Failsafe Mechanism Design for Autonomous Aerial Refueling using State Tree Structure

Ke Dong[1]

Abstract

Autonomous Aerial Refueling (AAR) is vulnerable to various failures and involves co-operation between autonomous receivers, tankers and pilots. Wrong or empty flight maneuvers may be executed when unexpected failures or command conflicts happen. To solve this problem, a failsafe mechanism based on State Tree Structure (STS) is proposed. The failsafe mechanism is a control logic that guides what subsequent actions the autonomous receiver should take, by observing real-time information of internal low-level subsystems such as guidance and drogue&probe and external instructions from tankers and pilots. To generate such a controller using STS, the whole AAR procedure is decomposed into five modes plus six sub-modes. Common safety issues related with five low-level subsystems are summarized and nine safety requirements are put forward to restrict the receiver's behaviors. On this basis, the AAR plants and specifications are modeled by STS. Then a supervisor with 15 binary decision diagrams is synthesized to cover the whole AAR model with 2.5×10^{12} states within 2 seconds. The design procedures presented in this paper can be used in decision-making strategies for similar flight tasks. Supporting materials can be downloaded in Github[1] , including related software, input documents and output files.

Keywords: Autonomous aerial refueling, failsafe mechanism, state tree structure

Email address: dongk@buaa.edu.cn (Ke Dong)

[1] https://github.com/KevinDong0810/Failsafe-Design-for-AAR-using-STS/

1. Introduction

The Autonomous Aerial Refueling (AAR) is an effective approach to increase the range and endurance of Unmanned Aerial Vehicle(UAV) by refueling them in air. Since the first successful AAR concept demonstration by Boeing company, several ambitious AAR projects have been carried on by the NASA[1], US Air Force and US Navy [2], which prove the feasibility and reliability of AAR. During an AAR operation shown in Figure.1, a UAV (or a receiver) detaches from its formation and approaches the rear of a tanker for refueling. The boom system or Drogu &Probe system would be used for fuel transfer. Once refueled and cleared, the receiver disengages from the tanker and rejoins the formation.

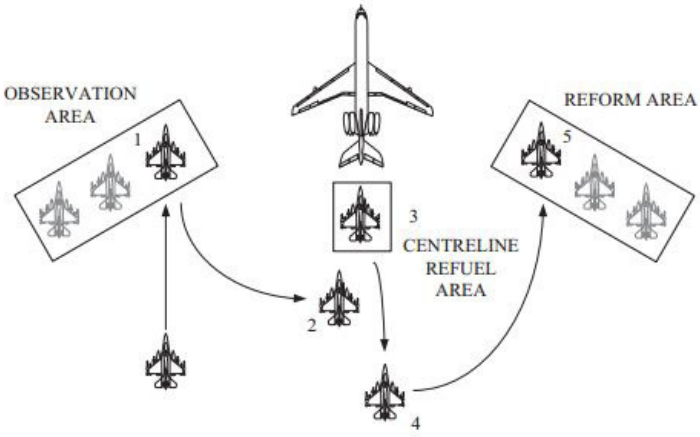


Figure 1: Typical AAR refueling procedures

The AAR is a semi-automated process, where the control of unmanned receivers is frequently disturbed by on-board receiver controller, external tanker controller and remote human pilots. Besides, AAR process is vulnerable to various system failures like probe damage and unsteady airflow. Therefore, a high-level logic controller that coordinates all these control components and produces the desired safe functionality with the presence of faults and failures is needed, namely the *Failsafe Mechanism*. A wealth of researches have been done on this domain. [3] formalizes the general framework for formal verification of human-

automation interface, where both automaton and human controller share the authority over the system. [4, 5] investigate the safe path planning in AAR to avoid collision with consideration of receiver-pilot interaction. In the Research Flight Control System develop by NASA in [1], human-machine interface is put into practice. But none of them has taken system failures into consideration for logic design, under which dangerous maneuver or empty actions may happen. In contrast, fault-related literature mainly focus on low-level resource controllers, such as fault detection techniques [6] and fault-tolerant control algorithms[7, 8]. No high-level logic controller is considered.

At present, failsafe mechanism design for AAR is seldom investigated. Related application scenarios like rotorcraft mainly use engineering experiences and manual computation to design, like DJI autopilot and ArduPilot. Man-made mistakes, logical bugs or an incomplete treatment are unavoidable in this approach and it's time-consuming for large-scale systems. Therefore, This paper aims to use a model-based method, namely Supervisory Control Theory (SCT) of State Tree Structure (STS), to design a comprehensive failsafe mechanism.

SCT is a formal method for synthesizing supervisors that on observing events strings generated by the plant, at each state would disable a suitable subset of controllable events to satisfy the given control specifications. First put forward by Ramage and Wonham thirty years ago[9], the SCT and STS have a solid theoretical foundation [10, 11, 12], and numerous industrial applications such as testbed assembly process [13], telephone directory assistance call center [14] and magnetic resonance image (MRI) scanner [15]. Compared with the current failsafe mechanisms based on engineering experience, the SCT approach has two main advantages: 1) *Correct by design*: the supervisor will only enable those controllable events which will make the plant's behavior always satisfy the control specifications. 2) *Evolvability*: the plant and specifications are composed of AND superstates and OR superstates, which allows the modular modeling method and promises the evolvability of the whole system. Different specifications can be added when taking different subsystems and control hierarchies into consideration.

The main contributions of this paper are given as follows:

- This paper first investigates the failsafe logic design of autonomous receivers using a solid formal verification theory. Trustable supervisor is synthesized to cover common system failures and interaction between receivers, tankers and pilots. This design framework can be easily adopted for different aerial refueling procedures and other similar control tasks.
- A chain of tool for synthesizing, implementing and simulating supervisors gained through STS for aerial refueling has been developed and posted in Github. This will accelerate similar failsafe mechanism design in related domains.

The remainder of this paper is organized as follows. Section 2 introduces the preliminary knowledge of STS and SCT. In Section 3, AAR procedures are decomposed, common safety issues are summarized and requirements are put forward. On this basis, Section 4 builds the model of AAR using the mathematical tool of STS and synthesizes the supervisor. Finally, conclusion is made in Section 5.

2. Preliminary of STS

In this section we give the basics of STS theory. Details can be found in textbooks [10, 11, 12]. An example of Simple AAR modeled with STS is illustrated for better understanding.

2.1. State Tree Structure

STS is the extension of the automaton in SCT which introduces natural hierarchical structure into the system model. It mainly contains two parts: *state tree* and *holon*. State tree, say **ST**, organizes the state space of an STS. The nodes on a state tree are called states. Every state tree has a unique root state. A state on a state tree is called an OR (resp. AND) *superstate* if it can be represented as the disjoint union $\dot{U}$ (resp. Cartesian product X) of its

children. Each child state is called an OR (resp. AND) component. The lowest level states are called *simple states*, which are required to be OR components to reduce redundant information.

In most application scenarios, OR components work sequentially while AND components work simultaneously. The state tree of Simple AAR $\mathbf{ST}_{simAAR}$ is shown in Figure2 as an example. **Receiver** is a AND superstate and its state is up to the state of **Autopilot** and **Guidance**. **Autopilot** is an OR superstate and would be at **Standby**, **Rendezvous**, or **Refueling** state.

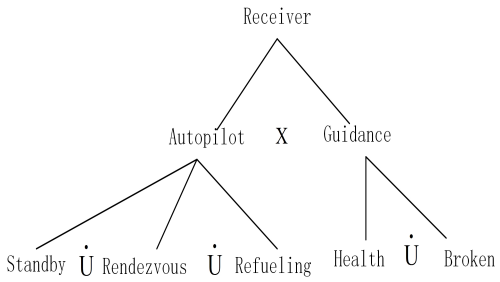


Figure 2: The STS model of Simple AAR

The state transition of STS is organized as a *holon*, namely an automation with hierarchy. It depicts the boundary and local state transitions of state tree. A holon $\mathbf{H}$ is a 5-tuple $\mathbf{H} := \{X, \Sigma, \delta, X_o, X_m\}$, where X is a finite set consisting of states in $\mathbf{ST}$; Σ is the finite event set (also called an *alphabet*); δ is the (partial) transition function on state set: $\delta : X \times \Sigma \to X$; $X_o \subseteq X$ is the subset of *initial states*; $X_m \subseteq X$ is the subset of *marker states* that the whole system desires to reach.

In SCT, the alphabet Σ is partitioned as $\Sigma = \Sigma_c \cup \Sigma_u$ where Σ_c is the subset of *controllable events* that can be started or stopped at will, while Σ_u is the subset of *uncontrollable events* that cannot do so. In engineering practice, Σ_c represents relevant discrete commands to the low-level resource control, while Σ_u represents messages sent from resource control to the supervisor, like failure notification and sensor event.

The holon of $\mathbf{ST}_{simAAR}$ is illustrated in Figure.3. In this holon, Σ_c =

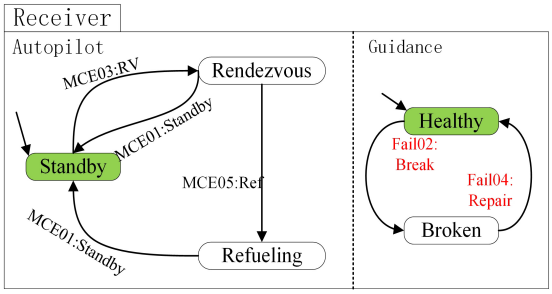


Figure 3: The holon for a simplified AAR

$\{MCE01, MCE03, MCE05\}$[2], $\Sigma_u = \{Fail02, Fail04\}$, $X_o = X_m = \{$**Standby**, **Healthy**$\}$. In the normal work cycle, **Autopilot** will go through **Standby** $\rightarrow$ **Rendezvous** $\rightarrow$ **Refueling** and finally return to **Standby** state. If *Fail02* happens, **Guidance** would switch to **Broken**, and the **Autopilot** can return to **Standby** from **Rendezvous**.

2.2. Plant, Specification and Supervisor

Supervisory Control Theory aims to synthesize the supremal non-blocking language (event sequences) of *Plants*, while satisfying the requirements of *Specifications*. The generated language is called *Supervisor*, and their detailed descriptions are given as follows:

- **Plant**. Plants are models of uncontrolled behaviors of physical systems and processes, which usually contain all the possible states and transitions. For example, $\mathbf{ST}_{simAAR}$ and the corresponding holon are the plant for Simple AAR.

- **Specification**. Specifications are models of control requirements. It can be given as illegal state sets in form of logical expression [16], or illegal event sequence in AND superstates [17], which is mainly used in this paper.

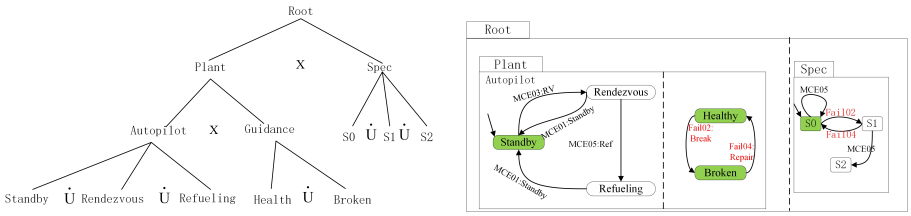
[2]Usually, controllable events are assigned odd numbers while uncontrollable events are assigned even numbers

- **Supervisor**. The Supervisor is the minimally restrictive event sequence of plants under the restriction of specification, and is implemented by state feedback control functions for events. It can be seen as a dictionary of controllable events of at each state of the plant, from which an associated controller can choose an appropriate control action.

In the generation algorithms for supervisor[11], all event sequences leading to blocked states would be deleted, namely states that cannot reach to marker states by any event transitions. Therefore, to model the specification with AND superstates, we just need to make the undesired event sequence lead to *blocked states*.

Now we add the safety requirement that *when the Guidance breaks down, the Autopilot should not allow the system to proceed Refueling* into the Simple AAR. The corresponding AND superstate can be seen in Figure.4. The undesired event sequence is (*Fail02*, *MCE05*) and this will lead **Spec** into blocked state **S2**. Therefore, when **Guidance** is at **Broken**, transition from **Rendezvous** to **Refueling** through event *MCE05* should be forbidden.

With plant and specification in hand, STSLib developed by [18] is used to compute the Simple AAR. The original supervisor is given in the form of Binary Decision Diagram (BDD). To make it more readable to engineers without knowledge of BDD, we develop a wrapper function using MATLAB to transform the BDD into the look-up table, namely *control data* table.



(a) (b)

Figure 4: STS of Simple AAR, (a) State Tree, (b) Holon

As can be seen in Table.1, *MCE05* is allowed to happen when **Autopilot**

is at state **Standby**, **Rendezvous** or **Refueling** && **Guidance** is at state **Healthy**, which satisfies our safety requirement.

Table 1: The control data table for Simple AAR

Events	Autopilot	Guidance
MCE05	**Standby**, **Rendezvous**, **Refueling**	**Healthy**

3. AAR Mode Decomposition and Safety Issues

In this section, we decompose the main AAR work cycle and summarize the common safety issues for preparation of utilization of STS. We are interested in the high-level logic controller of autonomous receivers in AAR. It is supposed to select out appropriate control actions at each state while dealing with information from receivers, tankers and pilots to satisfy the safety requirements.

Note that there are two types of aerial refueling systems, namely the boom system and the Drogue&Probe system. According to the researches and engineering practice of [1, 19], this paper focus on Drogue&Probe system.

Figure 5: The Drogue & Probe system

3.1. Mode and Event Decomposition

Based on the standard Air-to-air refueling procedures for manned airplanes proposed by NATO [20] and the demonstration of NASA [1], the whole AAR work cycle is decomposed into following distinct modes:

a) Standby Mode: The receiver and the tanker keep their own bearings and velocity, waiting for next instructions. A safe distance between them is needed, which is usually vertical 1000ft and horizontal 1 nautical meter.

b) Rendezvous(RV) Mode: The receiver maneuvers from the safe position in Standby Mode to the *Observation Position* of the Tanker, which is usually at the left-hand side of tankers. Three sub-modes in low level are divided.

1. **Rendezvous Initial Point Sub-mode (RV_IP)**: Both the receiver and the tanker moves to the pre-defined geometry point, namely the Initial Point. In this mode preparations such as altimeter setting and datalink setting are made.
2. **Rendezvous Climb Sub-mode (RV_Climb)**: The receiver makes a progressive climb towards the tanker.
3. **Rendezvous Wait Sub-mode (RV_Wait)**:The receiver has to stay in the observation position and waits for the connect clearance from the tanker, since there could be plenty of UAVs waiting for refueling. When clearance is issued, the receiver can fly to the *Astern Position* and the autopilot would enter the Refueling mode. The astern position is the stabilized formation position behind the drogue with zero rate of closure.

c) Refueling(Ref) Mode: The receiver maneuvers carefully to engage its probe with the drogue of the tanker. Three sub-modes in low level are divided.

1. **Refueling Capture Sub-mode (Ref_Cap)**: The receiver attempts to connect with the tanker. If this capture attempt fails, it may retreat to the astern position and begin next trial, or maneuver according to pilot's instructions. If it successes, next sub-mode will be entered.

2. **Refueling Hold Sub-mode (Ref_Hold)**: The receiver and the tanker keep relatively stationary to transfer the fuel. When finished, next sub-Mode would be entered. Otherwise, the receiver should retreat to the initial state of Ref_Cap.

3. **Refueling Wait Sub-mode (Ref_Wait)**: Similar to the RV_Wait sub-mode, the receiver has to wait for the disconnection clearance from the tanker, to avoid possible hazardous influence on other UAVs still connected to the Tanker.

d) Reform Mode: The receiver disconnects with the tanker, maneuvers to the *Reform Area* and then leaves in formation. To form a closed-loop system, this paper assumes that the receiver enter the Standby Mode again.

e) Return to Land Mode (RTL): Owing to subsystem failures or pilot's instructions, the receiver is not able to carry on AAR task any more, but can return to the nearest airbase for a landing.

e) Emergency Landing Mode (EL): The receiver is not able to return to the nearest airbase and should make an emergency landing instead.

3.2. Safety Issues

The common failures of AAR are mainly related to the four major subsystems: Propulsion system, Guidance system, Drogue&Probe system and Datalink system. Their main safety issues are summarized as follows:

Propulsion System:

- **Fuel shortage for AAR**: The fuel is enough for the receiver to return to the airbase but not enough for it to continue the AAR task, when taking the possible failure of fuel transfer into consideration.

- **Fuel shortage for RTL**: The fuel is not enough for the receiver to return to the airbase.

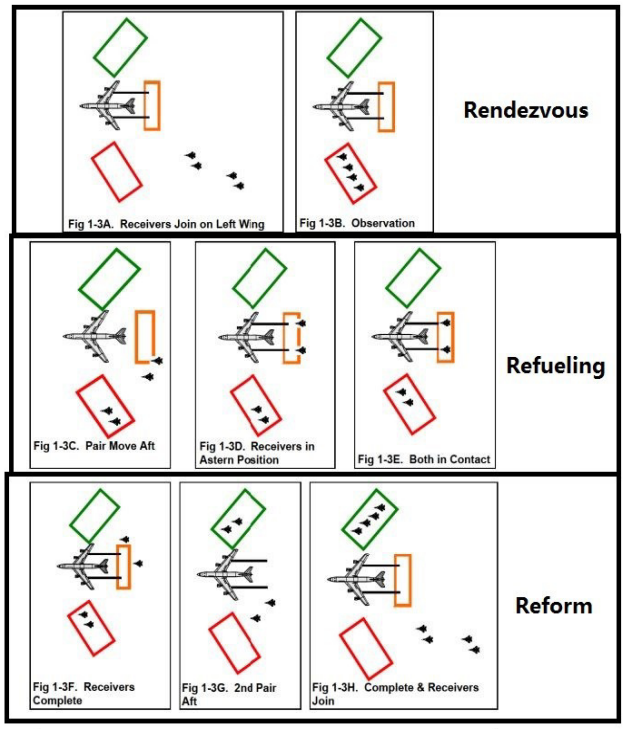


Figure 6: Illustrative Diagram for AAR[20]

- **Engine Breakdown**: The engine could break down owing to compressor surge, blade missing or accidental inhalation of foreign objects such as bird and Drogue&Probe wreckages after a failed AAR [20].

Guidance System:

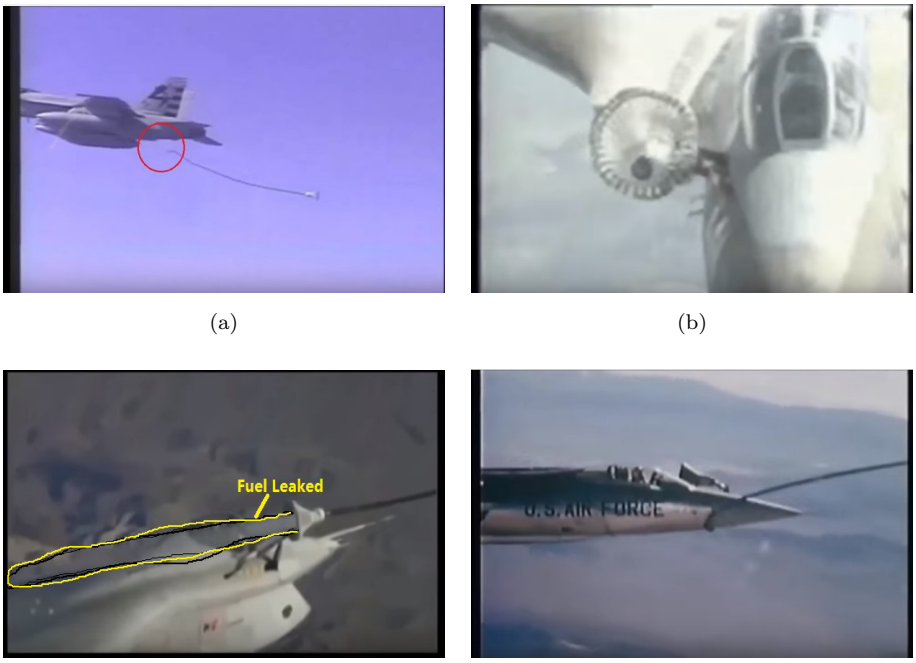
- **Guidance Suspension**: Owing to external conditions such as weak signals, unsteady airflow and poor visibility, the guidance system is not able collect enough data for position and attitude control. The conditions may be improved and then the guidance system can return to the normal state. but this transition is uncontrollable.

- **Guidance Equipment Breakdown**: Owing to the internal failures such as excessive temperature and shock, the guidance system loses the ability to collect data. It is impossible for the receiver to return to healthy state

in air.

Drogue&Probe:

- **AAR Equipment Damage**: Owing to the airflow turbulence, vibration of the receiver or tanker, etc., Drogue and Probe can be damaged seriously or even fall off during the Ref_Cap and Ref_Hold Mode.
- **Fuel Leakage**: Owing to the unstable contact or potential defects of Probe&Drogue, fuel may leaks outside and affects the vision of the receiver.
- **Hose Vibration or Entanglement**: The hose may be vibrating or entangled for the unstable airflow and unstable relative position of airplanes.



(a) (b) (c) (d)

Figure 7: Drogue&Probe Failures: (a) Hose Dropping, (b) Drogue Split (c) Fuel Leakage (d) Hose Vibration

Datalink:

- **Datalink Interruption**: Owing to the external conditions such as weak radio signals and hostile EMI attacks, the datalink system is not able to exchange data between the tanker and receiver temporarily. The conditions may be improved and then the guidance system can return to the normal state. but this transition is uncontrollable.

- **Datalink Equipment Breakdown**: Like the Guidance Equipment Breakdown, the datalink equipment could break down with no possibility of recovering before landing.

3.3. Safety Requirement

Mode transitions and safety issues summarized above contain all the behaviors we are interested in and serve as the plants of SCT. To restrict those undesired behaviors, e.g. proceeding to Refueling mode even the fuel is not enough, we now introduce the safety requirements, or specifications.

In the following paragraph, *healthy* or *function well* indicates that the subsystem cannot be out of service temporarily, while *intact* means that the system can be out of service temporarily.

SRs in Standby Mode:

- **SR1**: If the fuel is adequate for AAR and engine, guidance system, drogue&probe equipment and datalink system are all healthy, then the receiver should automatically switch to RV_IP mode. Otherwise such transition is forbidden.

- **SR2**: The transition from other modes to Standby mode or staying in Standby mode requires that the fuel is adequate for AAR, airplane engine is healthy, and guidance system, drogue&probe equipment and datalink system are all intact. Otherwise, the transition to RTL mode should be made instead.

SRs in RTL and EL Mode:

- **SR3**: The transition from other modes to RTL mode and staying in RTL mode requires that the fuel is adequate for RTL, engine is healthy and guidance system is intact. Otherwise, the transition to EL mode should be made.

SRs in Rendezvous Mode:

- **SR4**: If the fuel is adequate for AAR, engine and guidance system are healthy and datalink system is intact, the Rendezvous Climb Attempt is allowed to be made. And if all the requirements mentioned above are satisfied except that guidance system is out of service temporarily, the receiver should wait at the initial state of RV_Climb mode.

- **SR5**: When the receiver are waiting for clearance in the RV_Wait mode, if it does not satisfy the requirements that the fuel is adequate for AAR, engine is healthy and guidance system and datalink system are intact, the transition to Standby mode, RTL mode or EL mode should be made accordingly.

SRs in Refueling Mode:

- **SR6**: The transition from other modes to RTL mode and staying in RTL mode requires that the fuel is adequate for RTL, engine is healthy and guidance system is intact. Otherwise, the transition to EL mode should be made.

- **SR7**: In EL mode, the request for transition to Standby Mode should not be responded by the receiver, but the request for RTL Mode should be responded if the health condition allows so.

SRs in all modes:

- **SR8**:If other systems not listed above such as the electrical power supply system and hydraulic system breakdown, the receivers should be switched to Standby mode, waiting for the pilot's instruction. Similarly, if the other

system is out of service temporarily, the receiver is allowed to wait in its current state.

- **SR9**:The pilot can manually switch the receiver to the Standby mode, RTL mode or EL mode.

4. STS model of Aerial Refueling

Based on mode decomposition, safety issues and requirements presented in Section 3, this section builds the whole STS model of AAR, whose rough diagram is shown in Figure.8. We use italic labels to represent events, while bold labels for states. And all the events in AAR are summarized into four categories: Manual Input Events (MIEs), Mode Control Events (MCEs), Automatic Trigger Events (ATEs) and Subsystem Failure Events (SFEs). MIEs and MCEs are controllable events, while ATEs and SFEs are uncontrollable events. Their descriptions are shown in Table.2. Meanwhile, in our diagrams of holons, initial states are denoted by an unconnected incoming arrow, and marker states are denoted by filled vertices.

Table 2: Descriptions of the four event types

Name	Description
MIEs	Instructions from the pilots to change the automatic AAR procedures
MCEs	Commands from the autopilot to proceed AAR procedures automatically
SFEs	Failures of subsystems
ATEs	Uncontrollable events except for SFEs

4.1. Plant

A.Autopilot

Autopilot is the AND component of **Receiver**, and it is an OR superstate with three simple states: (**Standby**, **RTL** and **EL**) and two OR superstates (**Rendezvous** and **Refueling**). The inner transitions of **Autopilot** are shown

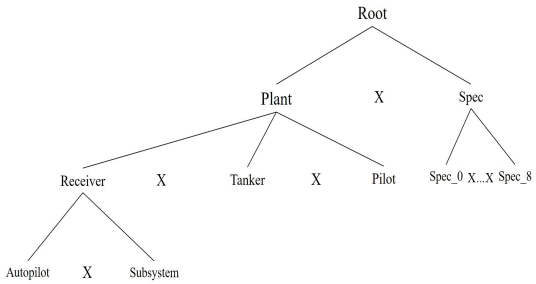


Figure 8: The rough diagram of AAR

in Figure.9, while the detailed information for the superstate **Rendezvous** and **Refueling** are introduced later. The receiver can carry out the normal task sequence: **Standby** $\longrightarrow$ **Rendezvous** $\longrightarrow$ **Refueling** or abandon the current AAR attempt and retreat to **Standby**, **RTL** or **EL** mode. When health conditions become better, it can make transitions like **EL** $\longrightarrow$ **RTL** and **RTL** $\longrightarrow$ **Standby** to resume AAR tasks.

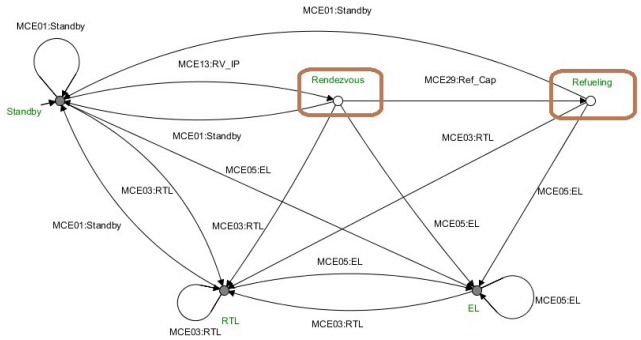


Figure 9: Plant:Autopilot

1).Rendezvous:

This OR superstate has seven simple states, which belong to RV_IP, RV_Climb and RV_Wait sub-modes. The inner transitions of **Rendezvous** are shown in Figure.10. When *MCE13* and *MCE15* happen, **Autopilot** will switch to **Rendezvous** from **Standby** and stay in **RV_01**, the initial state of RV_Climb. At this state, it can make the climbing attempt (*MCE19*), wait (*MCE17*) or retreat

to **Standby**, **RTL** or **EL** state according to current health conditions. There are two possible results for *Climb_Attemp*: climb_success (*ATE24*) or climb_fail (*ATE26*). When *ATE26* happens, **Rendezvous** enters **RV_03**, the initial state of RV_Wait. Similarly, there are two possible routes controlled by the tanker now. If the clearance is issued (*MCE27*), then **Receiver** will enter **RV_04** and then **Autopilot** can switch to **Refueling** superstate (*MCE29*). Otherwise, the receiver has to keep waiting in the observation position and check the waiting time. If it exceeds the threshold, **Rendezvous** should enter **RV_07** state (*ATE30*), where the receiver can retreat to **Standby**, **RTL** and **EL**. If not , it may return to the **RV_03** state (*MCE25*), or switch to the **RV_07** when some subsystem failures happen.[3]

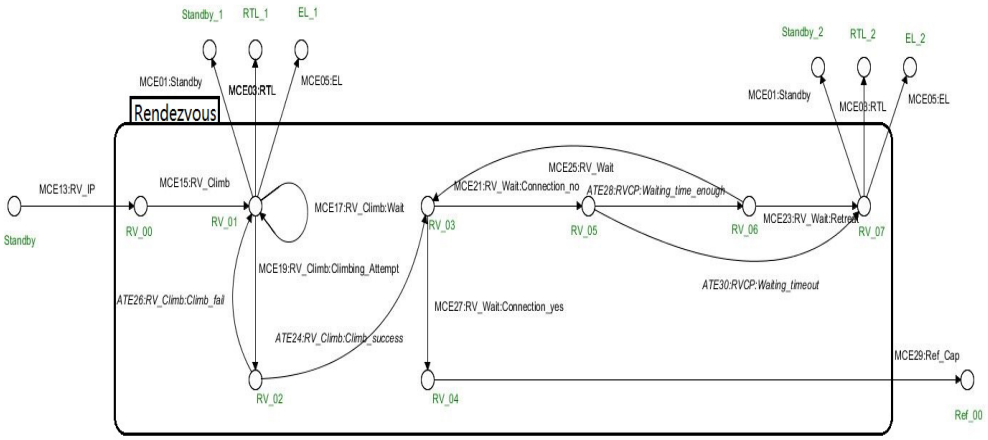


Figure 10: Plant:Rendezvous

2).Refueling:

As mentioned above, when *MCE29* happens, **Autopilot** enters **Refueling** and stays at **Ref_00**, the initial state of Ref_Cap sub-mode. In Ref_Cap mode, the receiver can choose to execute the capture attempt, wait or retreat. When

[3]To avoid the messy transition lines in Figure .10, we draw the **Standby_1,Standby_2** states, which equal to Standby state. Similar for RTL, EL and subsequent pictures

capture successes (*MCE36*), **Refueling** enters Ref_Hold sub-mode to transfer the fuel from the tanker to the receiver. In this mode, failures like hose vibration and fuel leakage may happen. Thus the receiver has the choice to wait (*MCE37*) or retreat(*MCE35*). When transfer successes (*ATE40*), **Refueling** enters the Ref_Wait mode, similar to RV_Wait. After the finish of AAR task (*MCE47*), **Refueling** still offers the choices for the receiver to switch to **RTL** and **EL** states, in case any failures.

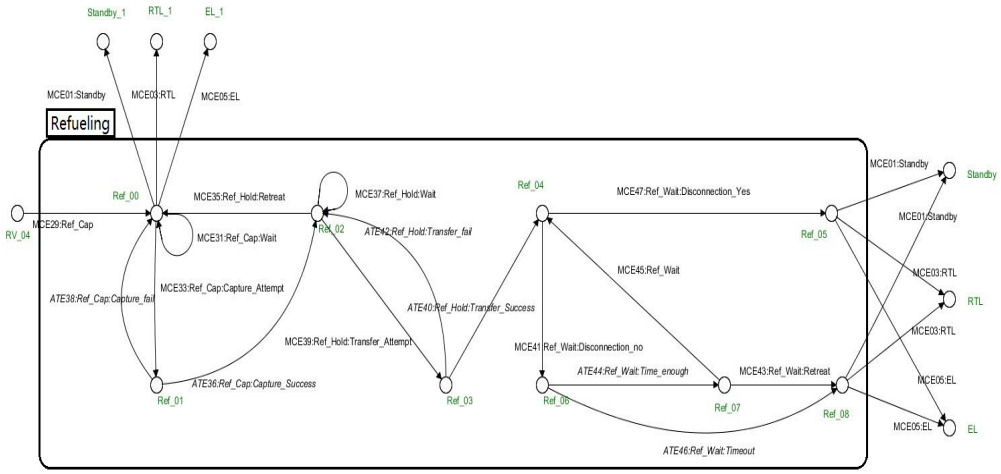


Figure 11: Plant:Refueling

B.Subsystem

Autopilot has modeled the normal task behaviors of AAR, while **Subsystem** models the failure behaviors of AAR that work simultaneously with **Autopilot**. This AND superstate has five components, namely **Propulsion**, **Guidance**, **Drogue&Probe**, **Datalink** and **Other**. The **Other** superstate depicts the rest of subsystems such as electricity supply system and hydraulic system. It can be changed to any system interested in future work.

The holons of **Propulsion**, **Guidance**, **Datalink** and **Other** are quite simple and similar, shown in Figure.12. Take the holon of **Guidance** as an example, when the receiver encounters weak signal, poor visibility, etc., *Fail16*

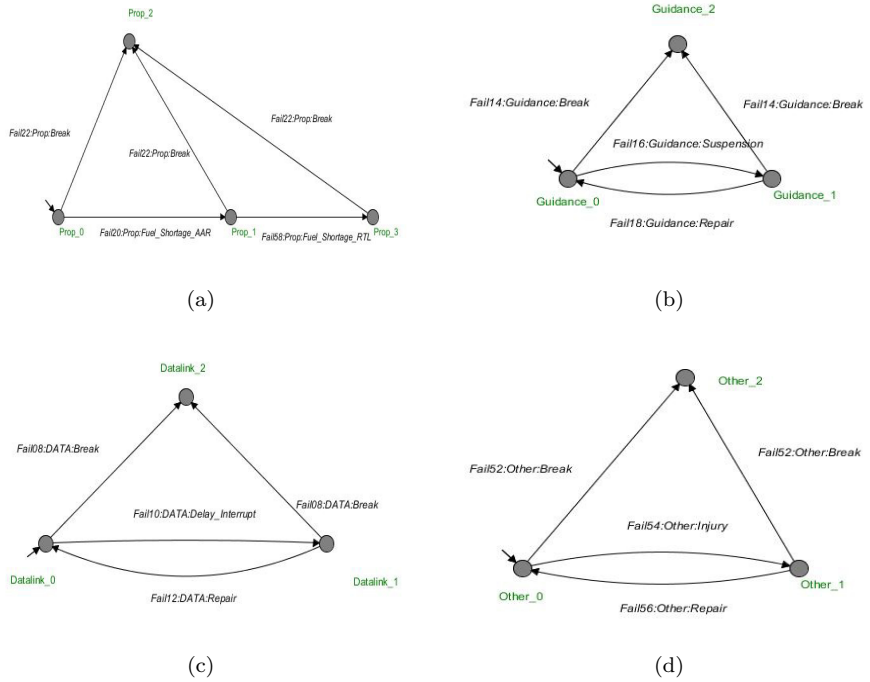


Figure 12: Plant:subsystems (a) Propulsion, (b) Guidance (c) Datalink (d) Other

will happen and **Guidance** enters **Guidance_1** state. It may return back to the normal state, or break down totally (*Fail14*), or stay here.

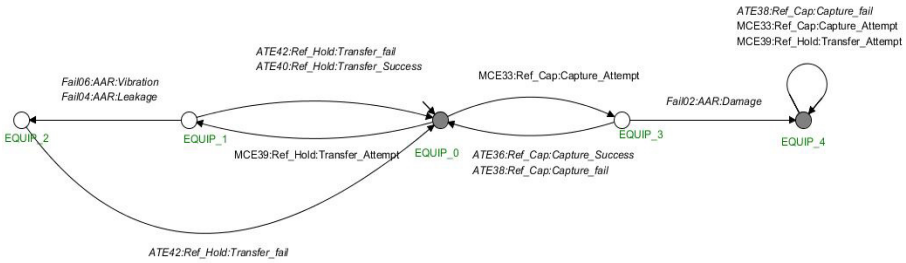


Figure 13: Plant:Drogue&Probe

The holon for **Drogue&Probe** system is more complicated, since related failures will only happen in the **Refueling** Mode. In the right half of Figure.13,

when the receiver begins the capture attempt (*MCE33*), **Drogue&Probe** arrives at **EQUIP_3**. If no failure happens, it will return to the normal state through *ATE36* or *ATE38*, depending on the capture result. If the drogue or probe is damaged in this attempt, *Fail02* will happen and **EQUIP_4** is entered. Since the uncontrolled plant may still try to make capture and transfer fuel, this holon selfloops **EQUIP_4** with *MCE33*, *MCE39* and *ATE38*. The left half of Figure.13. is almost the same as the right half, except that **Equip_2** can return to the initial state.

C.Pilot

Pilot is an OR superstate with four simple states, shown in Figure 14. This holon tells that the pilot's instructions can be responded by the corresponding mode transition or transitions less demanding. For example, if the pilot requires the receiver to return to land (*MIE09*) , then *MCE03* as well as *MCE05* are allowed to happen. In addition, pilot's instruction has priority over the automatic transition of receivers. That is to say, when MIE happens, the receiver should abandon the current procedures and respond to it as soon as possible. This requirement is stated in SR9 and implemented in **Spec_6**.

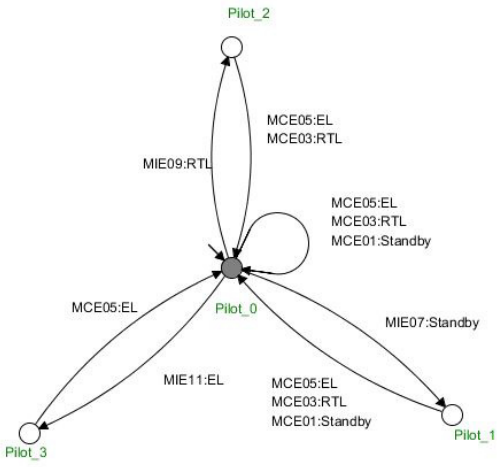


Figure 14: Plant:Pilot

D. Tanker

Tanker is an OR superstate with seven simple states, shown in Figure.15. This plant tells that the tanker can decide when to give the connection/disconnection clearance to the receiver. When the receiver enters the RV_CP mode (*ATE24* or *MCE25* happens), **Tanker** will be at **Tanker_1**. There are two possible routes here. If drogue is busy, *ATE34* would happen and now **Tanker** can only allow *MCE21* to happen, which means the receiver has to keep waiting in the observation position. But if there are available drogues at present, *ATE32* would happen and *MCE27* becomes possible. It is the same for the tanker to give the disconnection clearance.

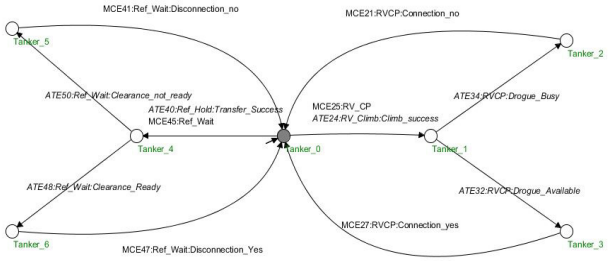


Figure 15: Plant:Tanker

4.2. Specification

In this subsection, holons for specification are presented. Part of these specification holons are shown in Figure.16 and all of them can be found in supporting materials posted online. One safety requirement in Sec.3.2 may be implemented by more than one specification holons, and one holon may depict more than one requirements. This relationship between specifications and safety requirements is shown in Table.3.

These holons are quite similar to the specification holon in Simple AAR, Figure.4. Specifications restrict systems' behaviors through blocking states. Take **Spec_0** as an example, which mainly depicts the SR1 and SR2. In the initial state **Spec_0_0**, transitions to **Standby** or RV_IP mode are allowed to happen.

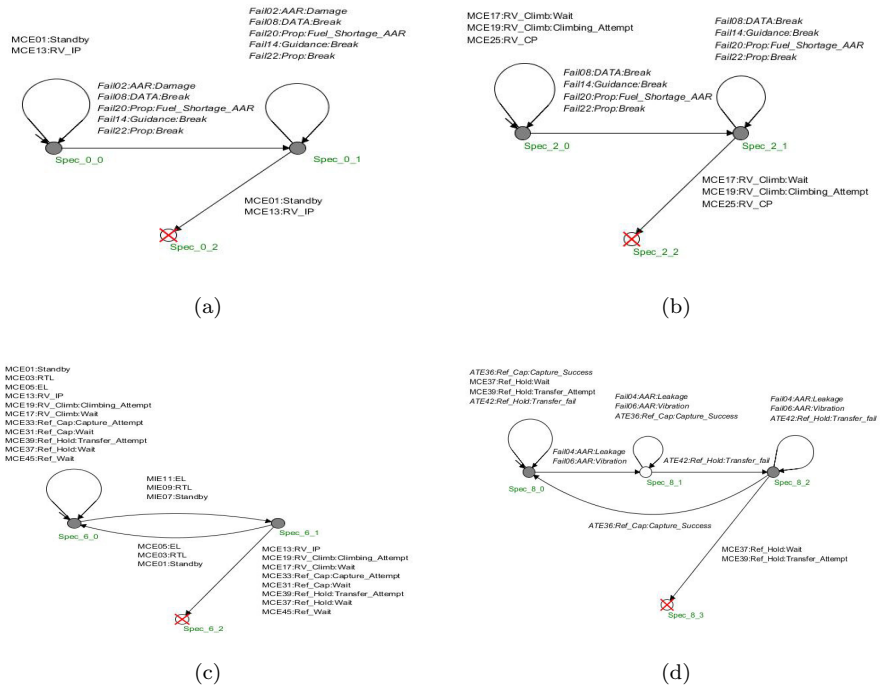


Figure 16: Plant:specification (a) Spec_0:Standby, (b) Spec_2:RV (c) Spec_6:Pilot Priority (d) Spec_8:AAR_Injury

Specification	Spec_0	Spec_1	Spec_2	Spec_3	Spec_4
Safety Requirement	SR1, SR2	SR3	SR4, SR5	SR6	SR1, SR4, SR6
Specification	Spec_5	Spec_6	Spec_7	Spec_8	
Safety Requirement	SR1, SR6	SR9	SR8	SR7	

Table 3: Implementation relationship between Spec and SR

After the occurrence of *Fail02*, *Fail08*, *Fail14*, *Fail20* or *Fail22* (anyone of them will make the transition from **Spec_0_0** to **Spec_0_1**), this holon will enter **Spec_0_1**. In this state, *MCE01* and *MCE13* will lead the system to blocking state **Spec_0_2**, therefore these two events are forbidden. Notice that those uncontrollable failure events are still allowed to happen because uncontrollable

events cannot be controlled

4.3. Supervisor

The whole plants and specification of AAR is computed in STSLib. The whole system has 2.5×10^{12} states, and STSLib computes it and output the supervisor within 2 seconds on a personal computer with 2.6G HZ I5 CPU and 8GB RAM. The supervisor only has 15 binary decision diagrams or 51 records in the control data table. This means that the logic controller just has to search for these 51 records to decide whether an mode transition should be enabled and the whole system with 2.5×10^{12} states can operate without unexpected failure scenarios. The supervisor can also be found in the supporting materials posted online.

5. Conclusion and Future Work

This paper proposes an STS-based method to design a failsafe mechanism for receivers in Autonomous Aerial Refueling (AAR). The whole AAR procedure is decomposed to five modes, namely Standby, Rendezvous, Refueling, Return to Land and Emergency Landing. Common safety issues related with four critical subsystems are summarized, namely Propulsion, Guidance, Drogue&Probe and Datalink system. To restrict the uncontrolled plant consisting of mode transitions and subsystems, nine safety issues are put forward and modeled as the specifications. With plants and specifications in hand, STSLib is used to compute the final supervisor and a 51-records control data table is achieved.

In recent future, we are going to implement our supervisor with the simulation environment in MATLAB to verify its correctness. Meanwhile, we will demonstrate the flexibility of our STS-based method by modifying the original system. We aim to write an international journal paper about our work at the end.

Acknowledgments

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada, Grant #DG_480599.

References

[1] R. P. Dibley, M. J. Allen, N. Nabaa, Autonomous airborne refueling demonstration: Phase i flight-test results.

[2] P. R. Thomas, U. Bhandari, S. Bullock, T. S. Richardson, J. L. du Bois, Advances in air to air refuelling, Progress in Aerospace Sciences 71 (Supplement C) (2014) 14 – 35. doi:https://doi.org/10.1016/j.paerosci.2014.07.001.

[3] A. Degani, M. Heymann, Formal verification of human-automation interaction, Human factors 44 (1) (2002) 28–43.

[4] I. M. Mitchell, A. M. Bayen, C. J. Tomlin, A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games, IEEE Transactions on automatic control 50 (7) (2005) 947–957.

[5] J. Ding, J. Sprinkle, C. J. Tomlin, S. S. Sastry, J. L. Paunicka, Reachability calculations for vehicle safety during manned/unmanned vehicle interaction, University of California, Berkeley 94720.

[6] N. Meskin, K. Khorasani, C. A. Rabbath, A hybrid fault detection and isolation strategy for a network of unmanned vehicles in presence of large environmental disturbances, IEEE Transactions on Control Systems Technology 18 (6) (2010) 1422–1429.

[7] G. P. Falconí, F. Holzapfel, Adaptive fault tolerant control allocation for a hexacopter system, in: American Control Conference (ACC), 2016, IEEE, 2016, pp. 6760–6766.

[8] Y. Zhang, J. Jiang, Issues on integration of fault diagnosis and reconfigurable control in active fault-tolerant control systems, IFAC Proceedings Volumes 39 (13) (2006) 1437–1448.

[9] P. J. G. Ramadge, W. M. Wonham, The control of discrete event systems, Proceedings of the IEEE 77 (1) (1989) 81–98. doi:10.1109/5.21072.

[10] W. M. Wonham, Supervisory control of discrete-event systems, Encyclopedia of Systems and Control (2015) 1396–1404.

[11] C. Ma, W. M. Wonham, Nonblocking supervisory control of state tree structures, IEEE Transactions on Automatic Control 51 (5) (2006) 782–793.

[12] K. Cai, W. M. Wonham, Supervisor localization: a top-down approach to distributed control of discrete-event systems, IEEE Transactions on Automatic Control 55 (3) (2010) 605–618.

[13] B. A. Brandin, F. Charbonnier, The supervisory control of the automated manufacturing system of the aip, in: Computer Integrated Manufacturing and Automation Technology, 1994., Proceedings of the Fourth International Conference on, IEEE, 1994, pp. 319–324.

[14] M. Seidl, Systematic controller design to drive high-load call centers, IEEE transactions on control systems technology 14 (2) (2006) 216–223.

[15] R. J. Theunissen, M. Petreczky, R. R. Schiffelers, D. A. van Beek, J. E. Rooda, Application of supervisory control synthesis to a patient support table of a magnetic resonance imaging scanner, IEEE Transactions on Automation Science and Engineering 11 (1) (2014) 20–32.

[16] J. Markovski, K. G. M. Jacobs, D. A. V. Beek, L. J. A. M. Somers, J. E. Rooda, Coordination of resources using generalized state-based requirements, IFAC Proceedings Volumes 43 (12) (2010) 287–292.

[17] S. T. J. Forschelen, J. M. V. D. Mortel-Fronczak, R. Su, J. E. Rooda, Application of supervisory control theory to theme park vehicles, Ifac Proceedings Volumes 22 (4) (2012) 511–540.

[18] C. Ma, W. Wonham, Stslib and its application to two benchmarks, in: Discrete Event Systems, 2008. WODES 2008. 9th International Workshop on, IEEE, 2008, pp. 119–124.

[19] C. Bolkcom, Air force aerial refueling methods: flying boom versus hose-and-drogue, LIBRARY OF CONGRESS WASHINGTON DC CONGRESSIONAL RESEARCH SERVICE, 2006.

[20] Air to air refuelling, atp-56(b), NATO Standardization Agency.