown in this report, we have defined an analysis process called STPA-Sec [Young and Leveson, 2014] that uses the same top-down system engineering process for cyber security as STPA does for safety. In fact much of the analysis is shared, with some additional considerations added to Step 2 in STPA-Sec.

Theoretically, STAMP and STPA are applicable to any emergent system property. Goerges, for example, showed how it can be used to identify causal factors for quality loss in complex system design [Goerges, 2013]. By starting from a more general (more inclusive) causality model, the opportunity arises to create new, more powerful techniques in system engineering and risk management.

6.9 Cost and ease of use

Without a lot of careful experimental evidence in realistic industrial environments, it is difficult to make claims about ease and cost of use. We note, however, that some comparison data is available where STPA required many fewer resources and effort than FTA and FMEA, while at the same time providing more comprehensive results. For example, Balgos used STPA on a medical device that had been involved in an accident and recalled by the FDA [e.g., Balgos, 2012].

Thomas has shown how to automate much of STPA Step 1 [Thomas, 2013] and how tools can be built to automatically generate model-based safety requirements. Tools to automate or support other parts of the STPA process are being developed [Adhulkhaleq and Wagner, 2014; Suo and Thomas, 2014; Hommes, 2014].

7 Conclusions

This report compares the safety analysis process of ARP 4761 with STPA, using the wheel brake system example in ARP 4761. We show that STPA identifies hazards omitted by the ARP 4761 process, particularly those associated with software, human factors and operations. The goal of STPA is to identify detailed scenarios leading to accidents so that they can be eliminated or controlled in the design rather than showing that reliability goals have been met. The succeeding verification processes (DO-178C/DO-254) are still necessary to assure that the requirements provided by the process in ARP 4754A and supported by STPA, are fully verified.

In the reality of increasing aircraft complexity and software control, the traditional safety assessment process described in ARP 4761 omits important causes of aircraft accidents. The general lesson to be learned from the comparison in this report is that we need to create and employ more powerful and

inclusive approaches to evaluating safety that include more types of causal factors and integrate software and human factors directly into the evaluation. STPA is one possibility, but the potential for additional approaches should be explored as well as improvements or extensions to STPA. There is no going back to the simpler, less automated designs of the past, and engineering will need to adopt new approaches to handle the changes that are occurring.

References

Asim Adhulkhaleq and Stefan Wagner, Tool Support for STPA, STAMP/STPA Workshop, March 2014, http://psas.scripts.mit.edu/home/2014-stamp-workshop-presentations/

Vincent Balgos, *A Systems Theoretic Application to Design for the Safety of Medical Diagnostic Devices*, S.M. Thesis, Engineering Systems Division, MIT, February 2012.

Greg Bartley and Barbara Lingberg, Certification concerns of integrated modular avionics (IMA) systems, *27th Digital Avionics Systems Conference*, 2011.

Charles E. Billings, *Aviation Automation: The Search for a Human-Centered Ap*proach. CRC Press, 1996.

Sidney Dekker, *The Field Guide to Understanding Human Error*, Ashgate Publishers, 2006.

Cody H. Fleming, Melissa Spenser, John Thomas, Nancy Leveson, and Chris Wilkinson, Safety assurance in NextGen and complex transportation Systems, *Safety Science*, Vol. 55(6), June 2013, p. 173-187

Cody H. Fleming and Nancy G. Leveson, Improving hazard analysis and certification of integrated modular avionics, *Journal of Aerospace Information Systems*, AIAA, Vol. 11(6): 397-411, June 2014.

Stephanie L. Goerges, *System Theoretic Approach for Determining Causal Factors of Quality Loss in Complex Systems*, S.M. Thesis, Engineering Systems Division, MIT, February 2013.

Qi Hommes, The Volpe STPA Tool, STAMP/STPA Workshop, March 2014, http://psas.scripts.mit.edu/home/2014-stamp-workshop-presentations/

Nancy G. Leveson, *Safeware: System Safety and Computers,* Addison-Wesley, 1995.

Nancy G. Leveson, Role of software in spacecraft accidents, *AIAA Journal of Spacecraft and Rockets*, Vol 41(4): 564-575, 2004.

Nancy G. Leveson, *Engineering a Safer World,* MIT Press, 2012.

Nancy G. Leveson, STPA Primer, 2013, http://sunnyday.mit.edu/STPA-Primer-v0.pdf

Matthew Seth Placke, *Application of STPA to the Integration of Multiple Control Systems: A Case Study and New Approach*, S.M. Thesis, Engineering Systems Division, MIT, June 2014.

RTCA, DO-312: Safety, Performance and Interoperability Requirements Document for the In-Trail Procedure n the Oceanic Airspace (ATSA-ITP) Application, DO-312, Washington D.C., June 19, 2008.

RTCA, DO-254/ED-80: Design Assurance for Airborne Electronic Hardware, 2000.

RTCA, DO-178C/ED-12C: Software Considerations n Airborne Systems and Equipment Certification, 2012

SAE, ARP 4761: Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment, ARP 4761, Dec. 1996

SAE, ARP 4754A: Guidelines for Development of Civil Aircraft and Systems, 2010.

Nadine Sarter and David D. Woods, How in the world did I ever get into that mode? Mode error and awareness in supervisory control, *Human Factors*, 37(1): 5–19, November 1995

Nadine Sarter, David D. Woods, and Charles E. Billings, Automation surprises. In *Handbook of Human Factors and Ergonomics*, 2nd Ed., ed. G. Salvendy, Wiley, pp. 1926-1943, 1997.

Dajiang Suo and John Thomas, An STPA Tool, STAMP/STPA Workshop, March 2014, http://psas.scripts.mit.edu/home/2014-stamp-workshop-presentations/

Dajiang Suo, personal communication, June 23, 2014 (draft report on incidents from NHTSA database in preparation).

John Thomas, *Extending and Automating a Systems-Theoretic Hazard Analysis for Requirements Generation and Analysis*, Ph.D. Dissertation, Engineering Systems Division, MIT, June 2013.

Cameron L. Thornberry, Extending the Human-Controller Methodology in Systems-Theoretic Process Analysis (STPA), S.M. Thesis, Aeronautics and Astronautics, MIT, June 2014.

William Young and Nancy G. Leveson, An integrated approach to safety and security based on systems theory, Communications of the ACM, Vol. 57(2): 31-35, February 2014.

Accident Reports

[Ariane 5] J.L. Lions, Report by the Inquiry Board, Flight 501 Failure, 1996.

[Cali] Aeronautica Civil of the Republic of Columbia, AA965 Cali Accident Report, September 1996.

[Warsaw] Main Commission Aircraft Accident Investigation Warsaw, "Report on the Accident to Airbus A320-211 Aircraft in Warsaw on 14 September 1993,

[T-1A] AIB, Aircraft Accident Investigation T-1A, S/N 93-0633, Laughlin AFB, Texas 21 May 2008

[Ibiza] (Civil Aviation Accident and Incident Investigation Commission (CIAIAC), Spain, Failure of Braking Accident on 21 May 1998 at Ibiza Airport, Balearic Islands, A-19/98, Spain

[Cardiff] AAIB Bulletin No: 2/2005 - Accident Report on A320-200 Flight C-FTDF at Cardiff International Airport on August 3, 2003," Air Accident Investigation Branch, EW/C2003/08/11] [http://www.aaib.gov.uk/cms_resources.cfm?file=/C-FTDF.pdf

Appendix: WBS Design Assumptions used in the STPA Analysis

STPA generates functional safety requirements. At some point, an evaluation must be made of the implementation of those requirements in the hardware design. In addition, the STPA analysis can be iterated down to a low level of detailed design. The FHA, PSSA, and SSA in ARP 4761 only considers component failures so the design description in the ARP (Section 2 of this report) is adequate. To do a broader type of analysis and verification, we needed to augment the physical design shown in Figure 2.2.

Figure A1 shows an augmented physical diagram of the Wheel Brake System. It is based on the architecture described in ARP 4761 with some additions to clarify the system behavior. The logic equations describing the operation are shown at the bottom of the figure.

During normal operation, the WBS uses hydraulic pressure from the green pump to engage the brakes. The green shut-off valve is normally open and the green side of the selector valve is kept open when there is hydraulic pressure on the green side of the selector. The green meter valve can be adjusted to a variable position by the BSCU (not just open/close) to achieve the desired hydraulic pressure at PR1. The meter valve uses hydraulic supply from the green pump to increase brake pressure at PR1 (e.g. when the meter valve is opened), and uses a return line to release brake pressure (e.g. when the meter valve is closed). A piston at PR1 uses the hydraulic pressure to reduce wheel speed. The BSCU controls the meter valve based on manual braking commands from the flight crew or from automated braking commands generated within the BSCU for Autobrake. The BSCU also monitors wheel speeds to detect skidding, in which case the BSCU will pulse the green meter valve to prevent the skid.

Whenever the green line pressure at the selector valve decreases below a threshold, the selector blocks the green line and opens the blue line. Otherwise the green line is kept open and the blue line is blocked. When the green line is open the system is in normal braking mode, and when the blue line is open the system is in alternate braking mode. Note that although this functionality is represented in the diagram by a spring-loaded selector valve for simplicity, any implementation with the same functionality could be used.

Note that the BSCU has no way to detect the state of the selector valve. Whenever the BSCU detects skidding, it pulses both the green line meter valve and the blue line anti-skid valve simultaneously. In this way, anti-skid functionality is available in either mode. However, the auto-braking functionality is only possible in normal braking mode because the blue anti-skid valve is not continuously variable. Instead, in alternate braking mode the blue meter valve controls braking force based on mechanical inputs directly from the brake pedals. Normal mode and alternate mode each use a separate set of pistons at the wheel connected to PR1 and PR2 respectively.

The accumulator on the blue line is a passive device that helps dampen pressure surges by holding a limited amount of air that is compressed by hydraulic pressure. It can also maintain a limited amount of hydraulic pressure in the event of certain failures in the blue system. If the blue hydraulic system is operating normally and the wheel braking system is in alternate braking mode, the accumulator is pressurized by hydraulic pressure. If hydraulic pressure above the non-return valve becomes less than the pressure below the non-return valve, for example due to a hydraulic leak or pump failure, then the non-return valve closes. The accumulator can then passively maintain the hydraulic pressure to the blue valves and therefore to PR2 for a small number of brake applications.

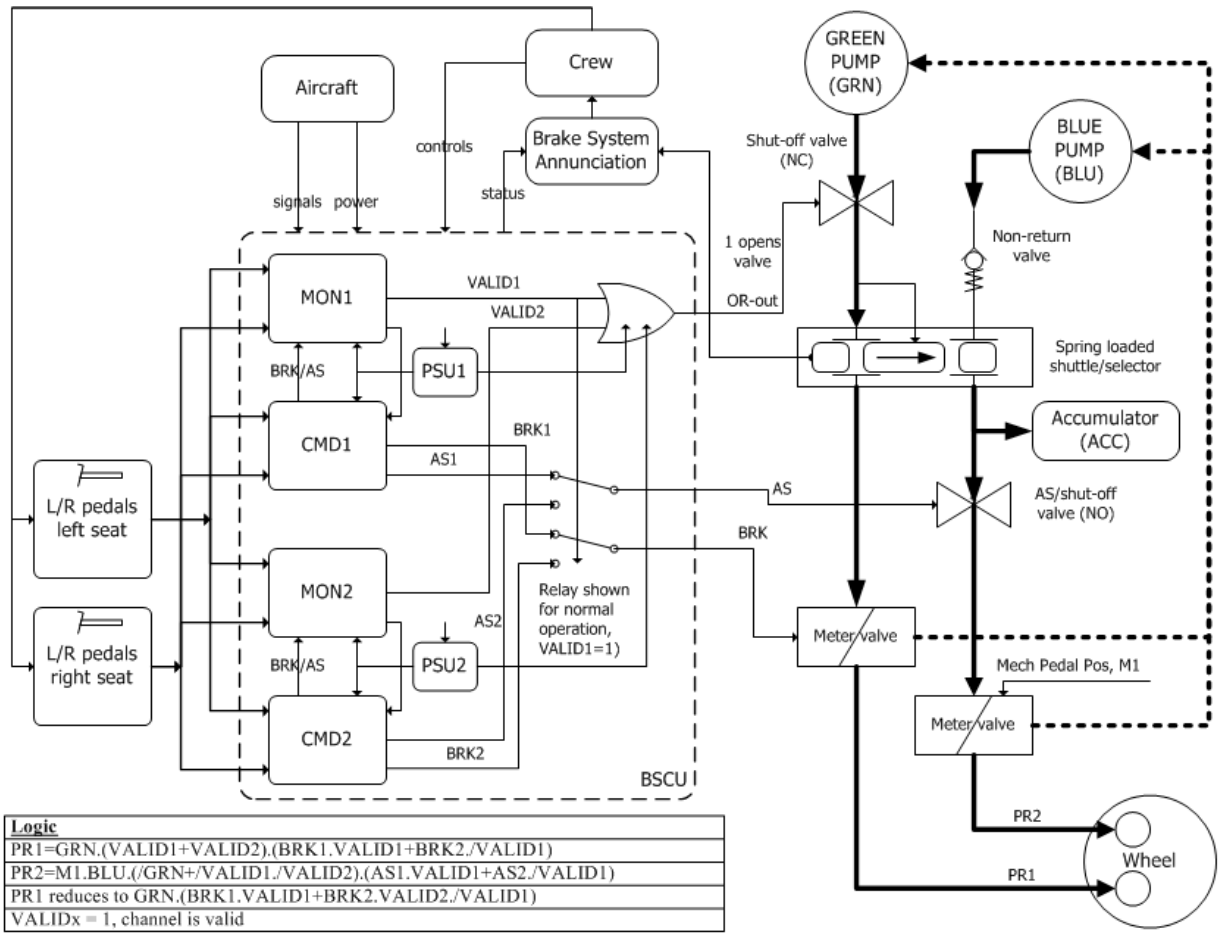

Figure A1: Wheel brake system diagram

The BSCU may force the spring-loaded selector to switch to the blue system by closing the green shutoff valve thus blocking the green hydraulic supply from the pump to the spring-loaded selector. The flight crew can manually command the BSCU to command the switchover. Otherwise, BSCU will automatically command the switchover when it detects an internal BSCU fault or a power supply fault.

The BSCU actuates all valves by providing power or not providing power (there is no separate power supply for the valves). In the event of a loss of power to the BSCU, no power is provided to BSCU outputs. The outputs are designed so that a loss of power brings the system to a known state: a fault is indicated to the brake system annunciation, the green shut-off valve is closed, the green meter valve is closed, and the blue anti-skid valve is open.

The BSCU is comprised of CMD and MON functionality. Both functions are duplicated twice for redundancy. CMD receives pilot auto-brake commands, pilot manual braking commands, and wheel speeds. The manual braking commands are received from two pedal position sensors, one per pilot (differential braking is not analyzed in the ARP 4761 example). CMD provides two outputs: a variable braking command that may include anti-skid pulses (to control the green meter valve) as well as a separate dedicated command for anti-skid pulses only (to control the blue anti-skid valve). Both commands are always output simultaneously.

MON produces a single output indicating whether the CMD output is valid. MON receives the same inputs as CMD, implements the same algorithm, and compares the result with the commands produced by CMD. If there is a mismatch, MON will flag the CMD output as invalid. In addition, MON monitors its own power supply and flags the output as invalid if the voltage fluctuates outside specified parameters but is still sufficient for MON operation. In either case, the MON output indicates that there is a fault. CMD receives the MON output, and CMD disables its own outputs (same outputs as when power is lost) for as long as MON reports a fault condition. Once MON flags CMD output as invalid, MON latches the flag and it cannot return to valid unless the BSCU is powered off. CMD and MON are both powered by the same power supply. If power to MON is lost, the MON output is designed such that it will indicate the channel is invalid.

The BSCU contains two redundant and identical channels with CMD and MON functionality. The first channel consists of MON1 and CMD1, and the second channel consists of MON2 and CMD2. Each channel implements identical functionality.

The BSCU also implements an overall validity monitor to determine the behavior when one or both channels are flagged as invalid. When both channels are flagged as invalid, the green shut-off valve is closed to allow the selector valve to switch to alternate braking mode using the blue hydraulic system. If at least one channel is valid, the shut-off valve remains open. Whenever channel 1 is valid, CMD1 outputs are forwarded to the valves and CMD2 outputs are not. When channel 1 is flagged as invalid, CMD2 outputs are forwarded to the valves.

The BSCU receives a single source of electrical power from the aircraft. The BSCU contains two internal power supplies that convert external power into the necessary voltage for each channel. MON1 and CMD1 are powered by one power supply, while MON2 and CMD2 are powered by the other. When the relay receives power from MON1 indicating that CMD1 has valid output, the switch connects CMD1 outputs to the BSCU outputs. When the switch does not receive power from MON1, it passively defaults to connecting CMD2 outputs to BSCU outputs. The BSCU overall validity monitor, represented by an OR gate in Figure A1, is powered directly by the BSCU's external power source and not by either internal power supply. When the external aircraft power to the BSCU is lost, the BSCU outputs return to a known default valve. That is, the output to brake system annunciation indicates a fault, the output to the green shutoff valve commands valve closure, and the output to the green meter valve commands the valve closed, and the output to the blue shutoff valve commands the valve open.

In addition to accepting manual braking commands from the pedals, the BSCU also includes automation that can provide Autobrake functionality. The flight crew commands related to Autobrake include arm/set deceleration rate and disarm. During normal operation, the crew may arm the system and set the deceleration rate prior to takeoff or landing. Arm and set commands are provided with a single action, for example by pushing a button corresponding to low, medium, or high deceleration. Once armed, the BSCU can automatically activate and begin wheel braking upon touchdown to achieve the programmed deceleration rate. The touchdown input to the BSCU indicates when a touchdown has occurred, and the exact algorithm used to detect a touchdown is outside the scope of the example in ARP 4761. The BSCU can also automatically activate for a rejected takeoff, indicated by a takeoff input to the BSCU. The exact algorithm used to detect a rejected takeoff is also outside the scope of the example in ARP 4761.

The crew can disarm Autobrake at any time. If the disarm command is provided when Autobrake is armed but not actively braking, the system is disarmed. If the disarm command is provided after Autobrake has been triggered and is actively braking, Autobrake will immediately stop braking and become disarmed. During rollout deceleration, manual depression of the brake pedals will also disable

Autobrake and transfer braking control back to the crew. The crew is always notified of the Autobrake armed status, the programmed deceleration rate, and whether or not Autobrake is actively braking.

Because Autobrake functionality is achieved using the green meter valve, it is only available when the braking system is in normal braking mode. Anti-skid functionality is provided in both normal and alternate braking modes unless the crew powers off the BSCU, both BSCU channels are in a faulted mode, or the ground speed is less than 2 meters per second.

The significant assumptions made about the WBS are:

1. MON1, MON2, CMD1, and CMD2 all receive the same inputs and perform the same calculations. Comparisons are made to determine the health status, setting the VALID1/VALID2 signals false if a failure is detected
2. ARP 4761 states that the BSCU disables its outputs in the event that VALID1/VALID2 are both false, however this does not seem to be implemented in the proposed relay architecture. It is assumed CMD1 and CMD2 are designed to disable their own outputs when VALID1 and VALID2 become false, and therefore CMD1 and CMD2 must receive the VALID1 and VALID2 signals (respectively).
3. The selector valve functions equivalently to a spring loaded selector valve that will automatically open the blue line if green pressure fails or is shut off by the BSCU
4. Feedback to the crew is provided based on the selector valve position, which indicates alternate braking mode or normal braking mode
5. Each channel of the BSCU contains a separate internal power supply. Both BSCU power supplies are fed separate external sources of aircraft power
6. The OR gate in the drawing remains powered, and therefore functional, if either BSCU internal power supply is operational
7. For total loss of power to the WBS or failure of both BSCU power supplies, the system is designed such that the OR gate output will indicate a known default value (i.e. close the valve).
8. The valves are operated only by the BSCU and are powered by the BSCU signals. If the BSCU loses all power, the valves remain in their mechanical default position.
9. The relay is spring loaded to default to channel 2 unless actively held to engage channel 1
10. ARP 4761 states that the pilots are able to manually command alternate mode, but the proposed OR logic does not seem to implement this functionality. Therefore it is assumed that the way the pilots command alternate mode is by turning off BSCU power. This causes the valves to mechanically default to alternate mode.