Safety Risk Assessment for Aircraft Fuel Management

Master thesis

Viktor Gregor

Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

Study programme	Economic and Financial Mathematics
Study field	9.1.9. Applied Mathematics
Department	Department of Applied Mathematics and Statistics
Date	31.7.2015

Author:	Supervisor:	Supervisory specialists:
Bc. Viktor Gregor	*prof. RNDr. Marek Fila, DrSc.*	*Mgr. Sybert Stroeve, PhD.* *prof. Dr. Michel Mandjes*

99488828

Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

THESIS ASSIGNMENT

Name and Surname: Bc. Viktor Gregor
Study programme: Economic and Financial Mathematics (Single degree study, master II. deg., full time form)
Field of Study: 9.1.9. Applied Mathematics
Type of Thesis: Diploma Thesis
Language of Thesis: English
Secondary language: Slovak

Title: Safety Risk Assessment for Aircraft Fuel Management

Aim: To develop and evaluate a risk model for aircraft fuel management. The model should enable probability assessment of fuel management related events such as fuel starvation and landing with less than final reserve fuel.

Supervisor: prof. RNDr. Marek Fila, DrSc.
Department: FMFI.KAMŠ - Department of Applied Mathematics and Statistics
Head of department: prof. RNDr. Daniel Ševčovič, CSc.

Assigned: 29.01.2014

Approved: 10.02.2014 prof. RNDr. Daniel Ševčovič, CSc.
Guarantor of Study Programme

....
Student Supervisor

99488828

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Viktor Gregor
Študijný program: ekonomická a finančná matematika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: 9.1.9. aplikovaná matematika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: anglický
Sekundárny jazyk: slovenský

Názov: Safety Risk Assessment for Aircraft Fuel Management
Vyhodnocovanie bezpečnostného rizika v manažmente paliva pre lietadlo

Cieľ: To develop and evaluate a risk model for aircraft fuel management. The model should enable probability assessment of fuel management related events such as fuel starvation and landing with less than final reserve fuel.

Vedúci: prof. RNDr. Marek Fila, DrSc.
Katedra: FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky
Vedúci katedry: prof. RNDr. Daniel Ševčovič, CSc.

Dátum zadania: 29.01.2014

Dátum schválenia: 10.02.2014

prof. RNDr. Daniel Ševčovič, CSc.
garant študijného programu

....

študent vedúci práce

Abstract

Focus of this thesis is the assessment of risk probabilities of events related to fuel management. This includes events like landing at the airport with less than final reserve fuel (FRF) and running out of all usable fuel. FRF is the amount of fuel defined by regulations and an aircraft has to have at least this amount remaining in the tanks after the flight is finished. It is considered a safety incident if this is not achieved.

The goal was to develop a mathematical model and simulation model with which it is possible to estimate the probabilities of these two events. The model was developed using dynamically coloured Petri nets. To estimate the probabilities, we used regular Monte Carlo simulation method and splitting method. Monte Carlo provided important insights, but was insufficient to provide results for rare events like fuel starvation. The splitting method is an adequate algorithm to speed up the simulations, especially when the size and complexity of the model makes the use of other methods, like importance sampling, infeasible.

Purpose of this project is to develop the first version of a risk assessment model within the context of fuel management. This model should serve as a basis for further research of this topic.

Table of Contents

Abbreviations

Acronym	Description
AOC	Airline operations control
ATC	Air traffic control
ATCo	Air traffic controller
FMS	Flight management system
FRF	Final reserve fuel
IATA	International Air Transport Association
ICAO	International Civic Aviation Organization
LPN	Local Petri net
OFP	Operational flight plan
PF	Pilot flying
PM	Pilot monitoring
PNF	Pilot not flying
PNR	Point of no return

1 Introduction

1.1 Context

Fuel management is an important part of flight planning, both before flight commencement and in-flight. In current flight operations, fuel management is strongly based on regulations. These pose requirements for several fuel components for the planned flight, including taxi fuel, trip fuel, contingency fuel, final reserve fuel (FRF), alternate fuel and extra fuel. During flight, pilots must ensure that the aircraft can land with at least the FRF left in the tanks (FRF is amount of fuel required to fly for 30 minutes for an aircraft with turbine engines). It is considered a safety incident if this is not achieved. In decision-making during flight, pilots have to consider fuel sufficiency for reaching the destination or alternate airport while considering various uncertainties that may impact flight duration and fuel consumption, e.g. adverse weather conditions, diversions, delays near airports, closed airports or fuel leakage. This means that pilots are (implicitly) making a risk assessment for fuel sufficiency. Typically, the risks of attaining too low fuel levels are considered to be small, but pilots may have to make decisions with high stakes. For instance, pilot may have to decide to approach to either airport A or airport B, without having sufficient fuel left to reconsider this choice and divert to another airport at a later stage. On the contrary of flight safety, aircraft fuel management is being influenced by economic and environmental factors which tend to promote reduction of fuel intake.

1.2 Objective

The objective of this study is to develop and evaluate a risk model for aircraft fuel management. This risk model should enable probability assessment of fuel management related events like landing with less than FRF and fuel starvation. The probability estimates of such events should be conditional on given flight-related parameters such as length of the flight, expected circumstances of the air traffic at destination or expected weather during the route. The risk model should evaluate fuel consumption during the flight taking into account uncertainty in expected conditions including regular situations, e.g. delays near airports as well as rarer situations, e.g. airport closure or fuel leakage.

1.3 Safety risk assessment approach

To develop this model, we will use the TOPAZ (Traffic Organization and Perturbation AnalyZer) safety risk assessment cycle described in chapter 2 of paper (Blom, Stroeve, & de Jong, 2006). The steps of the safety risk assessment approach, as shown in Figure 1.1, are the following:

Step 0: Identify objective

The objective of the assessment is determined, as well as the safety context and the scope.

Step 1: Determine operation

During this part, a complete and concise overview of the operation is obtained. This part should also include any particular assumptions made about the operation during the assessment.

Step 2: Identify hazards

Hazards associated with the operation are identified in this step. By hazard we mean any event or situation that may lead to danger (this is called root hazard) or that can hamper the resolution of danger caused by different hazard (resolution hazard). The goal of this step is to construct a list of hazards that is as complete as possible considering the risks that are being assessed.

Step 3: Construct scenarios

After completing the list of hazards, it is processed to deal with overlapping, similar or ambiguously described hazards. Hazards are then grouped into clusters that have similar effects or causes. A list of all relevant scenarios is constructed and described using the hazard clusters.

Step 4: Identify severities

Scenarios identified in previous step are now categorized according to the severity of their possible impacts.

Step 5: Assess frequency

In this step, the frequency of each outcome of each relevant scenario is evaluated. This is the step where the actual risk assessment model is developed and simulations are carried out.

Step 6: Assess risk tolerability

During this step the tolerability of the risk is assessed for each scenario outcome and severity category.

Step 7: Identify safety bottlenecks

For each scenario, the hazards that most contribute to high risk levels are identified as safety bottlenecks. This step is important as it gives the conceptual operational designers directions for finding potential risk mitigating measures of the operation.

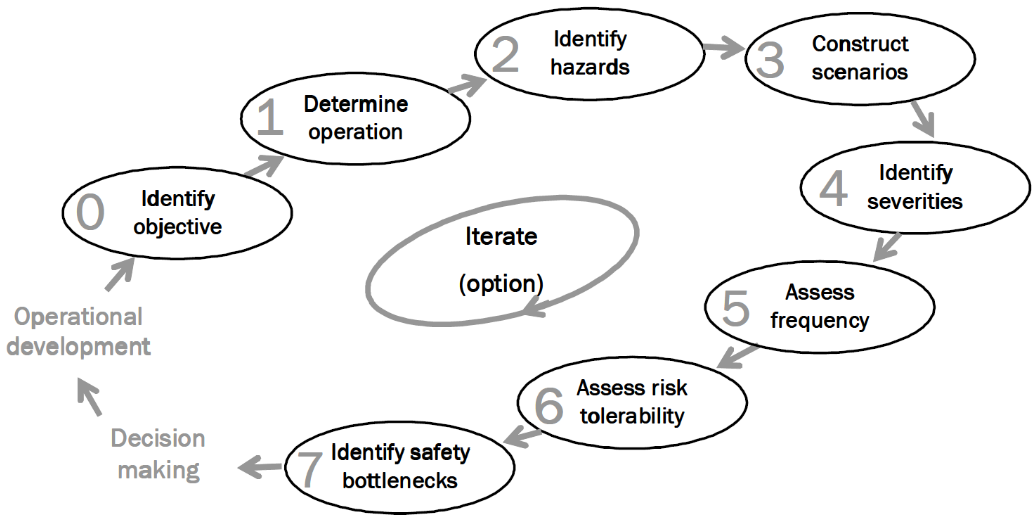


Figure 1.1: steps of safety risk assessment cycle as defined in TOPAZ

Based on the described objective of this study, our goal is to develop a model capable of estimating the risk probabilities. That means our objective is to complete the steps up to step *5 Assess frequency*. Next steps are the objective of further research and are out of scope of this study.

1.4 Set-up of the report

In chapter 2, we are going to determine the scope of the assessment to define which aspects of the real world operation are covered in the model.

Third chapter explains the operation itself. We describe the important human roles of the operation and procedures. The purpose of this chapter is to provide context for the reader before we define the model. This chapter is based on manuals and regulations related to the operation.

Chapter 4 explains how we collected the list of hazards related to the fuel management and categorized them into relevant clusters. This is an important step also for future versions of the model since it can function as a checklist of things that are still to be included in the model.

In fifth chapter, we illustrate the theoretical background that is necessary to understand the risk assessment model. We will describe dynamically coloured Petri nets and explain the concept on a short example. The complete definitions can be found in (Everdij, 2010). Then we will explain the idea of the simulating method used to estimate the probabilities of rare events, the splitting method. The algorithm of the splitting method together with three possible variations is also included.

Chapter 6 defines the whole model that was developed. We describe it by defining local Petri nets for each of the agents of the model: environment, airports, airline operations control (AOC), aircraft and crew of the aircraft.

In chapter 7, we outline the process of implementing the model into Java programming language and illustrate some of the outputs of the program.

Chapter 8 states the results of the simulation together with the exact parameters of the splitting method that were used.

The recomendations for further extension of the model are listed in chapter 9. At this point, we also discuss the results of simulations in relation to reality and possible future versions of the model.

The conclusion of the report is in chapter 10.

2 Scope of the risk assessment and safety context

2.1 Scope

The scope of this safety assessment is limited to the risk of attaining low usable fuel levels during flight due to insufficient planning, loss of fuel or inability to use remaining fuel. Other types of risks, such as a collision between aircraft or controlled flight into terrain are out of the scope of this safety assessment. The safety risks of the aircraft fuelling process or of the aircraft fuel combustion process are also out of the scope of this study.

The considered operations are current-day commercial air transport operations with turbine engine aeroplanes. The fuel management related operations included in this study are pre-flight fuel planning and in-flight fuel management. Flights are considered from gate to gate, meaning starting at the gate of the departure airport until arrival at the gate of the destination airport. In addition to the roles of the cockpit crew in fuel management, the roles of airline operation control centres and air traffic control centres are included to the extent that they effect the aircraft routing and flight duration.

2.2 Safety context

In Book 2, section AMC 25.1309 of (European Aviation Safety Agency, 2014), the classification of failure conditions according to the severity of their effects is defined as follows:

1. *No safety effect:* Failure conditions that would have no effect on safety.
2. *Minor:* Failure conditions which would not significantly reduce aeroplane safety, and which involve crew actions that are well within their capabilities. This may include, for example, a slight reduction in safety margins or functional capabilities, a slight increase in crew workload, such as routine flight plan changes, or some physical discomfort to passengers or cabin crew.
3. *Major:* Failure conditions which would reduce the capability of the aeroplane or the ability of the crew to cope with adverse operating conditions meaning, for example, a significant reduction in safety margins or functional capabilities, a significant increase in crew workload or in conditions impairing crew efficiency, or physical distress to passengers or cabin crew, possibly including injuries.
4. *Hazardous:* Failure conditions, which would reduce the capability of the aeroplane or the ability of the crew to cope with adverse operating conditions to the extent that there would be a large reduction in safety margins or functional capabilities, physical distress or excessive workload of the flight crew or serious or fatal injury to a relatively small number of the occupants other than the flight crew.

5. *Catastrophic:* Failure Conditions, which would result in multiple fatalities, usually with the loss of the aeroplane.

Safety objective is defined as follows:

1. No safety effect failure conditions have no probability requirement.
2. Minor failure conditions also have no probability requirement.
3. The probability of the major failure conditions per flight hour must be less than 10^{-5}.
4. The probability of the hazardous failure conditions per flight hour must be less than 10^{-7}.
5. The probability of the catastrophic failure conditions per flight hour must be less than 10^{-9}.

In the context of fuel management, we consider two main failure conditions. One is a fuel emergency situation when the predicted fuel available upon landing at the nearest airport is less than the planned FRF. Second is the fuel starvation meaning that aircraft has no remaining usable fuel left. The first event can be considered as minor, major or even hazardous failure condition depending on the situation (e.g. distance from the nearest airport) so the range of probability should be 10^{-5} and less. Fuel starvation on the other hand can be considered as the ultimate fuel-related failure event. This is a catastrophic failure condition and the probability of that happening has to be less than 10^{-9} according to regulations.

3 Description of the operation

Third chapter explains the operation from perspective of regulations and laws. Throughout this chapter, we mostly paraphrase manuals and guidance materials from International Air Transport association (IATA) and International Civil Aviation Organization (ICAO).

3.1 Human roles and responsibilities

3.1.1 Pilot flying and pilot not flying

Commercial aircrafts usually require a two-person flight crew. One pilot is the aircraft commander (also referred to as the pilot-in-command), who must be appropriately qualified and hold the rank of Captain. The other is the First Officer or Co-Pilot.

Before the commencement of the flight, the aircraft commander decides which pilot will take direct responsibility for flying the aircraft for the complete flight or for particular parts of it and he becomes *Pilot Flying* (PF). The other pilot is then designated for that part as *Pilot Monitoring* (PM) or alternatively as *Pilot Not Flying* (PNF). In that role, he must monitor the flight management and aircraft control actions of the PF and carry out support duties such as communication and check-list reading. One of the most important aspects of the duties of any PM/PNF is the cross-check of the actions of PF. Indeed, this part of the role represents the most important reason why a two-pilot flight crew is required.

Whatever their role, the designated aircraft commander is responsible for all aspects of the safe operation of the aircraft.

3.1.2 Air traffic controllers

Air traffic controller (ATCo) maintains the safe, orderly and efficient flow of air traffic in the global ATC system. Controllers apply separation rules to keep aircraft at a safe distance from each other in their area of responsibility and move all aircraft safely and efficiently through their assigned sector of airspace, as well as on the ground.

According to (International Air Transport Association, 2009), whenever safety permits, ATCo should:

- inform pilots or operators of any delay
- endeavour to keep aircraft on taxiways moving at all times
- approve alternate runways when practicable
- approve take-off in the direction of flight
- cancel speed restrictions as soon as practicable
- have available the latest meteorological information for their sector to assist pilots in assessing appropriate level and speed
- be aware of the impact of assigned levels on fuel efficiency
- try to approve optimum altitudes

- co-ordinate and issue descent clearance timely and at pilot`s discretion
- if a hold is anticipated, advise the pilot as soon as possible

3.1.3 Airline operation controllers

According to (Grandeau, 1995), Airline Operations Control (AOC) is section of the airline responsible for day to day operations. Airline operations are handled in two phases, strategic and tactical. Strategic operations are concerned with scheduling and planning. The tactical side is responsible for the process of executing the airline schedules on a daily basis. This involves three activities: executing the pre-planned schedules, updating the schedules for minor operational deviations and rescheduling for irregular operations.

Flight Dispatchers (part of AOC) are licensed personnel responsible for the safety and operational control of each flight before take-off and during flight. By law, this responsibility is shared equally between the Dispatcher and the aircraft's captain.

Dispatcher's responsibilities can be broken down into flight planning, flight dispatch, and flight following. Flight planning is done in advance of take-off and includes computing and filing of the flight plan and calculating fuel requirements. Flight dispatch covers the collection of load information, the calculation of take-off and landing performance and the monitoring of the availability of all necessary flight resources. Flight following is the process of tracking a flight's progress and acting as intermediary between the pilots and the ATC system during any problems.

3.2 Procedures

Considered procedures can be divided into two main parts:

- *pre-flight fuel planning*
- *in-flight fuel management*

The main objective of the fuel planning and fuel management is to ensure that aeroplane has a sufficient amount of fuel to safely complete the planned flight and to allow for deviations from the planned trajectory.

3.2.1 Pre-flight fuel planning

Pre-flight fuel planning includes planning the optimal flight route (specified by waypoints) taking into account predictions of airspace and airports conditions. According to section 4.3.4.3.1 of (International Civil Aviation Organization, 2012), the flight plan also has to include at least one destination alternate airport or in case the destination airport is isolated, a point of no return (PNR) has to be determined along the route. The flight shall not continue past the PNR unless current assessment of conditions at destination indicates that a safe landing can be made.

Using the flight route, the dispatcher, who is responsible for the plan, calculates the amount of fuel intake before the flight. After the flight plan is constructed, the dispatcher sends it to flight crew.

Captain can request changes or ask for extra fuel (discretionary fuel). Captain always has to approve the plan before the flight commences.

The following description of fuel categories is from the Flight Planning and Fuel Management Manual (International Civil Aviation Organization, 2012). Calculated usable fuel includes these categories:

1. *taxi fuel*
2. *trip fuel*
3. *contingency fuel*
4. *alternate fuel*
5. *final reserve fuel*
6. *additional fuel*
7. *extra fuel*

3.2.1.1 Taxi fuel

The amount of fuel expected to be consumed before take-off. Local conditions at the airport are taken into account.

Many airlines use standard taxi fuel allowances. In many situations these allowances exceed the requirements and result in flight taking off with some of taxi fuel still remaining. This affects efficiency as some of the extra fuel is consumed to carry it to the destination. In (International Air Transport Association, 2009), it is recommended to use statistical taxi fuel that is responsive to changing conditions. Values are being measured for example by station, city-pair, time of the day and day of week.

3.2.1.2 Trip fuel

The amount required to enable the aeroplane to fly from take-off until landing at the destination airport. Trip fuel also includes compensation for foreseen factors such as meteorological conditions and anticipated delays.

More specifically, trip fuel includes fuel required for aircraft to:

- take-off and climb to cruising altitude
- cruise from top of climb to top of descent
- descent
- approach
- land

3.2.1.3 Contingency fuel

The amount of fuel calculated to compensate for unforeseen events during flight. It is the maximum of these two amounts: 5% of the planned trip fuel and the amount required to fly for five minutes at holding speed at 450m above the destination airport in standard conditions.

3.2.1.4 Alternate fuel

The amount of fuel required to perform a missed approach at the destination airport, fly to the alternate airport and land there.

3.2.1.5 Final reserve fuel

The amount of fuel required to fly at holding speed at 450m above airport in standard conditions for 30 minutes with turbine engine.

The amount is calculated with the estimated mass on arrival at the alternate airport. It is recommended that operators should determine one FRF value for each aeroplane type and variant in their fleet rounded up to an easily recalled figure.

3.2.1.6 Additional fuel

The amount of fuel that will enable the aircraft to descent as necessary and proceed to an alternate airport in the event of engine failure or loss of pressurization, whichever requires more fuel based on the assumption that such a failure occurs at the most critical point along the route. Furthermore it enables the aeroplane to fly for 15 minutes at holding speed at 450m above the airport in standard conditions and make an approach and landing.

Additional fuel is only required if the minimal amount of fuel calculated in paragraphs 3.2.1.1 – 3.2.1.5 is not sufficient for such an event.

3.2.1.7 Extra fuel

The amount of fuel carried at the discretion of the pilot-in-command.

3.2.2 In-flight fuel management

In chapter 6 of (International Civil Aviation Organization, 2012), the in-flight fuel management is summarized as follows:

- The operator establishes the policies and procedures to ensure that in-flight fuel checks and fuel management are performed.
- The pilot-in-command continually ensures that the amount of usable fuel available is not less than the amount required to proceed to an airport where a safe landing can be made with FRF remaining.

In-flight fuel management policies are not intended to replace pre-flight planning or in-flight re-planning activities but to act as controls to ensure planning assumptions are continually validated. Safe conclusion of any flight depends on the accuracy and completeness of initial planning as well as the intelligent use of on board fuel supply. Even the best fuel planning cannot ensure a safe outcome, if the execution of the plan is faulty or invalidated planning assumptions go undetected. As such, flight planning activities must be complemented by practical in-flight fuel management policies and procedures.

As is stated in chapter 6.2 of (International Civil Aviation Organization, 2012), these policies and procedures are given by the operator and among other things they address:

- The variables used in the calculation of the fuel required to take-off or to continue beyond the point of in-flight re-planning.
- The selection of the alternate airport and used fuel planning methods.
- Flight crew responsibilities and actions related to pre-flight fuel planning and fuel load determination.
- The operational flight plan (OFP) and instructions for its use.
- Deviations from the OFP or other actions that could invalidate flight planning.
- Assumptions (e.g. acceptance of direct routings, altitude changes or speed changes).
- Actions related to the acquisition of timely and accurate information that may affect in-flight fuel management (e.g. weather, airport condition).
- The practical means for the in-flight validation (or invalidation) of assumptions made including instructions for recording and evaluating remaining usable fuel at regular intervals.
- The factors to be considered and actions to be taken by the pilot-in-command if flight planning assumptions are invalidated (re-analysis and adjustment) including guidance on the addition of discretionary fuel.
- Actions to be taken by the pilot-in-command to protect FRF including instructions for requesting delay information from ATC, declaration of *MINIMUM FUEL* and declaration of a fuel emergency (*MAYDAY MAYDAY MAYDAY FUEL)*.

Chapter 6.5 of (International Civil Aviation Organization, 2012) declares that in order for successful fuel management to occur, operator policies and procedures require that at regular intervals, specified points in the OFP or when otherwise required, the pilot-in-command:

- Compares actual consumption with planned consumption.
- Verifies fuel quantity used against the fuel quantity expected to be used up to that point.
- Verifies fuel quantity remaining against the computed planned remaining quantity at that point.
- Records and forwards fuel use and quantity information to the data collection system.
- Identifies discrepancies between the information provided by the OFP and actual fuel remaining to find the cause and to initiate appropriate action.
- Considers operational factors and potential actions to be taken if flight planning assumptions are invalidated (re-analysis and adjustment). This is important if the fuel remaining is insufficient to complete the flight as originally planned. In such cases the pilot-in-command should evaluate the traffic and the operational conditions prevailing at the destination airport, at the alternate airport and at any other adequate airport before deciding on a new course of action.

- If operating to an isolated airport, re-calculates the position of the Point of No Return (PNR) based on actual fuel consumption and fuel remaining and determines if applicable conditions are satisfied for proceeding beyond the PNR to the destination airport.
- Determines if remaining fuel is sufficient to safely complete the flight as planned.
- Communicates with ATC to request delay information.
- Declares “MINIMUM FUEL” when required.
- Declares a “MAYDAY MAYDAY MAYDAY FUEL” to indicate a fuel emergency when required.
- After these declarations, he takes the appropriate action and proceeds to the nearest airport where a safe landing can be made.

3.2.2.1 Minimum fuel declaration

As stated in chapter 6.7 of (International Civil Aviation Organization, 2012), the pilot-in-command shall advise ATC of a minimum fuel state by declaring “MINIMUM FUEL” when, having committed to land at a specific airport, the pilot calculates that any change to the existing clearance to that airport may result in landing with less than planned FRF.

This is not an emergency situation but an indication that an emergency situation is possible should any additional delay occur. Therefore, pilots should not expect any form of priority handling as a result of a “MINIMUM FUEL” declaration. ATC will, however, advise the flight crew of any additional expected delays. When “MINIMUM FUEL” is declared, it means that the pilot-in-command has already committed to land at a specific airport and is concerned that a landing may occur with less than FRF remaining.

Pilot-in-command is required to declare a situation of emergency by broadcasting “MAYDAY MAYDAY MAYDAY FUEL” when the calculated usable fuel to be available upon landing at the nearest suitable airport where a safe landing can be made will be less than planned FRF. This declaration is used when all opportunities to protect FRF have been exploited. It is important to note that an emergency declaration opens all options for pilots and also allows ATC more flexibility in handling the aircraft.

4 Identification and clustering of hazards

4.1 Identification of hazards

Hazards are events or conditions with possibly negative effects on the safety of operations, implying that they may lead to injuries or loss of life or to damage of material. As basis for a safety risk assessment, a broad and diverse set of hazards needs to be identified. Such hazard identification can be done in various ways, including studying related literature, using hazard databases and organizing hazard brainstorm sessions with operational experts.

As a first step in the hazard identification, Flight Planning and Fuel Management Manual (International Civil Aviation Organization, 2012) was studied. The hazards identified in this document are listed in Table 12.2 in Appendix D.

As a second step, a database of hazards in (Stroeve, Doorn, & Everdij, 2013) was studied. In particular hazards that may affect current air traffic operations are presented in Appendix B of this paper. Table 12.3 in Appendix D lists relevant hazards that may affect fuel management.

Additionally, in Table 12.4 in Appendix D, we list hazards that were identified in (European Commission, 2008). In Table 12.5 in Appendix D, we list hazards from other various sources including consulting with NLR experts.

4.2 Clustering of hazards

After we complete the list of hazards, we categorize them into clusters with similar affect or cause and we connect the clusters to create a structure. First, we consider the causes of fuel-related problems. Practically in any situation in which we have less fuel than we need, the cause is from one of these general categories of root hazards:

A. Fuel consumption is higher than expected.
B. Flight route is longer than expected.
C. Part of planned fuel is unavailable because:
    1. We lost fuel from tanks due to fuel leakage
    2. Fraction of fuel in tanks cannot be used by engines
    3. The fuel intake before the flight was lower than it should have been according to adequate plan.

Besides root hazards, we also have hazards in our list that do not cause fuel-related problems by themselves, but they can contribute to the problem considerably. These are resolution hazards. We deliberately choose a subset of resolution hazards to form an important cluster and that is Situation Awareness.

Now we assign each hazard from the list to one (or more) cluster. Then we divide the clusters into subclusters based on similar causes of the problem. In Figure 4.1, we see the structured clusters. The hazards, included in each cluster and subcluster, are listed in Table 4.1.

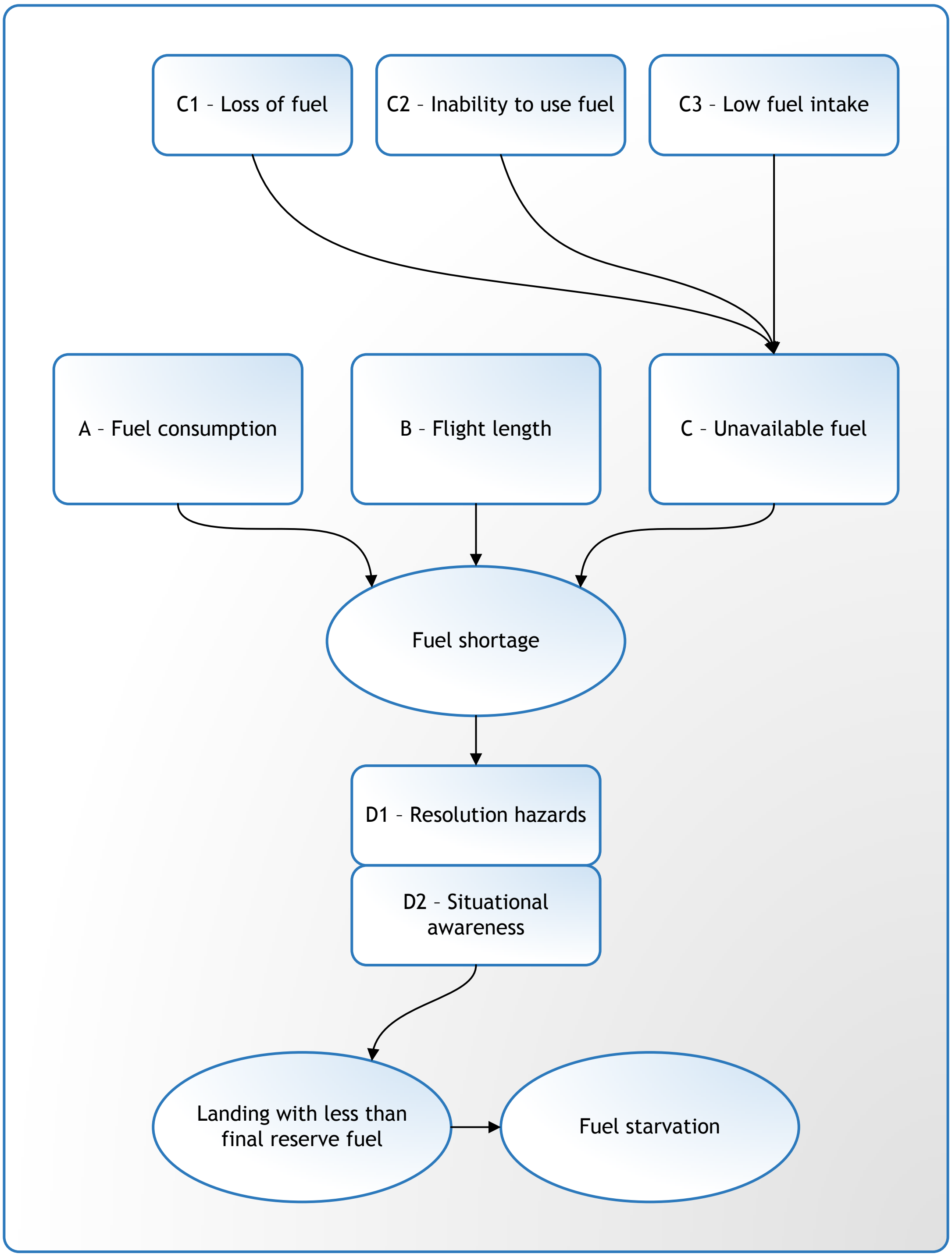


Figure 4.1: hazard clusters constructed from the list of hazards

Table 4.1: clusters and subclusters of hazards

A - FUEL CONSUMPTION	
A.1 - adverse wind	
H001	Tropical storm, winter storm, tornado, cyclone
H004	Strong winds
H006	Wind shear
H118	Weather influences the functioning of airborne systems
H127	Different wind speeds at different heights (vertical wind shear)
H128	Strong variation in wind
H130	Jet stream
H131	Mountain waves
A.2 - adverse weather	
H002	Icing, freezing precipitation, snow
H003	Heavy rain
H005	Thunderstorms
H008	Dust or sand storms
A.3 - unexpected weather	
H025	Hazards affecting meteorological reporting or forecasting
H121	Weather forecast wrong
H122	Sudden weather change disturbs planning
H124	Weather info not available
A.4 - degradation of the aircraft	
H046	Degradation of aircraft structure
H048	Degradation of one or multiple engines
H099	Pilot does not detect degradation of airborne system
H151	Incorrect fuel bias
A.5 - aircraft system failure	
H014	Mechanical failure of an aeroplane system
H043	Airborne systems not working, e.g. cockpit display, flight management system, or large electronic failure
H044	Problem with instrument landing system
H113	Rapid descent due to an aircraft system failure
H141	Pre-flight maintenance error
H146	Electrical failure
H147	Inability to fully retract flaps after missed approach
A.6 - aircraft flying with non-optimal parameters (e.g. altitude, speed)	
H078	Aircraft flies near its envelope extremes
H140	Crew does not follow the applicable procedures correctly
H148	Plane is flying in lower altitude than expected
H149	Pilot is not flying in optimal mode
A.7 - speed restrictions	
H029	En-route speed restriction

H143	Inadequate certification requirements
A.8 - improper use of new equipment	
H021	Operational changes, e.g. new equipment, adapted procedures
B - FLIGHT LENGTH	
B.1 - diversion from planned route by pilot	
H039	Lack of routing accuracy of flight management system
H060	Flight plans of ATC system and FMS differ
H083	Pilot selects wrong route in flight management system
H084	Pilots disconnect FMS
H095	Pilots are flying to wrong airport
H104	Pilot incapacitation, e.g. pilots falling asleep, pilots die, or pilot performance affected due to alcohol, drugs or medication
B.2 - diversion by ATCo by mistake	
H056	Poor coordination between civil and military ATC
H057	Poor coordination between ATC centres
H058	Misidentification of an aircraft by ATC
H061	Malfunctioning of ATC systems, e.g. radar
H062	Controller makes a wrong decision
H063	Controller makes a mistake in aircraft identity
H070	Controller does not know the intent of an aircraft
H071	Controller does not know aircraft position
H075	Controller is not well trained to deal with emergency situation
B.3 - diversion caused by extreme wind	
H001	Tropical storm, winter storm, tornado, cyclone
H004	Strong winds
H126	Overshoot of planned route due to wind
H130	Jet stream
B.4 - diversion caused by extreme weather	
H002	Icing, freezing precipitation, snow
H003	Heavy rain
H005	Thunderstorms
H007	Fog
H008	Dust or sand storms
H009	Lightning
H122	Sudden weather change disturbs planning
H129	Winter conditions at airport
H132	Significant temperature inversions
B.5 - diversion caused by natural phenomena	
H010	Volcanic eruption
H011	Geophysical event on the ground, e.g. earthquake or tsunami
H012	Space weather (e.g. solar activity variations) affecting satellite communication or navigation

B.6 - navigational system failure	
H042	Error in routing of flight management system, e.g. wrong waypoints in database, or outdated FMS plan
H047	Problem with the positioning system, e.g. failure of GPS, navigation error in own position
B.7 - airspace restrictions	
H018	Airspace closure
H030	En-route deviation
H053	Restricted airspace
B.8 - long taxiing	
H027	Longer taxi time than planned
H028	Taxi and ground delay
H051	No ATC on an airport
B.9 - delay caused by communication problem	
H059	ATIS does not provide correct information to pilots
H064	VHF R/T communication is not working or delayed
H065	Poor R/T ability or poor knowledge of English, e.g. leading to misunderstanding by ATC of fuel problem
H066	Misunderstanding in communication between controller and pilot
H067	Wrong VHF R/T frequency selected
H102	Cultural differences impact the performance of crews
B.10 - delay caused by additional holding	
H013	ATM congestion
H032	ATC flow management and aerodrome congestion
H033	Long time spent in holding
H035	Additional approaches
H069	Controller forgets aircraft
H072	Controller does not know the availability of airspace infrastructure
H114	Avoiding bad weather leads to higher traffic density
H115	High traffic density
B.11 - cannot land because of system failure	
H043	Airborne systems not working, e.g. cockpit display, flight management system, or large electronic failure
H044	Problem with instrument landing system
B.12 - cannot land because of unavailable runway	
H017	Runway closure
H049	Problem with approach or runway lights
H052	Runway blocked or contaminated
B.13 - missed approach	
H034	Missed approaches
H080	Pilot fails to obtain ATC authorization
H097	Pilots (intend to) use wrong runway

C1 – LOSS OF FUEL	
C1.1 - degradation of aircraft	
H046	Degradation of aircraft structure
H048	Degradation of one or multiple engines
H099	Pilot does not detect degradation of airborne system
C1.2 - fuel leakage	
H014	Mechanical failure of an aeroplane system
H139	Fuel leakage
H141	Pre-flight maintenance error
C1.3 - big amount of fuel lost, because crew did not notice the leakage	
H088	Pilot validates without actually checking, e.g. fuel load
H090	Alert causes attention tunnelling by pilots
H092	Risk of fuel problem is underestimated by pilots
H093	Pilots receive wrong information about fuel quantity
H094	Pilots misinterpret information about fuel quantity
H100	Delay into detection of a problem by pilots due to lack of trust in technical system
H101	Over-reliance of pilots on wrong system data
H103	Lack of situation awareness of pilot due to high level of automation
H105	Airline with poor safety culture
H107	Large workload of crew
H136	Fuel quantity indicator is malfunctioning
H137	Pilots do not check fuel quantity
H140	Crew does not follow the applicable procedures correctly
C2 - INABILITY TO USE FUEL	
C2.1 - failure of fuel system	
H014	Mechanical failure of an aeroplane system
H138	Failure in the fuel system such that part of the fuel cannot be used
H141	Pre-flight maintenance error
H142	Fuel management not working properly, e.g. automatic transfer of fuel
H144	Fuel imbalance
H146	Electrical failure
C2.2 - degradation resulting in failure of fuel system	
H046	Degradation of aircraft structure
H048	Degradation of one or multiple engines
H099	Pilot does not detect degradation of airborne system
C2.3 - accumulation of ice in fuel system	
H002	Icing, freezing precipitation, snow
H118	Weather influences the functioning of airborne systems
H145	Fuel freezing
C3 - LOW FUEL INTAKE BEFORE FLIGHT	
C3.1 - inaccurate weather forecast	
H025	Hazards affecting meteorological reporting or forecasting

H118	Weather influences the functioning of airborne systems
H121	Weather forecast wrong
H122	Sudden weather change disturbs planning
H124	Weather info not available
H126	Overshoot of planned route due to wind
H150	Malfunctioning of AOC systems
C3.2 - inaccurate planning by pilot or AOC	
H036	Insufficient aircraft type specific fuel planning experience of flight crew
H037	Flight crew unfamiliar with route
H088	Pilot validates without actually checking, e.g. fuel load
H089	Pilot makes an error in the calculation of the aircraft performance, e.g. aircraft weight, fuel quantity
H092	Risk of fuel problem is underestimated by pilots
H106	Pilot insufficiently trained for dealing with fuel management
H135	Pilots plan a nearby alternate destination, which is in practice not a feasible option (e.g. for political reasons)
H136	Fuel quantity indicator is malfunctioning
H137	Pilots do not check fuel quantity
H140	Crew does not follow the applicable procedures correctly
C3.3 - not accounting for higher consumption caused by aircraft degradation	
H046	Degradation of aircraft structure
H048	Degradation of one or multiple engines
H099	Pilot does not detect degradation of airborne system
C3.4 - improper use of new equipment	
H021	Operational changes, e.g. new equipment, adapted procedures
C3.5 - pilot pressed by management to reduce costs	
H134	Pilots feel pressed by management to reduce fuel intake
D1 - RESOLUTION HAZARDS	
D1.1 - aircraft	
H038	Route near maximum range of aeroplane
H041	Aircraft not equipped with technical system, e.g. auto-landing system
H078	Aircraft flies near its envelope extremes
H079	Aircraft is in a wrong mode for a particular action
H086	In an emergency procedure, aircraft may have to descend quickly and not have time to look out for other traffic
H143	Inadequate certification requirements
D1.2 - flight crew	
H081	Pilot is not following the clearance because he tries to solve a problem
H082	Cockpit crew disagreement
H085	Pilot does not know when to take action
H090	Alert causes attention tunnelling by pilots
H091	Difference in situation awareness of Pilot Flying and Pilot Not Flying

H092	Risk of fuel problem is underestimated by pilots
H096	Procedures and routes in TMA or at airport are not well known by pilots (e.g. because pilots enter it seldom)
H098	Aircrew unaware of loss of voice communication
H100	Delay into detection of a problem by pilots due to lack of trust in technical system
H101	Over-reliance of pilots on wrong system data
H102	Cultural differences impact the performance of crews
H104	Pilot incapacitation, e.g. pilots falling asleep, pilots die, or pilot performance affected due to alcohol, drugs or medication
H106	Pilot insufficiently trained for dealing with fuel management
H107	Large workload of crew
H117	Avoiding bad weather leads to increase in crew workload and/or to a shift in pilot attention
H140	Crew does not follow the applicable procedures correctly
D1.3 - ATC and AOC	
H051	No ATC on an airport
H055	Controller does not inform other controllers about an emergency situation
H056	Poor coordination between civil and military ATC
H057	Poor coordination between ATC centres
H062	Controller makes a wrong decision
H073	Controller is incapacitated
H074	Insufficient capacity of an ATC centre due to strike or illness
H075	Controller is not well trained to deal with emergency situation
H076	Large workload of a controller
D1.4 - aerodrome	
H016	Isolated aerodrome
H019	Political unrest or terrorism
H054	Complex standard arrival route
D1.5 - communication	
H059	ATIS does not provide correct information to pilots
H064	VHF R/T communication is not working or delayed
H065	Poor R/T ability or poor knowledge of English, e.g. leading to misunderstanding by ATC of fuel problem
H066	Misunderstanding in communication between controller and pilot
H067	Wrong VHF R/T frequency selected
D1.6 - environment	
H015	Adverse terrain or large bodies of water along the route
D1.7 - procedures	
H105	Airline with poor safety culture
H110	Occurrence of a situation which is not procedurally covered
H111	Difficult emergency procedures, leading to incorrect or late crew actions
H112	Wrong design of procedure

D1.8 - changes in operation	
H020	Organization changes, e.g. changes to key personnel, rapid growth, rapid contraction, corporate mergers
H021	Operational changes, e.g. new equipment, adapted procedures
H109	Changes or differences in procedures lead to confusion by pilots or controllers
D2 - SITUATION AWARENESS	
D2.1 - ATC situation awareness	
H058	Misidentification of an aircraft by ATC
H063	Controller makes a mistake in aircraft identity
H068	Controller does not know whether an aircraft can fly a procedure
H069	Controller forgets aircraft
H070	Controller does not know the intent of an aircraft
H071	Controller does not know aircraft position
H116	Darkness
H123	Aircraft reacts on meteorological conditions that are not known to ATC
D2.2 - pilot situation awareness	
H039	Lack of routing accuracy of flight management system
H087	Pilots cannot explain where they are, e.g. due to lack of waypoints
H091	Difference in situation awareness of Pilot Flying and Pilot Not Flying
H093	Pilots receive wrong information about fuel quantity
H094	Pilots misinterpret information about fuel quantity
H103	Lack of situation awareness of pilot due to high level of automation
H108	A pilot may lose interest when flight information updates (e.g. ATIS) are uploaded too frequently
H116	Darkness
H120	Pilot perception of weather areas may differ from info received
D2.3 - AOC situation awareness	
H025	Hazards affecting meteorological reporting or forecasting
H116	Darkness
H124	Weather info not available
H150	Malfunctioning of AOC systems

5 Risk assessment methods

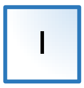
Purpose of this chapter is to introduce two key concepts that were used to assess the risk probabilities. In first section, we will introduce the dynamically coloured Petri nets which we used to construct the risk assessment model. In second section, we will illustrate the accelerating algorithm called the splitting method that was used for simulating the probabilities of the rare events.

5.1 Dynamically coloured Petri nets

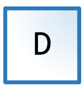
Dynamically coloured Petri nets (DCPN) have proven to be a useful tool in constructing the dynamic models with complex structure. We will not provide a complete definition of DCPN. That is written in (Everdij, 2010) together with a lot of related theoretical background. Here, we present only a short description of the main concepts of DCPN together with an example.

Basic elements of a DCPN are *places*, *tokens*, *transitions* and *arcs*. Places are connected to transitions by arcs. One arc always connects one transition with one place. A place can have a token. Token has *colour*, which is another name for a value. That means that the colour is a real vector. These values change over time according to differential equations. The differential equations are defined for each place. So the value of the token changes differently depending on a current place in which the token is. Tokens can move around the places of a Petri net using the transitions. Token can move from place *A* to place *B* only if there is a transition with incoming arc from place *A* and outgoing arc to place *B*. We say that the transition fires when it is moving the tokens. When the transition fires, it removes the token from the places that are connected to the transition by incoming arcs and puts the tokens to the places that are connected to the transition by outgoing arcs.

We distinguish three kinds of transitions:



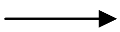
Immediate: This transition fires as soon as there is a token on each place that is connected to the transition by an incoming arc.



Delay: This transition fires under the same condition as the immediate transition, but not immediately. It fires only after time delay. This delay is defined by a random variable.

Guard: This transition requires an additional condition to be satisfied together with the same condition as in case of immediate transition. This condition is called guard condition and it is defined using the values of tokens that are on places connected to the transition by incoming arcs.

We distinguish two kinds of arcs:



Ordinary: This arc does not have any special characteristics. It is an incoming arc (or input arc) for the transition if the transition is at the end of the arc (then a place must be on the start of the arc). It is an outgoing (or output arc) for the transition is the transition is on the start of the arc (then a place must be on the end of the transition).

Enabling: This arc can be only an incoming arc for a transition. It has the same properties as the ordinary arc with the only difference being that when the transition fires, the token is not removed from the place that was connected to the transition with this arc.

We will often use the term *local Petri net*. A local Petri net (LPN) is a subset of the whole model and symbolizes a functional entity of the model. LPN is designed such that there is always one token within all the places of the LPN.

An example of a Petri net consisting of two local Petri nets is on Figure 5.1. The first LPN consists of four places (*Start*, *Main process*, *Background process* and *End*) and the second LPN consists of two places (*Working* and *Not working*). We have two tokens on places *Start* and *Working*. And we have four transitions: two delay transitions (*D1*, *D2*) and two guard transitions (*G1*, *G2*).

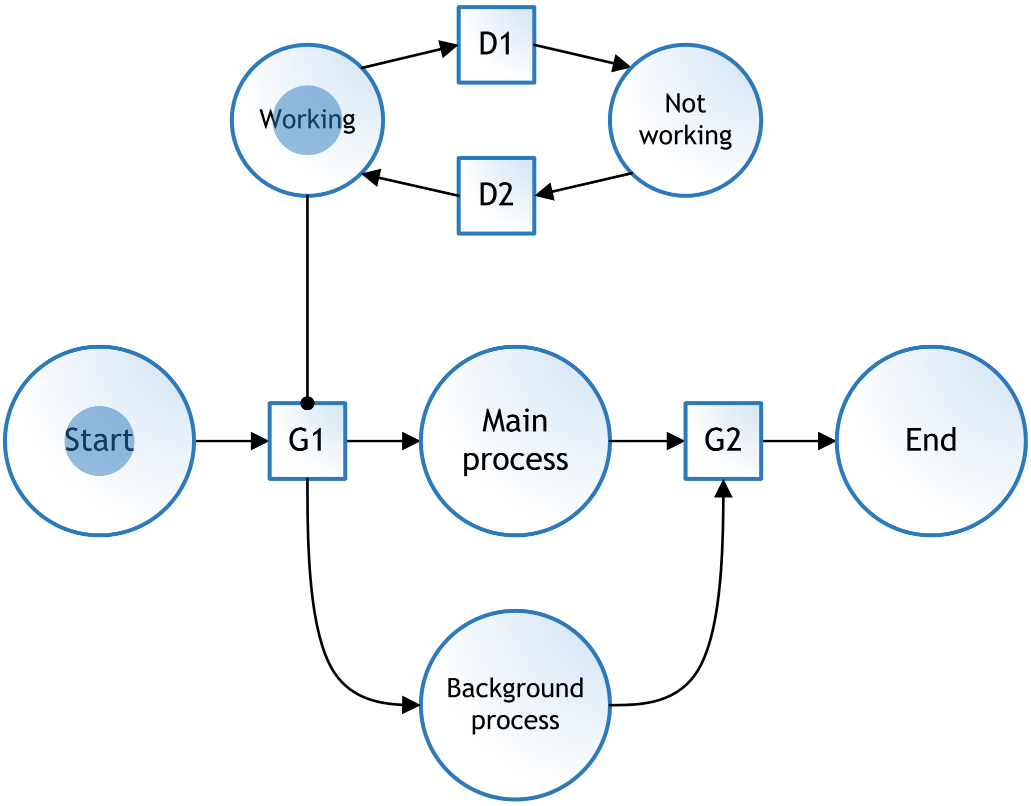


Figure 5.1: The example of a Petri net. This Petri net consists of two local Petri nets. The bottom LPN has four places (Start, Main process, Background process and End) and the second LPN has two places (Working and Not working).

Let us consider a simple example for the Petri net on Figure 5.1. This Petri net models a process of manufacturing chairs. The current state of the Petri net, as is on the figure, corresponds to the situation before the process of making chairs begin. The value of token on place *Start* determines the amount of available material that can be used for making chairs. The production of chairs begins when the transition *G1* fires. The guard condition of *G1* is satisfied if we have enough available material. Notice that *G1* can fire only if there is a token on place *Working*. There is a token now, but this token can switch between places *Working* and *Not working* with transition *D1* and *D2* which fire at random times. This secondary LPN simulates the state of machines needed to make chairs. If *D1* fires that means the machines stopped working and we cannot start the production. Assume that the token is on place *Working* and the guard condition of *G1* is satisfied. *G1* fires which means that the token from place *Start* is removed and now, we have tokens on places *Working*, *Main process* and *Background process*. The value of the token on *Main Process* specifies the amount of chairs already produced and the amount of material left that still can be used to produce new chairs. These values are changing according to differential equation defined for this place. In the *Background process*, the value of token specifies current price of the chairs. We produce new chairs until we use up all the available material. So the condition of *G2* is that the amount of remaining material reaches zero. Then, transition *G2* fires. In this transition we use the values from tokens from both *Main process* and *Background process* to calculate the profit made by selling the chairs. Now, we have tokens on places *Working* and *End*. The value of token on place *End* states the profit made by selling the chairs.

To fully define a dynamically coloured Petri net, we also need to determine following functions and sets:

- *Node function*: maps each arc to a pair of ordered nodes, where node is a place or transition.
- *Set of colour types*: set of all colour types used in the Petri net
- *Set of colour functions*: determines a colour type for each place of the Petri net
- *Initial marking measure*: probability measure that defines the initial distribution of tokens among the places of the Petri net and their colour
- *Set of transition guards*: defines the conditions for all guard transitions
- *Set of transition delay rates*: defines the random delays for all delay transitions
- *Set of firing measures*: defines for each transition which places will receive a token and what will be the values of the tokens

As is proven in (Everdij, 2010), Dynamically coloured Petri nets are equivalent to piecewise deterministic Markov processes (PDP) in a sense that, under certain conditions, there exists a one-to-one mapping from PDP to DCPN and an into mapping from DCPN to PDP. Piecewise deterministic Markov process is, vaguely speaking, a process that follows a solution of deterministic differential equation, but at some point it jumps and then follows a solution of different equation. These jumps can occur at stochastic or fixed times.

5.2 Rare event simulation

As discussed in section 2.2, we are going to estimate probabilities of events related to fuel management. One of them is the probability of fuel starvation. This event is considered as catastrophic event and it is expected that its probability is very low. According to section 2.2, the probability of a catastrophic failure event has to be lower than 10^{-9} per flight hour.

Now consider this example. If we assume that simulation of one trajectory with our model takes 0.01 second and we are estimating probability equal to 10^{-6}, the expected time to get 200 hits to the target set is more than 23 days. We are probably going to simulate probabilities of lower orders and it will most likely take more than 0.01 second to simulate one trajectory, so it is clear that we will need to use some accelerating algorithm for the simulations.

Two most common of such algorithms are:

- *Importance sampling*
- *Splitting method*

Importance sampling is very well known method to accelerate simulation in order to get the probability of a rare event. Main idea of this method is to change the underlying probability measure and run simulations using the new measure. The changed probability measure is chosen such that the targeted rare event is no longer rare under new measure. This method works very well if we know how to change the measure. Unfortunately, our model is too complex to do this. There are hundreds of variables with various interactions and it would be extremely difficult to construct the new measure.

The splitting method, on the other hand, uses the original probability measure. The idea is to prioritize certain trajectories that are in some sense closer to the rare event than other by "splitting" each of these trajectories into several independent trajectories. This way, we get more and more trajectories closer to the rare event finally hitting the rare event with significantly less total simulated trajectories. This method is suitable for our model. In following section, we are going to explain the basic idea of the Splitting method.

5.2.1 Principle of the splitting method

This section, definitions and notation are based on chapter 3.2 of (Rubino & Tuffin, 2009) and on (Amrein & Künsch, 2010). Assume that the model is described by a strong Markov process $X = \{X(t), t \geq 0\}$ with the state space S and that all trajectories of X are càdlàg (right continuous with left-hand limits). Let $B \subset S$ be some closed set that the process can enter with very small but positive probability. Our aim is to estimate the probability γ of reaching set B before stopping time T. Then

$$\gamma = \mathrm{P}(T_B \leq T),$$

where

$$T_B = \inf\{t \ge 0 : X(t) \in B\}$$
$$T = \inf\{t \ge 0 : X(t) \in D\}$$

Time T is defined as the first hitting time of some set D. In our situation, this will be defined as the time when the aircraft reaches the gate at the destination airport.

The fundamental idea of the splitting method is that there exist some intermediate sets $B_0, B_1, \ldots, B_n$ which have to be crossed by process X to get to set B and that getting from one of those sets to the next one is not a rare event. Upon visiting one of these sets, the trajectory of X is splitted into several independent trajectories. This way we favour the trajectories that get closer to B and thus gaining more positive observations with fewer simulations. The sets $B_0, B_1, \ldots, B_n$ are defined using the importance function h which satisfies $B = \{x \in S : h(x) \ge L\}$ for some level L.

With the importance function $h(x)$, we quantify how close a certain state x is to the targeted rare event. An increasing sequence of values $L_0 \le L_1 \le \ldots \le L_n = L$ defines decreasing sequence of sets

$$S \supset B_0 \supset B_1 \supset \ldots \supset B_n = B,$$

where

$$B_k = \{x \in S : h(x) \ge L_k\} \quad \text{for } k = 0, 1, \ldots, n.$$

Similarly, we can define a decreasing sequence of events

$$A_0 \supset A_1 \supset \ldots \supset A_n,$$

where

$$A_k = \{T_k \le T\}$$
$$T_k = \inf\{t \ge 0 : X(t) \in B_k\} \qquad \text{for } k = 0, 1, \ldots, n.$$

Using this notation, we can express the wanted probability using the conditional probabilities as follows.

$$\gamma = \mathrm{P}(T_B \le T) = \mathrm{P}(T_n \le T) = \mathrm{P}(A_n) = \mathrm{P}\left(\bigcap_{k=0}^{n} A_k\right) =$$
$$= \mathrm{P}(A_n \mid A_{n-1}) \cdot \mathrm{P}(A_{n-1} \mid A_{n-2}) \cdot \ldots \cdot \mathrm{P}(A_1 \mid A_0) \cdot \mathrm{P}(A_0)$$

The levels $L_0, L_1, \ldots, L_n$ should be defined in a way that the conditional probabilities are not too small to be estimated by regular Monte Carlo simulations.

5.2.2 Description of the splitting method algorithm

Assume that we have defined the importance function h, levels $L_0 \le L_1 \le \ldots \le L_n$ and thus also sets $B_0 \supset B_1 \supset \ldots \supset B_n$ and events $A_0 \supset A_1 \supset \ldots \supset A_n$.

We denote G_k as the distribution of values and times (t_k, x_k) of process X conditional on A_{k-1} for $k = 1, \ldots, n$. We call it the entrance distribution. Following is the algorithm to get estimate $\hat{\gamma}$ of probability γ.

1. We simulate N_0 independent trajectories of process X. Each trajectory is simulated until it hits set B_0 or until stopping time T. We denote the number of trajectories that reached B_0 as R_0. Now we have a sample $\left\{\left(t_0^1, x_0^1\right), \dots, \left(t_0^{R_0}, x_0^{R_0}\right)\right\}$ from distribution G_0. We denote this empirical distribution as $\hat{G}_0$.
2. At step k, we already have empirical distribution $\hat{G}_{k-1}$ from previous step for $k = 1, \dots, n$. Using $\hat{G}_{k-1}$, we resample N_k independent trajectories of process X. The trajectories are simulated until they hit set B_k or until stopping time T. We get sample of R_k values $\left\{\left(t_k^1, x_k^1\right), \dots, \left(t_k^{R_k}, x_k^{R_k}\right)\right\}$ that create the empirical distribution $\hat{G}_k$.
3. The estimates for conditional transition probabilities are

$$\hat{p}_k = \frac{R_k}{N_k},$$

where $k = 0, \dots, n$. The estimate for probability of the rare event is then

$$\hat{\gamma} = \prod_{k=0}^{n} \hat{p}_k .$$

There are several variations for this algorithm that provide different ways how to determine number of simulated trajectories N_k and how to resample values from $\hat{G}_k$. We will mention three of those variations.

5.2.2.1 Fixed splitting

Each of the R_k entrance values is resampled exactly $c_k \geq 1$ times. Then $N_k = c_k R_k$ and both N_k and R_{k+1} are random. Advantage of this approach is that we do not need to store all the values of $\hat{G}_k$ during the whole simulation, because we can simulate each complete trajectory separately.

5.2.2.2 Fixed effort

Values N_k are fixed and we resample using random sampling with replacement. That means that we independently draw N_k values from $\hat{G}_{k-1}$. Then values R_k are random values. In this approach, we control the computation time of the whole simulation.

5.2.2.3 Fixed number of successes

Now, we fix values R_k. That means that we resample values from $\hat{G}_{k-1}$ until we have simulated R_k trajectories that hit set B_k. Then values N_k are random value. In this approach, we control the accuracy of the result.

6 Model design

In this chapter, we will describe the developed model. The technique used to create this model is called the *Agent-based dynamic risk modelling*. This technique is illustrated in a white paper (Safety, Eurocontrol/FAA AP15, 2014). The model is considered dynamic because the events in the model take place at different (often stochastic) times and they can happen at various order. In contrast to this are static techniques such as event tree modelling. Agent is an autonomous entity within the model that interacts with other agents and exchanges information. It depends on the situation what is and what is not considered as an agent. The agents defined in current version of the model are introduced in following sections. The whole model is constructed as a dynamically coloured Petri net and each agent is modelled by one or more local Petri nets (LPN).

6.1 Overview of the agents

The agents included in the model are:

1. *Environment* (EN)
2. *Airports* (AP)
3. *Airline Operations Control* (AO)
4. *Aircraft* (AC)
5. *Crew of the aircrafts* (CR)

Next, we provide a short description of each of the agent and its interactions with other agent. These interactions are illustrated also in Figure 6.1. Here, we also list all the important assumptions made about the agents in this model.

6.1.1 Environment overview and assumptions

The environment describes the situation of the airspace sectors (whether they are available for aircraft or not) and the speed of wind in these sectors. The speed of wind has a big impact on the fuel consumption and the ground speed of the aircraft. AOC considers the airspace conditions during flight planning. Crew considers the airspace conditions as part of in-flight fuel management and it is an important part of their Situation Awareness.

6.1.1.1 Assumptions (EN)

- At the start of the simulation, wind speed and direction is the same for all the airspace sectors. After that these values follow a random walk independently for each sector.
- Availability of each sector is independent of all other sectors.
- Direction of wind in a sector is independent of the altitude.
- The density of air decreases with increasing altitude according to the International Standard Atmosphere defined in (Eurocontrol Experimental Centre, 2011).

- Wind speed increases with increasing altitude according to a simplified version of a model defined in chapter 6 of (Stroeve & Bakker, 2007).
- Wind speed and direction and sector availability changes at random times determined by exponential random variables.

6.1.2 Airports overview and assumptions

Each airport has its specific characteristics like taxiing time, holding time or holding altitude. These characteristics and the location of all the available airports are known by AOC, aircraft and the crew.

6.1.2.1 Assumptions (AP)

- Each airport is located in the centre of a sector.
- The decision whether an aircraft will be sent to hold before landing is decided in airport agent. The probability of deciding to send the aircraft to holding at airport i is equal to fixed probability p_{hold}^{i}.

6.1.3 Airline Operations Control overview and assumptions

AOC dispatcher is responsible for constructing the flight plan. That includes making predictions about the weather and airspace availability, planning the flight trajectory and calculating the recommended fuel intake. For this, the dispatcher uses information about the Environment, Airports and the Aircraft. After making the flight plan, he sends it to the Crew of the aircraft.

6.1.3.1 Assumptions (AO)

- AOC has the information about the availability of all sectors and the direction of the wind.
- Wind speed is known by AOC only with a random normal error $\varepsilon \sim N(0, \sigma_{wAO})$.
- The calculated waypoints are always located in the centre of a sector.
- The alternate airport is defined before the start of the simulation as a parameter of the model. It is not decided by AOC.

6.1.4 Aircraft overview and assumptions

This agent is divided into three parts (three Local Petri Nets): *Characteristics* (AC_CH), *Fuel system* (AC_FS) and the *Evolution* (AC_EV). By Characteristics, we mean all constant parameters of the aircraft like zero fuel weight, nominal true air speed at specific height or fuel consumption parameters. This information is used by AOC during planning and by crew during in-flight management. The Fuel system simulates fuel flow and the amount of fuel left in the tanks. Fuel flow is the rate at which the fuel decreases. It depends on flight mode, speed, altitude, weight of the aircraft and head wind.

Evolution is the crucial part of the model. It simulates the whole flight of the aircraft. Evolution is controlled by the Crew and it is affected by all other agents.

6.1.4.1 Assumptions (AC)

- Fuel consumption model and climbing and descending trajectory profiles are determined by a model from Base of Aircraft Data (Eurocontrol Experimental Centre, 2011).
- During the cruising part of the flight, the aircraft flies at constant cruising altitude determined by the AOC.
- Holding pattern is simplified to a line segment at a fixed altitude h_{hold}^{i}.
- If a missed approach is performed, the aircraft starts climbing just before landing at fixed altitude h_{MA}^{i}. It climbs until it reaches h_{hold}^{i} and then descents back to land.
- When the aircraft reaches a sector containing the next waypoint (in the centre of the sector), it considers the waypoint as reached and changes the direction toward the next waypoint without flying to the centre of the sector.

6.1.5 Crew overview and assumptions

This agent is divided into two parts (two Local Petri Nets): *Planning* (CR_PL) and *Situation awareness* (CR_SA). Before the flight, the crew of the aircraft receives the flight plan from the AOC. They can accept the plan or they can request modifications. The only modification we considered at this point is that they can add extra fuel (discretionary fuel). After that, the total amount is fuelled into the Aircraft. During the flight, crew is responsible for controlling the Aircraft. For this they use information from all agents.

6.1.5.1 Assumptions (CR)

- The probability of asking for extra fuel by the Crew is equal to fixed probability p_{fEX}.
- Crew has the information about the availability of all sectors and the direction of the wind.
- Wind speed is known by AOC only with a random normal error $\varepsilon \sim N\left(0, \sigma_{wCR}\right)$.
- The calculated waypoints are always located in the centre of a sector.
- The decision whether an aircraft will perform a missed approach is made every time the aircraft starts to descent above the airport. The probability of performing a missed approach is equal to fixed probability p_{MA}.
- When the crew is making a decision to divert to a different airport, they follow these rules:
  - If the aircraft was currently flying toward the original destination airport, they first consider the alternate airport. If that is not available, they divert to the closest airport at the moment.
  - If the aircraft was currently flying toward the alternate destination airport, they first consider the original destination airport. If that is not available, they divert to the closest airport at the moment.

- Every time the crew recalculates the route and checks the fuel requirements, they first update their information about the airspace. This happens at random times determined by exponential random variable.

6.1.6 Notes about LPN definitions

- Variables not mentioned in the Token colour functions are considered to remain constant.
- Variable t represents time. This variable is used in all LPNs. It increases with constant rate from the initiation of the model.
- In section 5.1, we described local Petri net (LPN) as a Petri net that always has one token among all its places. By this definition, Petri nets AC_FS (section 6.5.2) and CR_PL (section 6.6.1) defined later in this chapter are not LPNs. A token is either in place AC_FS_P1 or place CR_PL_P1. For the sake of maintaining logical structure of the model (AC_FS is the fuel system of the aircraft and CR_PL is part of the crew Petri net), we do not merge them into one LPN and we are going to refer to them as separate LPNs in this chapter.
- In some LPNs, we indicate that a real function is part of the colour type. Since colour type is by definition a real vector, we will always assume the function is expressed by a matrix with two columns. In the first column, we specify the domain by a number of points and in the second, we specify the corresponding values of the function. Values of function at points different than specified by the matrix are computed by linear interpolation. It means that all functions in the colour types are linear splines. This is also how the true air speed at different heights is specified in Base of Aircraft Data (Eurocontrol Experimental Centre, 2011) which is the main source of aircraft parameters in the model.

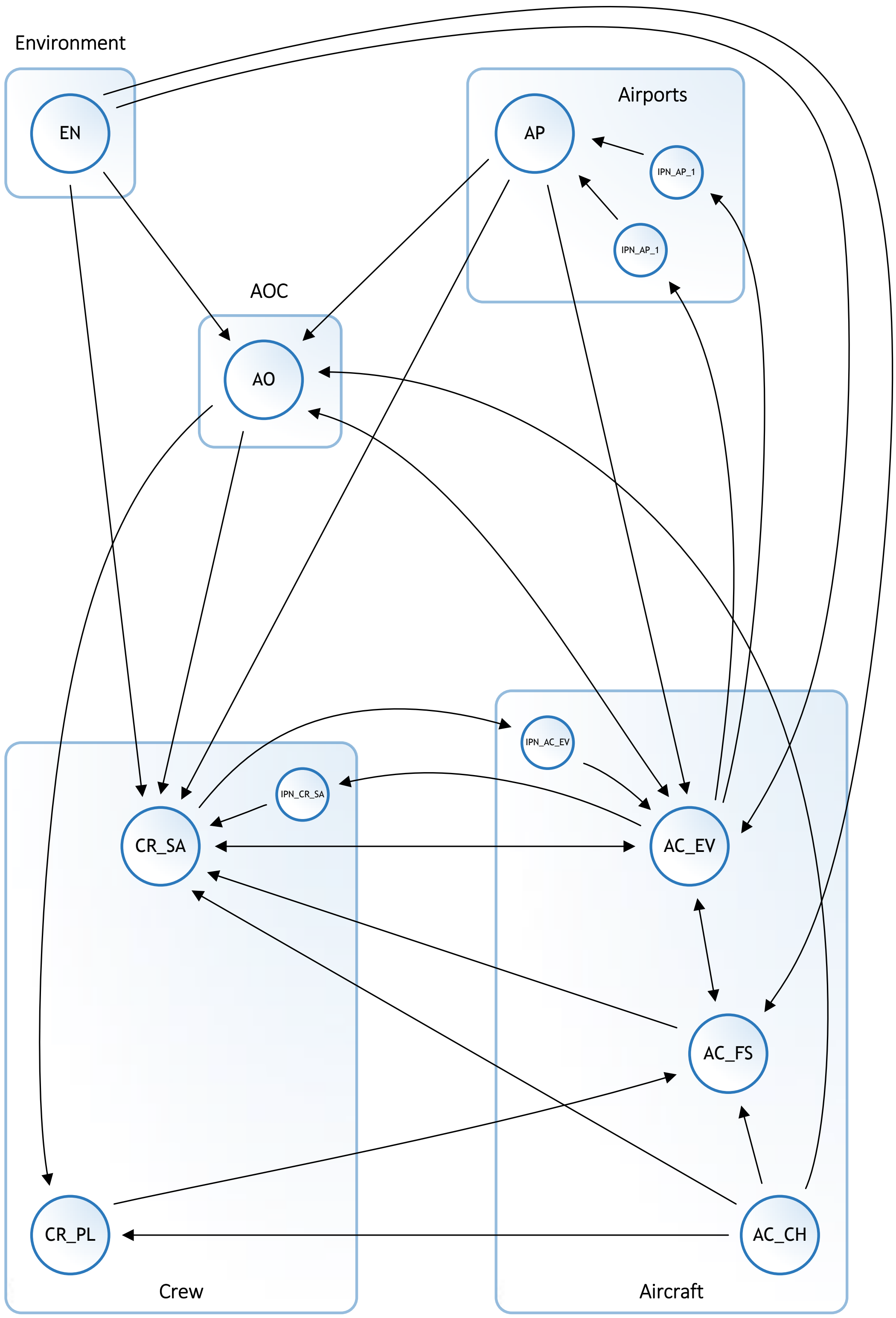


Figure 6.1: The illustration of the interactions between agents and local Petri nets. The arrows mark the flow of the information in the model.

6.2 Environment agent

In this agent, we keep information about the wind and the availability of the airspace. We divide the airspace into $N \times M$ sectors. The wind speed is initially generated for all sectors from truncated normal distribution and wind direction is generated from uniform distribution. After that, the wind speed and wind direction will both follow a random walk in each sector independently. This is regarding the wind speed at reference altitude h_{ref}. Wind speed at higher altitudes will be computed using simplified wind model from (Stroeve & Bakker, 2007). In this model, the atmosphere is divided into three layers: the Prandtl layer, Ekman layer and free atmosphere. In the first two layers, the wind speed rises as the altitude rises. In free atmosphere the wind speed is constant with regard to altitude. Each sector will have a Boolean variable that determines whether the sector is available (aircraft can fly through it) or closed (aircraft cannot fly through it and have to divert). The transition probabilities p_{sec}^{A}, p_{sec}^{C} determine the probability of sector being available if it was closed before and vice versa.

6.2.1 Environment LPN (EN)

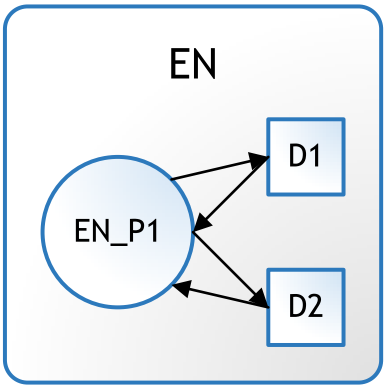


Figure 6.2: local Petri net of environment

6.2.1.1 Colour type: EN

Notation	State space	Description
w_{ref}^{nm}	$\mathbb{R}_+$	wind speed at reference altitude in sector n, m
φ_w^{nm}	$[0, 2\pi)$	direction of the wind in sector n, m
A^{nm}	$\{true, false\}$	availability of the sector n, m
Derived variables		
$w^{nm}(h) = \begin{pmatrix} w_x^{nm}(h) \\ w_y^{nm}(h) \end{pmatrix}$	$\mathbb{R}^2$	wind speed at altitude h in sector n, m
$\rho(h)$	$\mathbb{R}$	air density at altitude h
Indices follow these sets: $n = 0, \ldots, N-1$ and $m = 0, \ldots, M-1$		

6.2.1.2 Definition of derived variables from Colour type:

Notation	Definition
$w^{nm}(h)=\begin{pmatrix} w_x^{nm}(h) \\ w_y^{nm}(h) \end{pmatrix}$	if $h \le h_P$ $w^{nm}(h)=w_{ref}^{nm}\cdot\frac{\ln(1+C_{w1}h)}{\ln(1+C_{w1}h_{ref})}\cdot\begin{pmatrix}\cos\varphi_w^{nm}\\ \sin\varphi_w^{nm}\end{pmatrix}$ if $h_P < h \le h_E$ $w^{nm}(h)=w^{nm}(h_P)\cdot\left[1+(C_{w2}-1)\cdot\left(1-e^{-\frac{h-h_P}{D_E}}\right)\right]$ if $h > h_E$ $w^{nm}(h)=C_{w2}\cdot w^{nm}(h_P)$ where $h_E=C_{w3}\cdot\frac{2\alpha}{(1-\alpha)^2}\cdot\frac{w_{ref}^{nm}}{\ln(1+C_{w1}h_{ref})}$ $h_P=\alpha h_E$ $D_E=\sqrt{C_{w4}h_P\cdot\frac{w_{ref}^{nm}}{\ln(1+C_{w1}h_{ref})}}$
$\rho(h)$	$\rho(h)=\frac{p(h)}{R\tau(h)}$ where $\tau(h)=\begin{cases}\tau_0+\beta h & \text{if } h\le h_{trop}\\ \tau_0+\beta h_{trop} & \text{if } h>h_{trop}\end{cases}$ $\tau_{trop}=\tau(h_{trop})$ $\tau(h)=\begin{cases}p_0\cdot\left(\frac{\tau(h)}{\tau_0}\right)^{-\frac{g}{R\beta}} & \text{if } h\le h_{trop}\\ p_{trop}\cdot e^{\left[\frac{g}{R\tau_{trop}}\cdot(h_{trop}-h)\right]} & \text{if } h>h_{trop}\end{cases}$ $p_{trop}=p(h_{trop})$

6.2.1.3 Colour function:

Place	Colour type	Colour function
EN_P1	EN	constant

6.2.1.4 Initial marking:

Place	Initial colour
EN_P1	a token with colour EN: $w_{ref}\sim N_t(\mu_w,\sigma_w^2,w_{min},w_{max})$ $\varphi_w\sim Un(0,2\pi)$ $\forall n=0,\ldots,N-1, \quad \forall m=0,\ldots,M-1:$ $w_{ref}^{nm}(0)=w_{ref}$ $\varphi_w^{nm}(0)=\varphi_w$

	$A^{nm}(0)=\begin{cases} false & \text{with probability } \left(1+\frac{p_{sec}^{A}}{p_{sec}^{C}}\right)^{-1} \\ true & \text{with probability } 1-\left(1+\frac{p_{sec}^{A}}{p_{sec}^{C}}\right)^{-1} \end{cases}$

6.2.1.5 Delay transitions:

Transition	Delay rate	Firing function
EN_D1: EN_P1 → EN_P1	delay ~ $\mathrm{Exp}(\Delta t_w)$	a token with colour EN: $\forall n=0,\dots,N-1, \quad \forall m=0,\dots,M-1:$ $w_{ref}^{nm}(t)=w_{ref}^{nm}(t)\cdot k_w^{nm}$ $\varphi_w^{nm}(t)=\varphi_w^{nm}(t)+\Delta\varphi_w^{nm}$ where $k_w^{nm}\sim N_t(1,\sigma_k^2,k_{min},k_{max})$ $\Delta\varphi_w^{nm}\sim N(0,\sigma_\varphi^2)$
EN_D2: EN_P1 → EN_P1	delay ~ $\mathrm{Exp}(\Delta t_A)$	a token with colour EN: $\forall n=0,\dots,N-1, \quad \forall m=0,\dots,M-1:$ if $A^{nm}(t)=true$ $A^{nm}(t)=\begin{cases} false & \text{with probability } p_{sec}^{C} \\ true & \text{with probability } 1-p_{sec}^{C} \end{cases}$ if $A^{nm}(t)=false$ $A^{nm}(t)=\begin{cases} true & \text{with probability } p_{sec}^{A} \\ false & \text{with probability } 1-p_{sec}^{A} \end{cases}$

6.2.1.6 Parameters:

Parameters	Description	Value	Unit
N, M	number of rows and columns of square sectors	$40, 320$	dimensionless
Δt_w	mean of exponentially distributed time step in modelling wind speed	600	s
Δt_A	mean of exponentially distributed time step in modelling sector availability	900	s
p_{sec}^{A}, p_{sec}^{C}	transition probabilities for sector availability	$0.1, 0.001$	dimensionless
α	quotient of height of Prandtl layer and Ekman layer	0.025	dimensionless
h_{ref}	reference altitude	10	m
C_{w1}	wind coefficient	$\frac{100}{3}$	m^{-1}
C_{w2}	wind coefficient	1.6732	dimensionless
C_{w3}	wind coefficient	12302	s
C_{w4}	wind coefficient	840.5	s
μ_w	mean of initial reference wind speed	5	$m\cdot s^{-1}$
σ_w	standard deviation of initial reference wind speed	3	$m\cdot s^{-1}$

w_{min}, w_{max}	boundaries of the initial reference wind speed	$0, 20$	$m \cdot s^{-1}$
σ_k	standard deviation of coefficient k_w^{nm}	0.1	$m \cdot s^{-1}$
k_{min}, k_{max}	boundaries of coefficient k_w^{nm}	$0.8, 1.2$	$m \cdot s^{-1}$
σ_φ	standard deviation of coefficient $\Delta\varphi_w^{nm}$	$\frac{\pi}{32}$	$m \cdot s^{-1}$
h_{trop}	altitude at which the tropopause begins	11000	m
τ_0	temperature at sea level	288.15	K
β	temperature gradient below tropopause	-0.0065	$K \cdot m^{-1}$
p_0	pressure at sea level	$101\,325$	Pa
g	gravitational acceleration	9.80665	$m \cdot s^{-2}$
R	real gas constant for air	287.05287	$m^2 \cdot K^{-1} \cdot s^{-2}$

6.2.1.7 Incoming arcs from different LPNs:

There are no incoming arcs from different LPNs to EN.

6.2.1.8 Outgoing arcs to different LPNs

There are outgoing arcs to AO, AC_FS, AC_EV and CR_SA:

- AOC uses the initial airspace state to make predictions and construct the flight plan accordingly.
- AC_FS uses parameters and information about air density required to compute fuel flow.
- Wind speed affects the ground speed of the aircraft.
- The current state of the airspace is part of crew`s Situation Awareness and they use the information to plan the rest of the flight and estimate the amount of fuel needed for it.

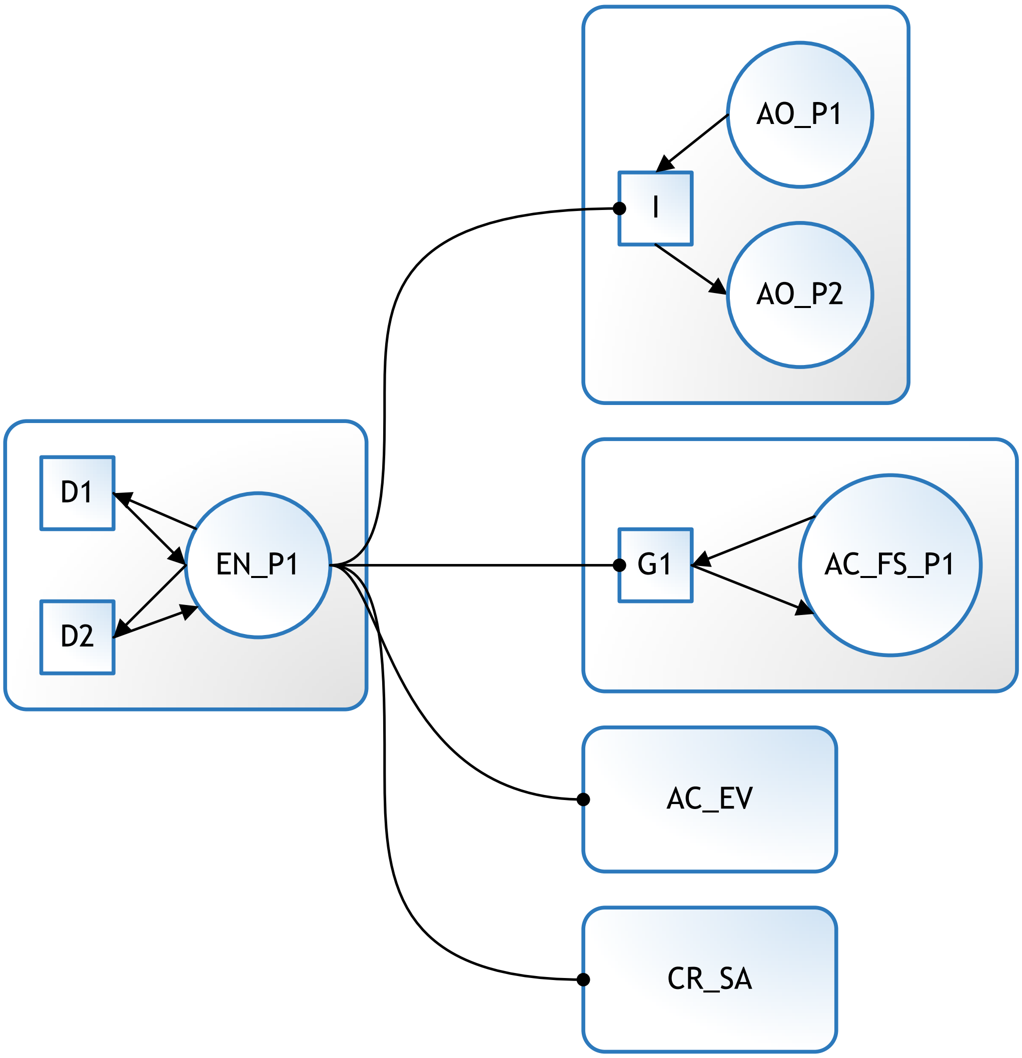


Figure 6.3: interactions between EN and other LPNs

6.3 Airports agent

We consider N_{AP} airports. Each airport is part of a specific airspace sector. For simplicity we assume the airports are in the centre of the sectors. The airport is considered unavailable if the sector where it is located is unavailable. Each airport has its parameters for operations like taxiing time, holding time and the probability of holding.

Holding time should depend on the number of aircraft that are currently in the sector of the airport. In current version of the model, we simplify this dependence and we assume that there is a parameter p^i_{hold} that is equal to probability that an aircraft will be send to holding before allowing to land at airport i .

6.3.1 Airports LPN (AP)

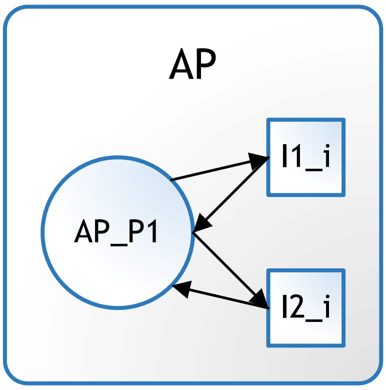


Figure 6.4: local Petri net of airports

6.3.1.1 Colour type: AP

Notation	State space	Description
n^i_A, m^i_A	$\mathbb{Z}^2$	indices of the sector in which the airport i is located
L^i_{hold}	$\{true, false\}$	determines if the aircraft will be sent to hold before landing at airport i
Derived variables		
x^i_A, y^i_A	$\mathbb{R}^2$	position of the airport i
Index follows this set: $i = 1, \ldots, N_{AP}$		

6.3.1.2 Definition of derived variables from Colour type:

Notation	Definition
x^i_A, y^i_A	$x^i_A = m^i_A d_{sec}$ $y^i_A = n^i_A d_{sec}$

6.3.1.3 Colour function:

Place	Colour type	Colour function
AP_P1	AP	constant

6.3.1.4 Initial marking:

Place	Initial colour
AP_P1	a token with colour AP: $\forall i = 1, \ldots, N_{AP}$: $\begin{pmatrix} n_A^i \\ m_A^i \end{pmatrix} = \begin{pmatrix} M_{AP}(i,1) \\ M_{AP}(i,2) \end{pmatrix}$ $L_{hold}^i = false$

6.3.1.5 Immediate transitions:

Place	Initial colour
AP_I1_i: AP_P1 ∧ IPN1_AP_i → AP_P1	a token with colour AP: $L_{hold}^i = \begin{cases} true & \text{with probability } p_{hold}^i \\ false & \text{with probability } 1 - p_{hold}^i \end{cases}$ Other variables retain their values from AP_P1.
AP_I2_i: AP_P1 ∧ IPN2_AP_i → AP_P1	a token with colour AP: $L_{hold}^i = false$ Other variables retain their values from AP_P1.

6.3.1.6 Parameters:

Parameters	Description	Value	Unit
d_{sec}	length of one side of sector	10000	m
N_{AP}	number of airports	8	dimensionless
$t_{tx}^i, \sigma_{tx}^i, t_{tx,min}^i, t_{tx,max}^i$	mean, standard deviation and boundaries of taxiing time on airport i	$480, 240, 60, 1200$	s
$t_{hold}^i, \sigma_{hold}^i, t_{hold,min}^i, t_{hold,max}^i$	mean, standard deviation and boundaries of holding time on airport i	$1200, 600, 300, 3600$	s
p_{hold}^i	probability of being send to holding before allowing to land at airport i	0.05	dimensionless
d_{hold}^i	length of holding pattern at airport i	8500	m
h_{hold}^i	altitude of holding at airport i	1000	m
h_{MA}^i	altitude at which the missed approach can be executed at airport i	100	m
M_{AP}	matrix of type $\mathbb{R}^{N_{AP} \times 2}$ where sectors of all airports are specified		dimensionless

6.3.1.7 Parameter specification:

Even though there is a possibility to have different parameters for each airport, in this version of the model, we consider general airports all having the same parameter values determining taxiing, holding and missed approach. In table below, we list the exact values of parameters n_A^i, m_A^i for each airport used in simulations.

airport i / parameter	1	2	3	4	5	6	7	8
$n_A^i = M_{AP}(i,1)$	20	20	25	25	15	20	15	20
$m_A^i = M_{AP}(i,2)$	10	310	300	50	100	150	200	250

6.3.1.8 Incoming arcs from different LPNs:

There are incoming arcs from IPN1_AP_i and IPN2_AP_i:

- The first interaction Petri net carries the information that the aircraft started to descent. This triggers the decision about holding.
- The second interaction Petri net carries the information that the aircraft has already been sent to holding. This resets the variable L_{hold}^i.

6.3.1.9 Outgoing arcs to different LPNs

There are outgoing arcs to AO, AC_EV and CR_SA:

- AOC uses the information about the airports to construct the flight plan.
- Information about the airport is used in the evolution of the aircraft taxiing or approaching the airport.
- CR_SA knows the position of the airports to make decisions during the flight.

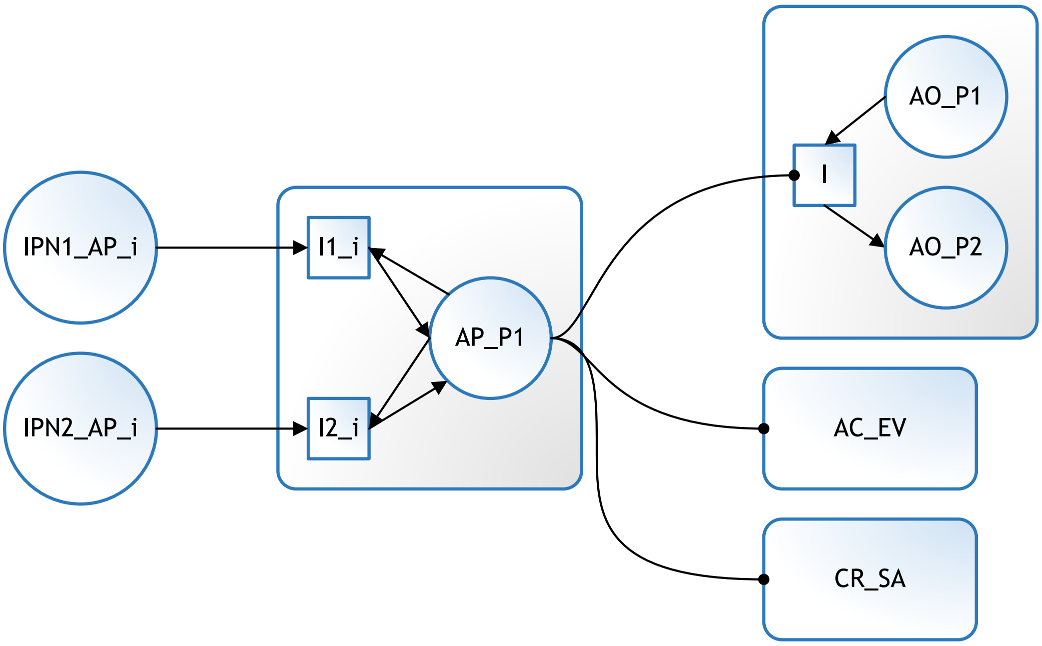


Figure 6.5: interactions between AP and other LPNs

6.3.2 Interaction Petri net (IPN1_AP_i)

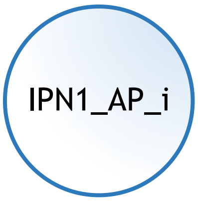


Figure 6.6: airport interaction Petri net 1

6.3.2.1 Colour function:

Place	Colour type	Colour function
IPN1_AP_i	no colour	none
Index follows this set: $i = 1, \ldots, N_{AP}$		

6.3.2.2 Initial marking:

Place	Initial colour
IPN1_AP_i	no token
Index follows this set: $i = 1, \ldots, N_{AP}$	

6.3.2.3 Incoming arcs

There are incoming arcs from AC_EV_G3, AC_EV_G4 and AC_EV_G8:

- The IPN gets the information that the aircraft started to descent.

6.3.2.4 Outgoing arcs

There are outgoing arcs to AP_I1_i:

- The airport gets the information that the aircraft started to descent.

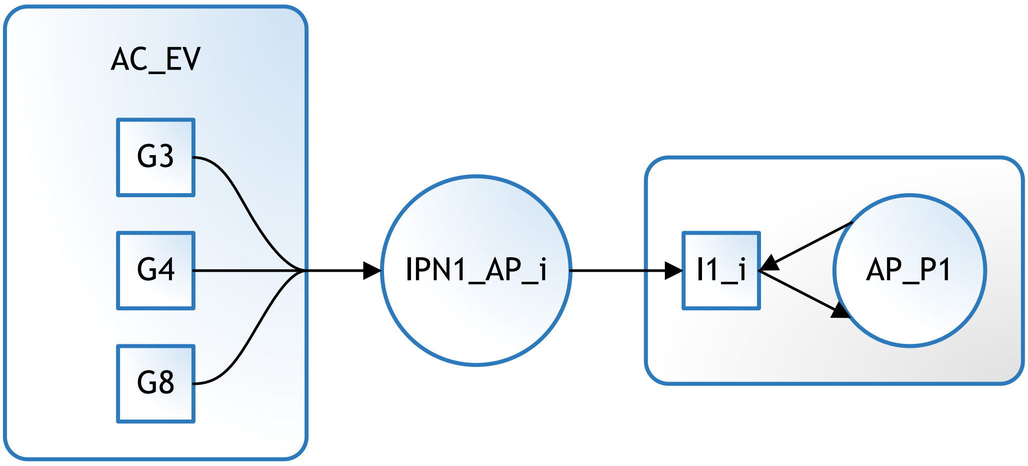


Figure 6.7: interactions between IPN1_AP_i and other LPNs

6.3.3 Interaction Petri net (IPN2_AP_i)

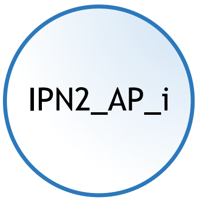


Figure 6.8: airport interaction Petri net 2

6.3.3.1 Colour function:

Place	Colour type	Colour function
IPN2_AP_i	no colour	none
Index follows this set: $i = 1, \ldots, N_{AP}$		

6.3.3.2 Initial marking:

Place	Initial colour
IPN2_AP_i	no token
Index follows this set: $i = 1, \ldots, N_{AP}$	

6.3.3.3 Incoming arcs

There is incoming arc from AC_EV_G9:

- The IPN gets the information that the aircraft has been sent to holding.

6.3.3.4 Outgoing arcs

There are outgoing arcs to AP_I1_i:

- The airport gets the information that the aircraft has been sent to holding.

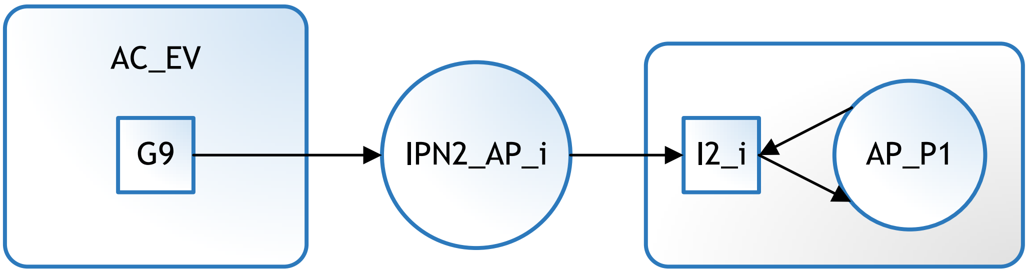


Figure 6.9: interactions between IPN2_AP_i and other LPNs

6.4 Airline operations control agent

Main function of Airline Operations Control (AOC) is to construct the flight plan (determine the waypoints and altitude) and calculate required fuel for the flight. AOC calculates total trip fuel and also planned fuel consumption along the route. This fuel plan is expressed by functions $m_{fAO,d}(d)$ and $m_{fAO,t}(t)$. The functions are expressed by matrices in the LPN as explained in 6.1.6.

To get these results, they need to make predictions about the weather and the airspace sectors availability. The calculations are described in detail in section 6.4.2.

6.4.1 Airline operations control LPN (AO)

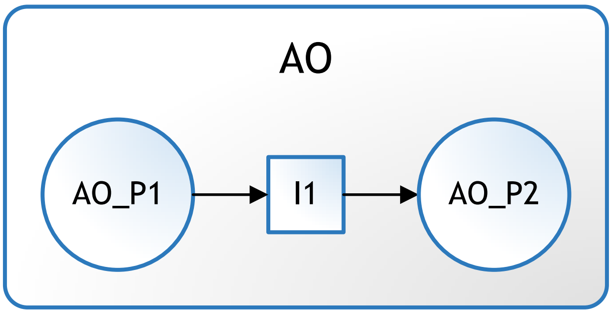


Figure 6.10: local Petri net of airline operations control

6.4.1.1 Colour type: AO

Notation	State space	Description
$W_{AO}^{j} = \begin{pmatrix} W_{AOx}^{j} \\ W_{AOy}^{j} \end{pmatrix}$	$\mathbb{R}^2$	Waypoints that form the route of the aircraft
N_{WAO}	$\mathbb{Z}$	number of waypoints
H_{crAO}	$\mathbb{R}_+$	planned cruising altitude
m_{fTX}	$\mathbb{R}_+$	planned taxi fuel
m_{fTR}	$\mathbb{R}_+$	planned trip fuel
m_{fCG}	$\mathbb{R}_+$	planned contingency fuel
m_{fAL}	$\mathbb{R}_+$	planned alternate fuel
m_{fFR}	$\mathbb{R}_+$	planned final reserve fuel
D_{total}	$\mathbb{R}_+$	planned trip distance
T_{total}	$\mathbb{R}_+$	planned trip time
$m_{fAO,d}$	$\mathbb{R}_+^{N_{fAO,t}\times 2}$	planned fuel left at specific distance from the destination
$m_{fAO,t}$	$\mathbb{R}_+^{N_{fAO,d}\times 2}$	planned fuel left at specific time of the trip
$N_{fAO,d}, N_{fAO,t}$	$\mathbb{Z}, \mathbb{Z}$	number of points at which the values of functions $m_{fAO,d}(d)$ and $m_{fAO,t}(t)$ are specified in the LPN

Derived variables		
m_{fSUM}	$\mathbb{R}_+$	planned total fuel
Index follows this set: $j = 0, \ldots, N_{W,AO} - 1$		

6.4.1.2 Definition of derived variables from Colour type:

Notation	Definition
m_{fSUM}	$m_{fSUM} = m_{fTX} + m_{fTR} + m_{fCG} + m_{fAL} + m_{fFR}$

6.4.1.3 Colour function:

Place	Colour type	Colour function
AO_P1	no colour	none
AO_P2	AO	constant

6.4.1.4 Initial marking:

Place	Initial colour
AO_P1	a token with no colour
AO_P2	no token

6.4.1.5 Immediate transitions:

Transition	Firing function
AO_I1: AO_P1 ∧ [EN_P1 ∧ AP_P1 ∧ AC_CH_P1 ∧ AC_EV_P] → AO_P2	a token with colour A0: Variables $W_{AO}^{j}, N_{WAO}, D_{total}$ are defined as described in section 6.4.2.1. Variables $H_{crAO}, m_{fTX}, m_{fTR}, m_{fCG}, m_{fAL}, m_{fFR}, T_{total}, m_{fAO,d}, m_{fAO,t},$ $N_{fAO,d}, N_{fAO,t}$ are defined as described in section 6.4.2.2.
	Used variables from other LPNs: • EN: $A^{nm}, \varphi_w^{nm}, w^{nm}(h), \rho(h)$ • AP: $t_{tx}^{I_S}, t_{tx}^{I_E}, n_A^i, m_A^i, d_{sec}$ • AC_CH: all the parameters related to fuel flow • AC_EV: $v_{cl}^{TAS}, v_{cr}^{TAS}, v_{de}^{TAS}, v_{ROC}, v_{ROD}$

6.4.1.6 Parameters:

Parameters	Description	Value	Unit
C_{H1}	optimal altitude coefficient	-0.0975	$m \cdot kg^{-1}$
C_{H2}	optimal altitude coefficient	17670	m
H_{max}	maximal cruising altitude	$12\,490$	m
h_{fAO}	altitude used in computing contingency and FRF	450	m
t_{fCG}, t_{fFR}	times used in computing contingency and FRF	$300, 1800$	s, s
I_{SAO}	index of the departure airport	1	dimensionless
I_{EAO}	index of the destination airport	2	dimensionless
I_{AAO}	index of the alternate airport	3	dimensionless

σ_{wAO}	standard deviation of error in wind speed prediction	1	$m \cdot s^{-1}$
m_{f0}	matrix of type $\mathbb{R}^{10\times 2}$ representing the function determining the initial estimate of fuel intake based on route distance		m, kg

6.4.1.7 Parameter specification:

In table below, we specify the matrix m_{f0}. Values in the first column of the matrix are different distances of the planned trip. The values in second column are the initial estimates of fuel intake based on corresponding trip distance.

trip distance $[km]$	0	100	250	500	750
fuel estimate $[kg]$	0	2227	2960	3800	4399
trip distance $[km]$	1000	2000	4000	8000	12000
fuel estimate $[kg]$	4943	7149	11764	21789	34195

6.4.1.8 Incoming arcs from different LPNs

There are incoming arcs from EN, AP, AC_CH and AC_EV:

- AOC uses the initial airspace state to make predictions about the wind and sector availability.
- AOC uses the information about the airports to construct the flight plan.
- AOC uses the information from AC_CH about the aircraft to calculate the fuel requirements and flight plan.
- AOC uses the information from AC_EV about the aircraft to calculate the fuel requirements and flight plan.

6.4.1.9 Outgoing arcs to different LPNs

There are outgoing arcs to CR_PL and CR_SA:

- AOC sends the flight plan to the flight crew before the flight.
- Flight plan is used by crew during the flight.

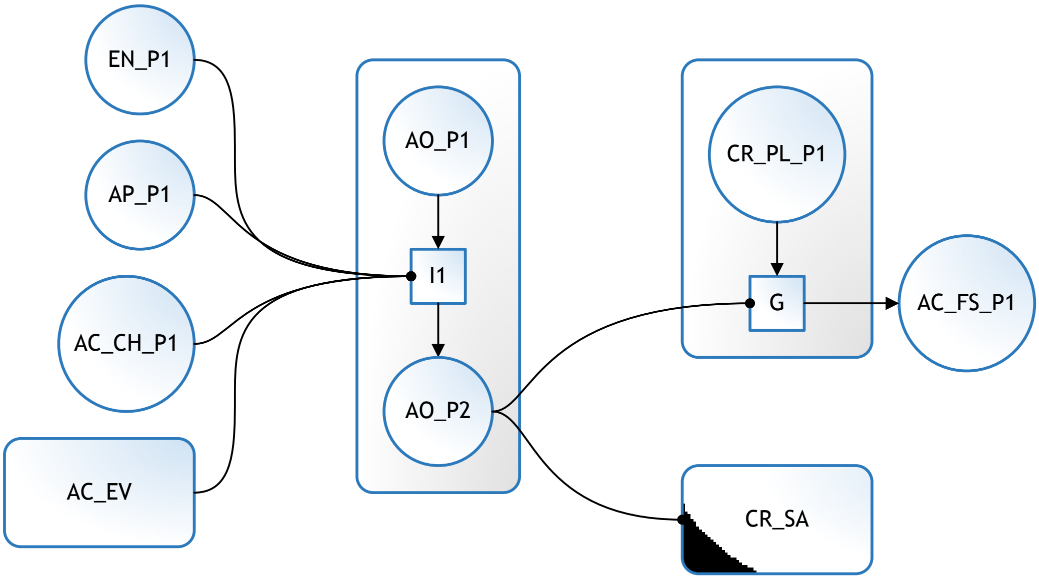


Figure 6.11: interactions between AO and other LPNs

6.4.2 Computation of the flight plan by AOC

6.4.2.1 Determining the flight route

In this section, we explain how the variables W_{AO}^{j}, N_{WAO} and D_{total} are computed in transition AO_I1 (section 6.4.1.5). We will use following variables:

- $I_{SAO}, I_{EAO}, I_{AAO}$ from AO
- A^{nm} where $n = 0, \ldots, N-1$ and $m = 0, \ldots, M-1$ from EN
- n_A^i, m_A^i, d_{sec} where $i \in \{I_{SAO}, I_{EAO}\}$ from AP

Variables W_{AO}^{j} are waypoints that, when connected, form the planned flight route. Waypoints are always in the middle of a sector. N_{WAO} is the number of all waypoints and D_{total} is the length of the planned route. Now, we will explain the algorithm, that determines the waypoints W_{AO}^{j}.

The route is constructed by several iterations. At start, we have the simplest preliminary route that consists of only two points denoted S and F connected by a straight line. We call this the initial line. These two points are the location of the departure airport S and the location of the destination airport F. We assume the sectors of these airports are open. Otherwise it would not make sense to calculate the route. Next steps of the algorithm will be explained using the example on Figure 6.12.

1. We start at the sector of departure airport (point S), which is open.
2. We check next section of the route. The checked section has the length d_{sec}.
   A. If the section is available, we move forward along the initial line to the last point that was checked and repeat step 2 of the algorithm.

B. If there is a point on the checked section that belongs to closed sector, we look for a possible diversion. We denote the sector we are currently in as A (see the figure).

3. Before finding the possible diversion, first we find the closest open sector that is on the initial line after the closed sector. We denote it B.
4. Now we look for a waypoint that, when connected to waypoints A and B, forms an available diversion. We check waypoints that are located on a line perpendicular to the initial line crossing it at the point in the middle between A and B. We check starting from the waypoints closest to the initial line alternating between the waypoints on one side of the line and the waypoints on the other side. Following the example on the figure, we first check the route through $W1$, then $W2$ and so on. The first waypoint that provides available diversion is $W6$.
5. We update the route. Before, the route was made of points $S - F$. Now we update it to $S - A - W6 - B - F$. We move to point B.
6. If we are at the destination airport (point F), we have the available flight route. Otherwise, we jump to step 2 of the algorithm and perform the next iteration.

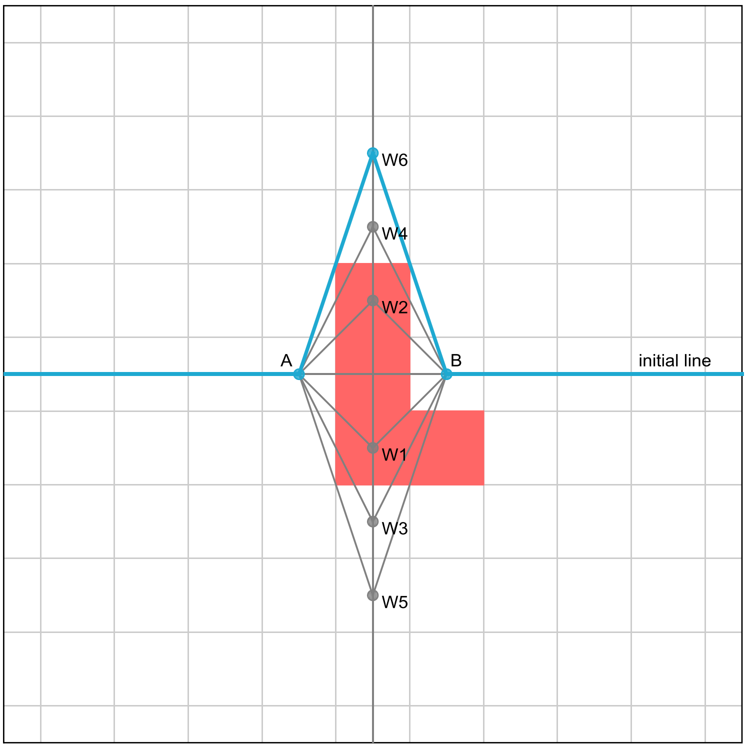


Figure 6.12: example illustrating the algorithm for finding available route, the available route is labelled by blue line

After finishing the above algorithm, we save the result to waypoints W_{AO}^{j}. Variables N_{WAO} and D_{total} are then easily computed once we have the flight route.
We use the same algorithm to also determine a route from destination airport to alternate airport. This route is not saved in any colour variable, but is necessary in order to compute alternate fuel m_{fAL}. That is described in the next section.

This algorithm is sufficient in our situation, because closed sectors are quite sparse in the airspace defined by the environment agent. Even the situation on the Figure 6.12 is very unlikely to happen. If the probability of a sector being closed would be significantly higher, we would need much more sophisticated algorithm, because this one would not be able to find any available route. On Figure 6.13, we see an example of a situation when this algorithm is not able to find available route. If this situation occurs during the flight planning, the AOC cannot find suitable flight route and the flight is cancelled. That means that this particular trajectory is omitted and we continue to simulate another flight. This imperfection is not very important, because the flights that are cancelled are not the flights that would result in especially long diversion significantly affecting the probability of reaching low fuel levels. They are just flights with a very specific configuration of closed airspace sectors as shown on Figure 6.13. Smarter algorithm could make a very short diversion in these situations.

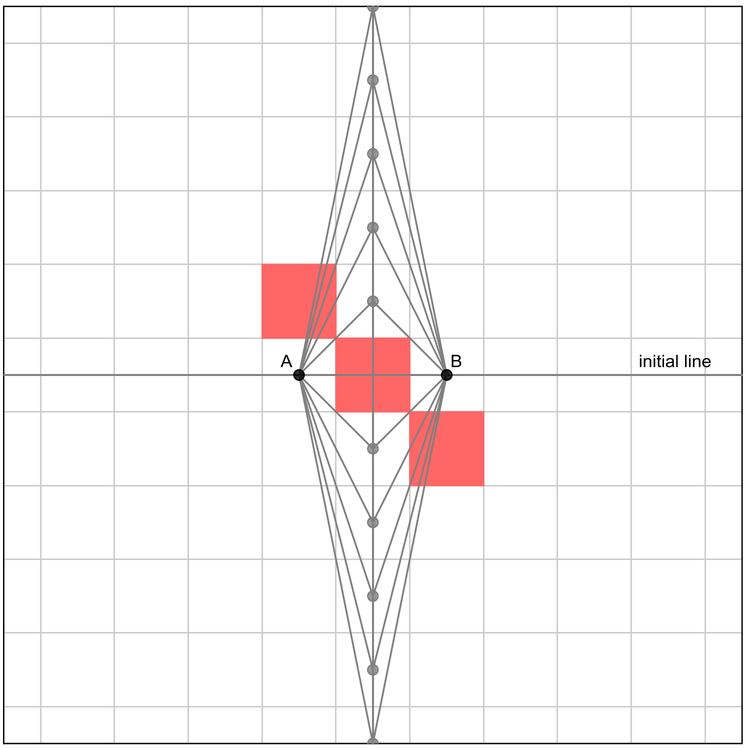


Figure 6.13: example of configuration of closed sectors that cannot be diverted by our routing algorithm

6.4.2.2 Computing planned fuel

In this section, we explain how the variables H_{crAO}, m_{fTX}, m_{fTR}, m_{fCG}, m_{fAL}, m_{fFR}, T_{total}, $m_{fAO,d}$, $m_{fAO,t}$, $N_{fAO,d}$ and $N_{fAO,t}$ are computed in transition AO_I1 (section 6.4.1.5). We will also define a function $d_{de,h}(h)$ that is used in AC_EV because it is based on very similar computations. We will use following variables:

- $W_{AO}^{j}, N_{WAO}, D_{total}, C_{H1}, C_{H2}, H_{max}, m_{f0}, I_{EAO}, I_{SAO}$ from AO
- $\varphi_{w}^{nm}, w^{nm}(h), \rho(h)$ where $n=0,\ldots,N-1$ and $m=0,\ldots,M-1$ from EN
- $t_{tx}^{I_{EAO}}, t_{tx}^{I_{SAO}}$ from AP
- $v_{cl}^{TAS}, v_{cr}^{TAS}, v_{de}^{TAS}, v_{ROC}, v_{ROD}$ from AC_EV
- all parameters and formulas regarding fuel flow from AC_CH and AC_FS

Main purpose of this section is to calculate initial fuel intake. The fuel intake determines the weight of the aircraft. But since the weight affects the fuel consumption, we first have to make a rough estimate of the weight of the aircraft. The estimate is based on the planned distance of the flight and is expressed by function m_{f0}. This function was determined empirically by running simulations of the model and observing the fuel consumption.

We denote the estimated weight of the aircraft used in this section as m.

$$m = m_0 + m_{f0} \cdot D_{total}$$

Using the estimated weight we determine the optimal cruising altitude. Following formula is from Flight Planning and Performance manual for Boeing 737-800 aircraft (The Boeing Company, 2005).

$$H_{crAO} = \min\left\{H_{max}, C_{H1} \cdot m + C_{H2}\right\}$$

In this section, we are going to assume that the aircraft only flies along a straight line. We can do this simplification because, as described in previous section, the planned route is very close to a straight line. The only difference is that there can be few of those small spikes as seen on Figure 6.12.

We calculate the direction of flight. Direction $s \in \mathbb{R}^2$ is a unit vector that states the direction from the first waypoint to the last:

$$s = \frac{W_{AO}^{N_{WAO}-1} - W_{AO}^{0}}{\left|W_{AO}^{N_{WAO}-1} - W_{AO}^{0}\right|}.$$

From the agent EN, we know functions φ_{w}^{nm} and $w^{nm}(h)$. As we can see in the definition of the initial marking of EN_P1 (see 6.2.1.4), the wind speed and wind direction is the same for all sectors at the start of the simulation. The wind speed is known by the AOC only with a random normal error $\varepsilon_w \sim N(0, \sigma_{wAO})$. So for this section, we define a function $w(h) \in \mathbb{R}^2$ that determines the wind for all sectors.

$$w(h) = \begin{pmatrix} \left(w^{00}(h) + \varepsilon_w\right) \cdot \cos\left(\varphi_w^{00}\right) \\ \left(w^{00}(h) + \varepsilon_w\right) \cdot \sin\left(\varphi_w^{00}\right) \end{pmatrix}$$

To calculate the fuel requirements we need to determine the whole trajectory of the flight together with the true air speed at each point of the trajectory. Using that we can express the fuel flow along the flight. By integrating that, we get the total fuel consumption. Now, we will describe these steps and used functions.

First, we compute function $h_{cl,t}(t)$ which determines the altitude that is climbed in time t assuming the rate of climb is given by function $v_{ROC}(h)$. It is computed as a solution of ordinary differential equation.

$$\begin{aligned}\frac{d}{dt}h_{cl,t}(t) &= v_{ROC}\left(h_{cl,t}(t)\right)\\ h_{cl,t}(0) &= 0\end{aligned}$$

Similarly, we compute function $h_{de,t}(t)$ using rate of descent given by $v_{ROD}(h)$.

$$\begin{aligned}\frac{d}{dt}h_{de,t}(t) &= v_{ROD}\left(h_{de,t}(t)\right)\\ h_{de,t}(0) &= 0\end{aligned}$$

These two functions depend only on the aircraft parameters which do not change so the functions are the same for each flight. That means in the implementation of the model, we will compute them only once and store them in a matrix as explained in section 6.1.6 for further use in all consecutive simulations.

Now, we compute functions $d_{cl,t}(t)$, $d_{cr,t}(t)$ and $d_{de,t}(t)$. These functions determine the distance travelled in time t during climb, cruise and descent respectively. Since we know the true air speed, not the ground speed, we also need to account for the wind.

$$\begin{aligned}d_{cl,t}(t) &= \int_0^t v_{cl}^{TAS}\left(h_{cl,t}(u)\right)du + \int_0^t \left|ss^{\mathsf{T}}w\left(h_{cl,t}(u)\right)\right|du\\ d_{cr,t}(t) &= \left(v_{cr}^{TAS}\left(H_{crAO}\right) + \left|ss^{\mathsf{T}}w\left(H_{crAO}\right)\right|\right)\cdot t\\ d_{de,t}(t) &= \int_0^t v_{de}^{TAS}\left(h_{de,t}(u)\right)du + \int_0^t \left|ss^{\mathsf{T}}w\left(h_{de,t}(u)\right)\right|du\end{aligned}$$

We can now compute times T_{cl}, T_{cr}, T_{de} and distances D_{cl}, D_{cr}, D_{de}. These are the times spent by climbing, cruising and descending and distances travelled during these parts of the flight.

$$\begin{aligned}T_{cl} &= \left(h_{cl,t}\right)^{-1}\left(H_{crAO}\right)\\ T_{de} &= \left(h_{de,t}\right)^{-1}\left(H_{crAO}\right)\\ D_{cl} &= d_{cl,t}\left(T_{cl}\right)\\ D_{de} &= d_{de,t}\left(T_{de}\right)\\ D_{cr} &= D_{total} - \left(D_{cl} + D_{de}\right)\end{aligned}$$

$$\begin{aligned}T_{cr} &= \frac{D_{cr}}{v_{cr}^{TAS}\left(H_{crAO}\right) + \left|ss^{\mathsf{T}}w\left(H_{crAO}\right)\right|}\\ T_{total} &= T_{cl} + T_{cr} + T_{de}\end{aligned}$$

If the route D_{total} is long enough, then $D_{cr} > 0$ and we continue to determine the trajectory along the flight. But it can be the case that D_{cr} computed by the formula above is negative. That means that the aircraft has to start descending before it reaches the planned optimal cruising altitude. This happens if the route is short or if there is strong tail wind. In this situation, we have to determine time $\tilde{T}$, when the aircraft starts descending. We do that by solving following equation for the unknown $\tilde{d}$ which then determines $\tilde{T}$.

$$h_{cl,t}\left(\left(d_{cl,t}\right)^{-1}\left(\tilde{d}\right)\right) = h_{de,t}\left(\left(d_{de,t}\right)^{-1}\left(D_{total} - \tilde{d}\right)\right)$$

$$\tilde{T} = \left(d_{cl,t}\right)^{-1}\left(\tilde{d}\right)$$

When we know $\tilde{T}$, we can continue similarly as if $D_{cr} > 0$. The only difference is that we skip the cruising part. We will now continue the computations assuming $D_{cr} > 0$.

Using the computed times and distances, we define functions $h_t(t)$, $d_t(t)$ and $v_t^{TAS}(t)$ that determine the altitude, distance travelled and true air speed of the aircraft along the whole flight.

$$h_t(t) = \begin{cases} h_{cl,t}(t) & \text{if} \quad t \in [0, T_{cl}) \\ H_{crAO} & \text{if} \quad t \in [T_{cl}, T_{cl} + T_{cr}) \\ h_{de,t}(T_{total} - t) & \text{if} \quad t \in [T_{cl} + T_{cr}, T_{total}] \end{cases}$$

$$d_t(t) = \begin{cases} d_{cl,t}(t) & \text{if} \quad t \in [0, T_{cl}) \\ D_{cl} + d_{cr,t}(t - T_{cl}) & \text{if} \quad t \in [T_{cl}, T_{cl} + T_{cr}) \\ D_{total} - d_{de,t}(T_{total} - t) & \text{if} \quad t \in [T_{cl} + T_{cr}, T_{total}] \end{cases}$$

$$v_t^{TAS}(t) = \begin{cases} v_{cl}^{TAS}\left(h_{cl,t}(t)\right) & \text{if} \quad t \in [0, T_{cl}) \\ v_{cr}^{TAS}\left(H_{crAO}\right) & \text{if} \quad t \in [T_{cl}, T_{cl} + T_{cr}) \\ v_{de}^{TAS}\left(h_{de,t}(T_{total} - t)\right) & \text{if} \quad t \in [T_{cl} + T_{cr}, T_{total}] \end{cases}$$

With the use of functions for fuel flow described in section 6.5.3, we define function $f_t(t)$ that determines the fuel flow at any time t of the flight.

$$f_t(t) = \begin{cases} f_f^{cl}\left(h_t(t), v_t^{TAS}(t)\right) & \text{if} \quad t \in [0, T_{cl}) \\ f_f^{cr}\left(h_t(t), v_t^{TAS}(t), m\right) & \text{if} \quad t \in [T_{cl}, T_{cl} + T_{cr}) \\ f_f^{de}\left(h_t(t), v_t^{TAS}(t)\right) & \text{if} \quad t \in [T_{cl} + T_{cr}, T_{total}] \end{cases}$$

Putting all together, we compute function $m_{f0,t}(t)$ that expresses the amount of fuel needed for the remainder of flight at specific time t of flight.

$$m_{f0,t}(t) = \int_t^{T_{total}} f_t(u)\,du$$

Similarly, we define function $m_{f0,d}(d)$ that expresses the amount of fuel needed for the remainder of flight at specific distance from departure airport d.

$$m_{f0,d}(d) = \int_d^{D_{total}} f_t\left(\left(d_t\right)^{-1}(v)\right)dv$$

Then trip fuel is

$$m_{fTR} = m_{f0,t}(0).$$

The alternate fuel m_{fAL} is computed by repeating the same procedure as for the trip fuel, only assuming the route from the destination to alternate airport instead of the original route.

Contingency fuel, as defined in section 3.2.1.3 is equal to

$$m_{fCG} = \max\left\{t_{fCG} \cdot f_f^{cr}\left(h_{fAO}, v_{cr}^{TAS}\left(h_{fAO}\right), m\right), 0.05 \cdot m_{fTR}\right\}.$$

Final reserve fuel is defined in section 3.2.1.5 as following

$$m_{fFR} = t_{fFR} \cdot f_f^{cr}\left(h_{fAO}, v_{cr}^{TAS}\left(h_{fAO}\right), m\right).$$

Taxi fuel is equal to

$$m_{fTX} = f_f^{taxi} \cdot \left(t_{tx}^{I_{SAO}} + t_{tx}^{I_{EAO}}\right).$$

Finally, we can compute function $m_{fAO,t}(t)$ and similarly also $m_{fAO,d}(d)$. We just have to add planned remaining fuel after the end of the flight.

$$m_{fAO,t}(t) = m_{f0,t}(t) + m_{fCG} + m_{fAL} + m_{fFR} + f_f^{taxi} \cdot t_{tx}^{I_E}$$
$$m_{fAO,d}(d) = m_{f0,d}(d) + m_{fCG} + m_{fAL} + m_{fFR} + f_f^{taxi} \cdot t_{tx}^{I_E}$$

Matrices $m_{fAO,t}$ and $m_{fAO,d}$ representing the functions are created such that they contain value corresponding to every minute of the flight in order to have sufficient precision. Values $N_{fAO,t}$ and $N_{fAO,d}$ are equal to the number of points at which the matrices specify the function values.

$$N_{fAO,t} = \left\lfloor \frac{T_{total}}{60} \right\rfloor$$
$$N_{fAO,d} = N_{fAO,t}$$

At this point, we are also going to define the function $d_{de,h}(h)$ used in AC_EV. It determines the distance that the aircraft travels before descending to ground from altitude h if we consider no wind.

$$\tilde{d}_{de,t} = \int_0^t v_{de}^{TAS}\left(h_{de,t}(u)\right) du$$
$$d_{de,h}(h) = \tilde{d}_{de,t}\left(\left(h_{de,t}\right)^{-1}(h)\right)$$

6.5 Aircraft agent

As explained in section 6.3, we assume only one aircraft in this version of the model but it can be easily extended to N_{AC} aircraft. The aircraft is assumed to be of type Boeing 737-800.

Aircraft agent consists of three local Petri nets:

- **Characteristics AC_CH:** This LPN stores all constant parameters of the aircraft.
- **Evolution AC_EV:** This local Petri net simulates the complete flight of the aircraft from gate to gate.
- **Fuel system AC_FS:** Fuel system simulates the fuel consumption. The fuel consumption model is explained in detail in section 6.5.3.

6.5.1 Aircraft characteristics LPN (AC_CH)

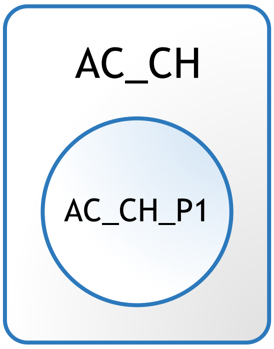


Figure 6.14: local Petri net of aircraft characteristics

6.5.1.1 Colour function:

Place	Colour type	Colour function
AC_CH_P1	no colour	none

6.5.1.2 Initial marking:

Place	Initial colour
AC_CH_P1	A token with no colour.

6.5.1.3 Parameters:

Parameters	Description	Value	Unit
m_0	zero fuel weight	41150	kg
m_{max}	maximum weight	78300	kg
C_{tx}	fuel flow during taxiing	0.1992	$kg \cdot s^{-1}$
C_{f1}	thrust specific fuel consumption parameter	$1.1676 \cdot 10^{-5}$	$kg \cdot s^{-1} \cdot N^{-1}$
C_{f2}	thrust specific fuel consumption parameter	1068.1	$m \cdot s^{-1}$
C_{f3}	idle thrust fuel flow parameter	0.2365	$kg \cdot s^{-1}$
C_{f4}	idle thrust fuel flow parameter	20096.0736	m
C_{fcr}	correction factor for fuel flow during	0.92958	dimensionless

	cruise		
C_{T1}	thrust parameter	146590	N
C_{T2}	thrust parameter	16420.1856	m
C_{T3}	thrust parameter	$3.2779 \cdot 10^{-10}$	m^{-2}
C_{Tcr}	maximum cruise thrust correction factor	0.95	dimensionless
C_{Tapp}	approach thrust correction factor	0.19448	dimensionless
C_{Tld}	landing thrust correction factor	0.3061	dimensionless
C_{D1}	drag parameter	0.025452	dimensionless
C_{D2}	drag parameter	0.035815	dimensionless
S	wing reference area	124.65	m^2
h_{app}	maximum altitude at which the formula for fuel flow during approach can be used	762	m
h_{ld}	maximum altitude at which the formula for fuel flow during landing can be used	304.8	m

6.5.1.4 Incoming arcs from different LPNs

There are no incoming arcs from different LPNs to this LPN.

6.5.1.5 Outgoing arcs to different LPNs

There are outgoing arcs to AO, AC_FS, CR_PL and CR_SA:

- AOC uses the information to construct the flight plan.
- Fuel system uses the parameters to calculate fuel flow.
- Crew (CR_PL) uses the information about the aircraft to determine initial fuel intake
- Crew (CR_SA) uses the information about the aircraft to make decisions along the flight.

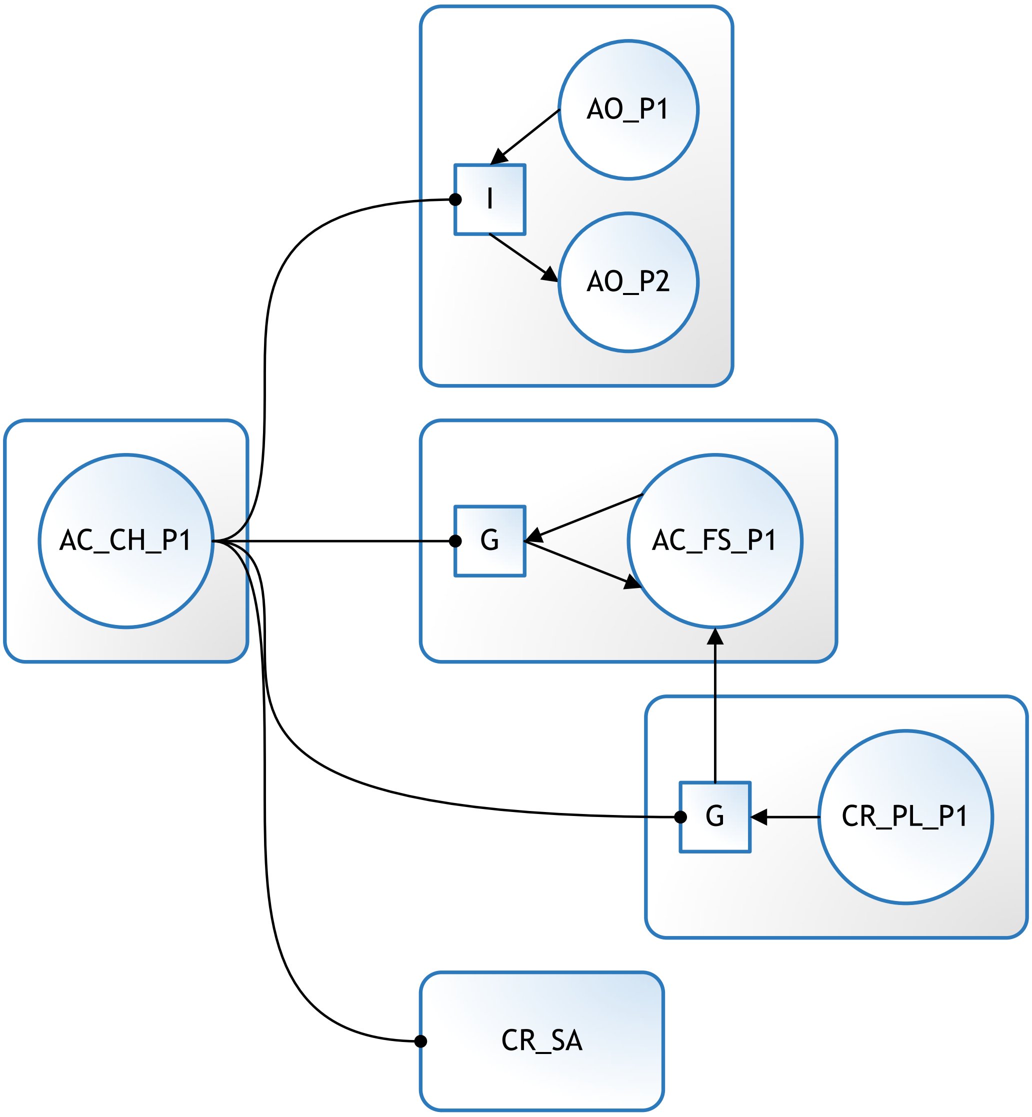


Figure 6.15: interactions between AC_CH and other LPNs

6.5.2 Aircraft fuel system LPN (AC_FS)

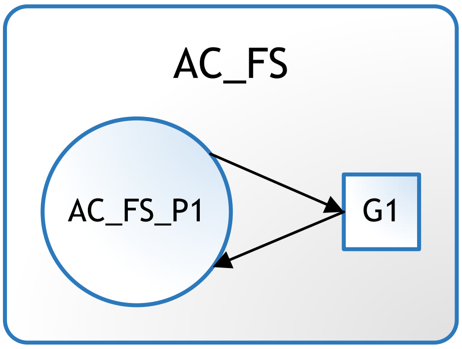


Figure 6.16: local Petri net of aircraft fuel system

6.5.2.1 Colour type: FS

Notation	State space	Description
f_f	$\mathbb{R}_+$	current fuel flow
m_f	$\mathbb{R}_+$	amount of fuel left in tanks
$t_{timerFS}$	$\mathbb{R}_+$	timer for simulating the fuel flow

6.5.2.2 Colour function:

Place	Colour type	Colour function
AC_FS_P1	FS	$dm_f = -f_f dt$ $dt_{timerFS} = -dt$

6.5.2.3 Initial marking:

Place	Initial colour
AC_FS_P1	no token

6.5.2.4 Guard transitions:

Transition	Firing condition	Firing function
AC_FS_G1: AC_FS_P1 ∧ [EN_P1 ∧ AC_CH_P1 ∧ AC_EV_P] → AC_FS_P1	$t_{timerFS} \le 0$	a token with colour FS: $t_{timerFS} = \Delta t_{timerFS}$ if $S_f = taxi$ $f_f = C_{tx}$ if $S_f = climb$ $f_f = T_{cl} \cdot \eta$ if $S_f = cruise$ $f_f = C_{fcr} \cdot T_{cr} \cdot \eta$ $T_{cr} = \begin{cases} D & \text{if} D < T_{cl} \cdot C_{Tcr} \\ T_{cl} \cdot C_{Tcr} & \text{if} D \ge T_{cl} \cdot C_{Tcr} \end{cases}$ $D = \frac{1}{2}\left(v^{TAS}\right)^2 S\rho \cdot \left[C_{D1} + C_{D2} \cdot \left(\frac{2g\left(m_0 + m_f\right)}{\left(v^{TAS}\right)^2 S\rho} \right)^2 \right]$

		if $S_f = descent$ if $h > h_{app}$ $f_f = C_{f3} \cdot \left(1 - \frac{h}{C_{f4}}\right)$ if $h \in \left(h_{ld}, h_{app}\right]$ $f_f = C_{Tapp} \cdot T_{cl} \cdot \eta$ if $h \le h_{ld}$ $f_f = C_{Tld} \cdot T_{cl} \cdot \eta$ where $\eta = C_{f1} \cdot \left(1 + \frac{v^{TAS}}{C_{f2}}\right)$ $T_{cl} = C_{T1} \cdot \left(1 - \frac{h}{C_{T2}} + C_{T3} h^2\right)$ $\rho = \rho(h)$
	Used variables from other LPNs: • EN: $g, \rho(h)$ • AC_CH: all the parameters related to fuel flow • AC_EV: S_f, v^{TAS}, h	

6.5.2.5 Parameters:

Parameters	Description	Value	Unit
$\Delta t_{timerFS}$	time step in modelling fuel consumption	30	s

6.5.2.6 Incoming arcs from different LPNs

There are incoming arcs from EN, AC_CH, AC_EV and CR_PL:

- Environment contains parameters and information about air density required to compute fuel flow.
- Parameters from characteristics are used to determine fuel flow.
- Information from evolution (mode, altitude, speed) determines fuel flow.
- Crew determines the initial fuel intake (based on plan from AOC).

6.5.2.7 Outgoing arcs to different LPNs

There are outgoing arcs to AC_EV and CR_SA:

- Aircraft continues in its trajectory only if there is enough fuel in tanks.
- Crew regularly checks the fuel level.

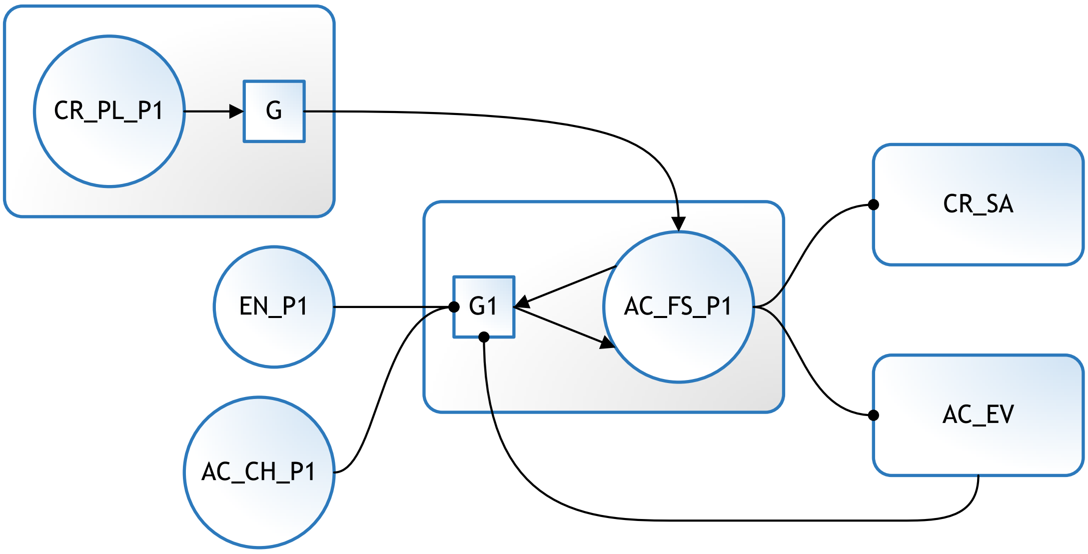


Figure 6.17: interactions between AC_FS and other LPNs

6.5.3 Fuel consumption model

In this section, we explain how the fuel flow f_f in AC_FS_G1 is determined. Four different modes of flying are distinguished when computing fuel flow: taxi, climb, cruise, descent. We used formulas and aircraft parameters from Base of Aircraft Data version 3.9 developed by Eurocontrol (Eurocontrol Experimental Centre, 2011) for climb, cruise and descent. To determine fuel flow during taxiing, we used actual flight data of a commercial airline.

First we will define the variables that are going to be used in the formulas.

Variable	Notation	Unit
fuel flow	f_f	$kg \cdot s^{-1}$
thrust	$T_{cl}, T_{cr}, T_{app}, T_{ld}$	N
thrust specific fuel consumption	η	$kg \cdot s^{-1} \cdot N^{-1}$
drag	D	N
weight of the aircraft	m	kg
air density	ρ	$kg \cdot m^{-3}$
true air speed	v	$m \cdot s^{-1}$
altitude	h	m

The parameters and constants used in the computation are listed as parameters of AC_CH (see 6.5.1.3) and EN (see 6.2.1.6). Aircraft-specific parameters are parameters for type Boeing 737-800. The atmosphere model considered in the computations is the International Standard Atmosphere (ISA) defined in chapter 3 of (Eurocontrol Experimental Centre, 2011). The air density is determined only by altitude and there is no wind considered. Since we always use true air speed (speed of the aircraft relative to the air around it) in computing fuel flow, we can consider wind model separately from ISA model. Then by adding wind speed to the true air speed, we will get ground speed of the aircraft. The ISA model is defined in the environment LPN in section 6.2.1.2.

In most cases, we compute fuel flow by determining the thrust and the thrust specific fuel consumption η which depends on the true air speed v.

$$\eta(v) = C_{f1} \cdot \left(1 + \frac{v}{C_{f2}}\right)$$

6.5.3.1 Taxi fuel flow

Considering the fact that taxiing takes only several minutes and the fuel flow is lowest during this mode, we simplify the model by assuming constant fuel flow during taxiing.

$$f_f^{taxi} = C_{tx}$$

The value of constant C_{tx} is the average fuel flow calculated using data from more than 8500 flights.

6.5.3.2 Climb fuel flow

We compute fuel flow during climb using thrust T_{cl} and thrust specific fuel consumption $\eta(v)$.

$$f_f^{cl}(h, v) = T_{cl} \cdot \eta(v)$$

$$T_{cl} = C_{T1} \cdot \left(1 - \frac{h}{C_{T2}} + C_{T3} h^2\right)$$

6.5.3.3 Cruise fuel flow

The thrust during cruise under nominal conditions is equal to drag. Thrust has an upper limit so the aircraft cannot achieve speed that would require thrust to be higher than this limit. Fuel flow is then determined using thrust specific fuel consumption $\eta(v)$ and cruise fuel flow correction factor C_{fcr}.

$$f_f^{cr}(h, v, m) = C_{fcr} \cdot T_{cr} \cdot \eta(v)$$

$$T_{cr} = \begin{cases} D & \text{if} \quad D < T_{cl} \cdot C_{Tcr} \\ T_{cl} \cdot C_{Tcr} & \text{if} \quad D \geq T_{cl} \cdot C_{Tcr} \end{cases}$$

$$D = \frac{1}{2}\left(v^{TAS}\right)^2 S\rho(h) \cdot \left[C_{D1} + C_{D2} \cdot \left(\frac{2g\left(m_0 + m_f\right)}{\left(v^{TAS}\right)^2 S\rho(h)}\right)^2\right]$$

6.5.3.4 Descent fuel flow

6.5.3.4.1 Idle thrust descent fuel flow

During idle thrust descent, the fuel flow is minimal and depends only on altitude. The aircraft is in idle thrust descent if the altitude satisfies $h > h_{app}$.

$$f_f^{idle}(h) = C_{f3} \cdot \left(1 - \frac{h}{C_{f4}}\right)$$

6.5.3.4.2 Approach fuel flow

For altitudes $h \in \left(h_{ld}, h_{app}\right]$, the aircraft is in approach phase. Thrust is determined from climb thrust by correction factor C_{Tapp}. Fuel flow is then determined using thrust specific fuel consumption $\eta(v)$.

$$f_f^{app}(h,v) = T_{app} \cdot \eta(v)$$
$$T_{app} = T_{cl} \cdot C_{Tapp}$$

6.5.3.4.3 Landing fuel flow

For altitudes $h < h_{ld}$, the aircraft is in landing phase. Thrust is determined from climb thrust by correction factor C_{Tapp}. Fuel flow is then determined using thrust specific fuel consumption $\eta(v)$.

$$f_f^{land}(h,v) = T_{ld} \cdot \eta(v)$$
$$T_{ld} = T_{cl} \cdot C_{Tld}$$

6.5.3.4.4 Descent fuel flow

For convenience, we define a function determining the fuel flow for the whole descent part of the flight.

$$f_f^{de}(h,v) = \begin{cases} f_f^{idle}(h) & \text{if} \quad h > h_{app} \\ f_f^{app}(h,v) & \text{if} \quad h \in \left(h_{ld}, h_{app}\right] \\ f_f^{land}(h,v) & \text{if} \quad h \in \left[0, h_{ld}\right] \end{cases}$$

6.5.4 Aircraft evolution LPN (AC_EV)

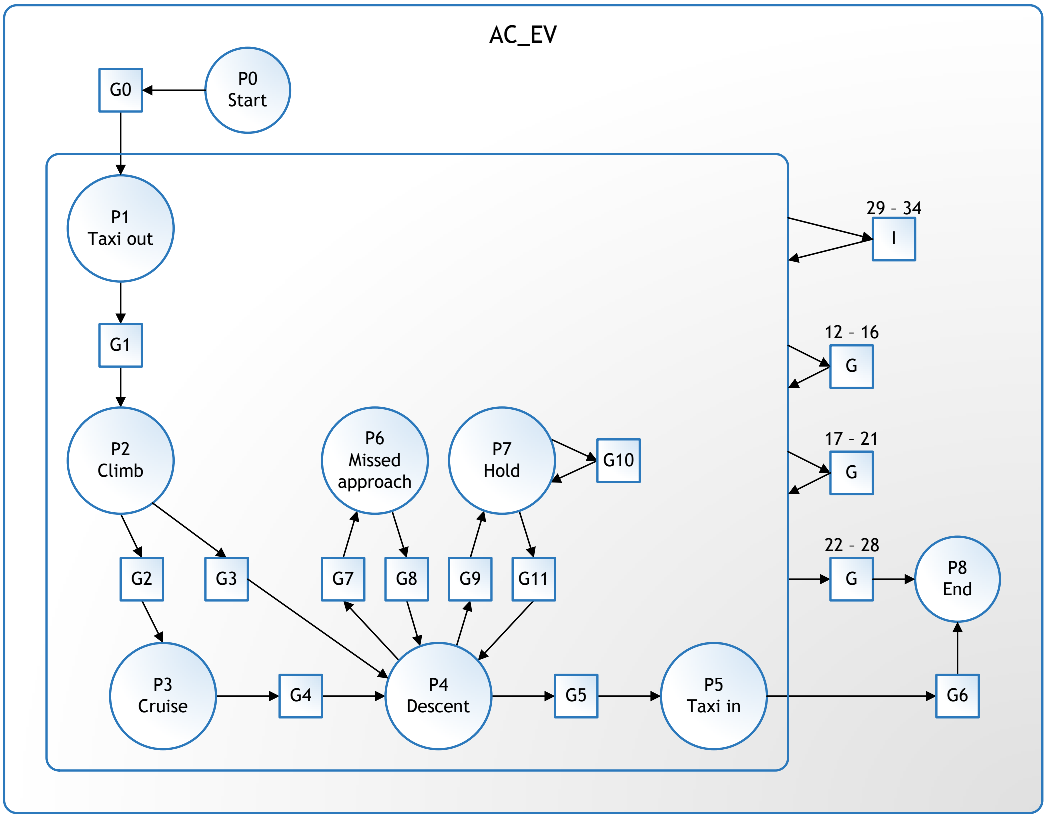


Figure 6.18: local Petri net of aircraft evolution

6.5.4.1 Colour type: EV

Notation	State space	Description
x, y	$\mathbb{R}^2$	current position
n, m	$\mathbb{Z}^2$	index of current sector
h	$\mathbb{R}_+$	current altitude
$s = \begin{pmatrix} s_x \\ s_y \end{pmatrix}$	$\mathbb{R}^2$	direction along x and y axes vector s is always a unit vector
v_h	$\mathbb{R}$	rate of climb/descent
v^{TAS}	$\mathbb{R}_+$	absolute true air speed of the aircraft
v^{GS}	$\mathbb{R}_+$	absolute ground speed of the aircraft
$w = \begin{pmatrix} w_x \\ w_y \end{pmatrix}$	$\mathbb{R}^2$	wind speed in current sector at current altitude
$W^j = \begin{pmatrix} W_x^j \\ W_y^j \end{pmatrix}$	$\mathbb{R}^2$	Waypoints that form the current route of the aircraft
N_W	$\mathbb{Z}$	number of waypoints in the current route
J_{next}	$\mathbb{Z}$	index of the next waypoint
x_{hold}, y_{hold}	$\mathbb{R}^2$	Point from which the aircraft is holding in current direction.
r_{de}	$\mathbb{R}_+$	descent correction ratio

S_f	$\{taxi, climb, cruise, descent\}$	mode of flight
$t_{timerTx}$	$\mathbb{R}_+$	timer for the taxiing
$t_{timerHold}$	$\mathbb{R}_+$	timer for the holding
Derived variables		
v_x^{TAS}, v_y^{TAS}	$\mathbb{R}_+, \mathbb{R}_+$	true air speed along x and y axes
v_x^{GS}, v_y^{GS}	$\mathbb{R}_+, \mathbb{R}_+$	ground speed along x and y axes
d_{left}	$\mathbb{R}_+$	length of the remaining route from current position to current destination
Index follows this set: $j = 0, \ldots, N_W - 1$		

6.5.4.2 Definition of derived variables from Colour type:

Notation	Definition
v_x^{TAS}, v_y^{TAS}	$v_x^{TAS} = s_x v^{TAS}$ $v_y^{TAS} = s_y v^{TAS}$
v_x^{GS}, v_y^{GS}	$v_x^{GS} = s_x v^{GS}$ $v_y^{GS} = s_y v^{GS}$
d_{left}	$d_{left} = \left\| W^{J_{next}} - \begin{pmatrix} x \\ y \end{pmatrix} \right\| + \sum_{j=J_{next}}^{N_W-2} \left\| W^{j+1} - W^{j} \right\|$

6.5.4.3 Colour function:

Place	Colour type	Colour function
AC_EV_P0 start	no colour	none
AC_EV_P1 taxi out	EV	$dt_{timerTx} = -dt$
AC_EV_P2 climb	EV	$v^{TAS} = v_{cl}^{TAS}(h)$ $v_h = v_{ROC}(h)$
AC_EV_P3 cruise	EV	$v^{TAS} = v_{cr}^{TAS}(h)$
AC_EV_P4 descent	EV	$v^{TAS} = v_{de}^{TAS}(h)$ $v_h = -r_{de} v_{ROD}(h)$
AC_EV_P5 taxi in	EV	$dt_{timerTx} = -dt$
AC_EV_P6 missed approach	EV	$v^{TAS} = v_{cl}^{TAS}(h)$ $v_h = v_{ROC}(h)$
AC_EV_P7 hold	EV	$dt_{timerHold} = -dt$ $v^{TAS} = v_{cr}^{TAS}(h)$
AC_EV_P8	no colour	$dt_{timerTx} = -dt$

AC_EV_P1 – AC_EV_P7	EV	$v^{GS} = v^{TAS} + \lvert ss^{T}w \rvert$ $dx = v_x^{GS}dt$ $dy = v_y^{GS}dt$ $dh = v_h dt$

6.5.4.4 Initial marking:

Place	Initial colour
AC_EV_P0	a token with no colour
AC_EV_P1 – AC_EV_P8	no token

6.5.4.5 Guard transitions:

Transition	Firing condition	Firing function
AC_EV_G0: start → taxi out AC_EV_P0 ∧ [AP_P1 ∧ AC_FS_P1 ∧ CR_SA_P2] → AC_EV_P1	$m_f > 0$	a token with colour EV: $x = x_A^{I_S}$ $y = y_A^{I_S}$ $n = n_A^{I_S}$ $m = m_A^{I_S}$ $h = 0$ $N_W = N_{WCR}$ $j = 0, \ldots, N_{WCR} - 1:$ $W^j = W_{CR}^j$ $J_{next} = 1$ $s = \left\lvert \begin{pmatrix} W_x^{J_{next}} - x \\ W_y^{J_{next}} - y \end{pmatrix} \right\rvert^{-1} \cdot \begin{pmatrix} W_x^{J_{next}} - x \\ W_y^{J_{next}} - y \end{pmatrix}$ $v_h = 0$ $v^{TAS} = 0$ $v^{GS} = 0$ $w = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ $x_{hold} = 0$ $y_{hold} = 0$ $r_{de} = 1$ $S_f = taxi$ $t_{timerHold} = 0$ $t_{timerTx} \sim N\left(t_{tx}^{I_S}, \sigma_{tx}^{I_S}, t_{txmin}^{I_S}, t_{txmax}^{I_S}\right)$
	Used variables from other LPNs: • AP: $x_A^{I_S}, y_A^{I_S}, n_A^{I_S}, m_A^{I_S}, t_{tx}^{I_S}, \sigma_{tx}^{I_S}, t_{txmin}^{I_S}, t_{txmax}^{I_S}$ • AC_FS: m_f • CR_SA: N_{WCR}, W_{CR}^j, I_S	

AC_EV_G1: taxi out → climb AC_EV_P1 ∧ [EN_P1] → AC_EV_P2	$t_{timerTx} \leq 0$	a token with colour EV: $w = w^{nm}(h)$ $v^{TAS} = v_{cl}^{TAS}(h)$ $v^{GS} = v^{TAS} + ss^{\mathsf{T}} w$ $S_f = climb$ other variables retain their values from AC_EV_P1
	Used variables from other LPNs: • EN: $w^{nm}(h)$	
AC_EV_G2: climb → cruise AC_EV_P2 ∧ [EN_P1 ∧ CR_SA_P2] → AC_EV_P3	$h \geq H_{cr}$	a token with colour EV: $w = w^{nm}(h)$ $v^{TAS} = v_{cr}^{TAS}(h)$ $v^{GS} = v^{TAS} + ss^{\mathsf{T}} w$ $v_h = 0$ $S_f = cruise$ other variables retain their values from AC_EV_P2
	Used variables from other LPNs: • EN: $w^{nm}(h)$ • CR_SA: H_{cr}	
AC_EV_G3: climb → descent AC_EV_P2 ∧ [EN_P1 ∧ CR_SA_P2] → AC_EV_P4 ∧ IPN_CR_SA ∧ IPN1_AP_I_D	$d_{left} \leq d_{de,h}$	place AC_EV_P4: a token with colour EV: $w = w^{nm}(h)$ $v^{TAS} = v_{de}^{TAS}(h)$ $v^{GS} = v^{TAS} + ss^{\mathsf{T}} w$ $v_h = 0$ $S_f = descent$ other variables retain their values from AC_EV_P2 place IPN_CR_SA: a token with no colour place IPN1_AP_I_D: a token with no colour
	Used variables from other LPNs: • EN: $w^{nm}(h)$ • CR_SA: I_D	
AC_EV_G4: cruise → descent AC_EV_P3 ∧ [EN_P1 ∧	$d_{left} \leq d_{de,h}$	place AC_EV_P4: a token with colour EV: $w = w^{nm}(h)$ $v^{TAS} = v_{de}^{TAS}(h)$ $v^{GS} = v^{TAS} + ss^{\mathsf{T}} w$

CR_SA_P2] → AC_EV_P4 ∧ IPN_CR_SA ∧ IPN1_AP_I_D		$v_h = 0$ $S_f = descent$ other variables retain their values from AC_EV_P3 place IPN_CR_SA: a token with no colour place IPN1_AP_I_D: a token with no colour
	Used variables from other LPNs: • EN: $w^{nm}(h)$ • CR_SA: I_D	
AC_EV_G5: descent → taxi in AC_EV_P4 ∧ AP_P1 → AC_EV_P5	$h \leq 0$	a token with colour EV: $v_h = 0$ $v^{TAS} = 0$ $v^{GS} = 0$ $w = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ $S_f = taxi$ $t_{timerTx} \sim N\left(t_{tx}^{I_D}, \sigma_{tx}^{I_D}, t_{txmin}^{I_D}, t_{txmax}^{I_D}\right)$ other variables retain their values from AC_EV_P4
	Used variables from other LPNs: • AP: $t_{tx}^{I_D}, \sigma_{tx}^{I_D}, t_{txmin}^{I_D}, t_{txmax}^{I_D}$ • CR_SA: I_D	
AC_EV_G6: taxi in → end AC_EV_P5 → AC_EV_P8	$t_{timerTx} \leq 0$	a token with no colour
AC_EV_G7: descent → missed approach AC_EV_P4 ∧ [EN_P1 ∧ AP_P1 ∧ CR_SA_P2] → AC_EV_P6	$L_{MA} = true$ and $h \leq h_{MA}^{I_D}$	a token with colour EV: $w = w^{nm}(h)$ $v^{TAS} = v_{cl}^{TAS}(h)$ $v^{GS} = v^{TAS} + ss^{\mathsf{T}} w$ $S_f = climb$ other variables retain their values from AC_EV_P4
	Used variables from other LPNs: • EN: $w^{nm}(h)$ • AP: $h_{MA}^{I_D}$ • CR_SA: L_{MA}, I_D	

<table>
<tr>
<td rowspan="2">AC_EV_G8:
missed approach
→ descent

AC_EV_P6 ∧
[EN_P1 ∧ AP_P1 ∧
CR_SA_P2] →
AC_EV_P4 ∧
IPN_CR_SA ∧
IPN1_AP_I$_D$</td>
<td>$h \geq h_{hold}^{I_D}$</td>
<td>place AC_EV_P4:
a token with colour EV:
$w = w^{nm}(h)$
$s = \left\| \begin{pmatrix} W_x^{J_{next}} - x \\ W_y^{J_{next}} - y \end{pmatrix} \right\|^{-1} \cdot \begin{pmatrix} W_x^{J_{next}} - x \\ W_y^{J_{next}} - y \end{pmatrix}$
$v^{TAS} = v_{de}^{TAS}(h)$
$v^{GS} = v^{TAS} + ss^{\mathsf{T}} w$
$r_{de} = \frac{d_{de,h}(h)}{d_{left}}$
$S_f = descent$

other variables retain their values from AC_EV_P6

place IPN_CR_SA:
a token with no colour

place IPN1_AP_I$_D$:
a token with no colour</td>
</tr>
<tr>
<td colspan="2">Used variables from other LPNs:
• EN: $w^{nm}(h)$
• AP: $h_{hold}^{I_D}$
• CR_SA: I_D</td>
</tr>
<tr>
<td rowspan="2">AC_EV_G9:
descent → hold

AC_EV_P4 ∧
[EN_P1 ∧ AP_P1 ∧
CR_SA_P2] →
AC_EV_P7 ∧
IPN2_AP_I$_D$</td>
<td>$L_{hold}^{I_D} = true$ and
$h \leq h_{hold}^{I_D}$</td>
<td>place AC_EV_P7:
a token with colour EV:
$s = \begin{pmatrix} -s_y \\ s_x \end{pmatrix}$
$x_{hold} = x$
$y_{hold} = y$
$w = w^{nm}(h)$
$v^{TAS} = v_{cr}^{TAS}(h)$
$v^{GS} = v^{TAS} + ss^{\mathsf{T}} w$
$v_h = 0$
$S_f = cruise$
$t_{timerHold} \sim N\left(t_{hold}^{I_D}, \sigma_{hold}^{I_D}, t_{holdmin}^{I_D}, t_{holdmax}^{I_D}\right)$

other variables retain their values from AC_EV_P4

place IPN2_AP_I$_D$:
a token with no colour</td>
</tr>
<tr>
<td colspan="2">Used variables from other LPNs:
• EN: $w^{nm}(h)$
• AP: $L_{hold}^{I_D}, h_{hold}^{I_D}, t_{hold}^{I_D}, \sigma_{hold}^{I_D}, t_{holdmin}^{I_D}, t_{holdmax}^{I_D}$
• CR_SA: I_D</td>
</tr>
</table>

AC_EV_G10: hold → hold AC_EV_P7 ∧ EN_P1 → AC_EV_P7	$\left\|\begin{pmatrix} x - x_{hold} \\ y - y_{hold} \end{pmatrix}\right\| \geq d_{hold}^{I_D}$	a token with colour EV: $s = \left\|\begin{pmatrix} x_{hold} - x \\ y_{hold} - y \end{pmatrix}\right\|^{-1} \cdot \begin{pmatrix} x_{hold} - x \\ y_{hold} - y \end{pmatrix}$ $x_{hold} = x$ $y_{hold} = y$ $w = w^{nm}(h)$ $v^{TAS} = v_{cr}^{TAS}(h)$ $v^{GS} = v^{TAS} + ss^{\mathsf{T}}w$ other variables retain their values from AC_EV_P7
	Used variables from other LPNs: • EN: $w^{nm}(h)$	
AC_EV_G11: hold → descent AC_EV_P7 ∧ EN_P1 → AC_EV_P4 ∧ IPN_CR_SA	$t_{timerHold} \leq 0$	place AC_EV_P4: a token with colour EV: $w = w^{nm}(h)$ $s = \left\|\begin{pmatrix} W_x^{J_{next}} - x \\ W_y^{J_{next}} - y \end{pmatrix}\right\|^{-1} \cdot \begin{pmatrix} W_x^{J_{next}} - x \\ W_y^{J_{next}} - y \end{pmatrix}$ $v^{TAS} = v_{de}^{TAS}(h)$ $v^{GS} = v^{TAS} + ss^{\mathsf{T}}w$ $r_{de} = \frac{d_{de,h}(h)}{d_{left}}$ $S_f = descent$ other variables retain their values from AC_EV_P7 place IPN_CR_SA: a token with no colour
	Used variables from other LPNs: • EN: $w^{nm}(h)$	
AC_EV_G12 – AC_EV_G16: new sector AC_EV_P ∧ [EN_P1 ∧ AP_P1] → AC_EV_P	$\|m \cdot d_{sec} - x\| \geq d_{sec}$ or $\|n \cdot d_{sec} - y\| \geq d_{sec}$	a token with colour EV: $n = \left\lfloor \frac{y}{d_{sec}} + \frac{1}{2} \right\rfloor$ $m = \left\lfloor \frac{x}{d_{sec}} + \frac{1}{2} \right\rfloor$ $w = w^{nm}(h)$ $v^{GS} = v^{TAS} + ss^{\mathsf{T}}w$ $r_{de} = \frac{d_{de,h}(h)}{d_{left}}$ other variables retain their values from AC_EV_P
	used variables from other LPNs: • EN: $w^{nm}(h)$	

	• AP: d_{sec} transition is defined for places: AC_EV_P ∊ {AC_EV_P2, AC_EV_P3, AC_EV_P4, AC_EV_P6, AC_EV_P7}	
AC_EV_G17 – AC_EV_G21: new waypoint AC_EV_P ∧ AP_P1 → AC_EV_P	$n=\left\lfloor \frac{W_y^{J_{next}}}{d_{sec}}+\frac{1}{2}\right\rfloor$ and $m=\left\lfloor \frac{W_x^{J_{next}}}{d_{sec}}+\frac{1}{2}\right\rfloor$	a token with colour EV: $J_{next} = J_{next} + 1$ $s = \left\lvert \begin{pmatrix} W_x^{J_{next}} - x \\ W_y^{J_{next}} - y \end{pmatrix} \right\rvert^{-1} \cdot \begin{pmatrix} W_x^{J_{next}} - x \\ W_y^{J_{next}} - y \end{pmatrix}$ $v^{GS} = v^{TAS} + ss^{\mathsf{T}}w$ other variables retain their values from AC_EV_P
	used variables from other LPNs: • AP: d_{sec} transition is defined for places: AC_EV_P ∊ {AC_EV_P2, AC_EV_P3, AC_EV_P4, AC_EV_P6, AC_EV_P7}	
AC_EV_G22 – AC_EV_G28: fuel starvation AC_EV_P ∧ AC_FS_P1 → AC_EV_P8	$m_f \le 0$	a token with no colour
	used variables from other LPNs: • AC_FS: m_f transition is defined for places: AC_EV_P ∊ { AC_EV_P1, AC_EV_P2, AC_EV_P3, AC_EV_P4, AC_EV_P5, AC_EV_P6, AC_EV_P7}	

6.5.4.6 Immediate transitions:

Place	Initial colour
AC_EV_I29: change route AC_EV_P1 ∧ IPN_AC_EV → AC_EV_P1	a token with colour EV: $N_W = N_{IPN}$ $j = 0, \ldots, N_{IPN} - 1:$ $W^j = W_{IPN}^j$ $J_{next} = 1$ $s = \left\lvert \begin{pmatrix} W_x^{J_{next}} - x \\ W_y^{J_{next}} - y \end{pmatrix} \right\rvert^{-1} \cdot \begin{pmatrix} W_x^{J_{next}} - x \\ W_y^{J_{next}} - y \end{pmatrix}$ other variables retain their values from AC_EV_P1
	used variables from other LPNs: • IPN_AC_EV: N_{IPN}, W_{IPN}^j
AC_EV_I30: change route AC_EV_P3 ∧ IPN_AC_EV → AC_EV_P3	a token with colour EV: $N_W = N_{IPN}$ $j = 0, \ldots, N_{IPN} - 1:$ $W^j = W_{IPN}^j$ $J_{next} = 1$ $s = \left\lvert \begin{pmatrix} W_x^{J_{next}} - x \\ W_y^{J_{next}} - y \end{pmatrix} \right\rvert^{-1} \cdot \begin{pmatrix} W_x^{J_{next}} - x \\ W_y^{J_{next}} - y \end{pmatrix}$ $v^{GS} = v^{TAS} + ss^{\mathsf{T}}w$

	other variables retain their values from AC_EV_P3
	used variables from other LPNs: IPN_AC_EV: N_{IPN}, W_{IPN}^{j}
AC_EV_I31 – AC_EV_I34: change route AC_EV_P ∧ IPN_AC_EV → AC_EV_P2	a token with colour EV: $N_W = N_{IPN}$ $j = 0, \ldots, N_{IPN} - 1:$ $W^j = W_{IPN}^{j}$ $J_{next} = 1$ $s = \left\| \begin{pmatrix} W_x^{J_{next}} - x \\ W_y^{J_{next}} - y \end{pmatrix} \right\|^{-1} \cdot \begin{pmatrix} W_x^{J_{next}} - x \\ W_y^{J_{next}} - y \end{pmatrix}$ $v^{TAS} = v_{cl}^{TAS}(h)$ $v^{GS} = v^{TAS} + ss^{\mathsf{T}} w$ other variables retain their values from AC_EV_P
	used variables from other LPNs: IPN_AC_EV: N_{IPN}, W_{IPN}^{j} transition is defined for places: AC_EV_P ∊ {AC_EV_P2, AC_EV_P4, AC_EV_P6, AC_EV_P7}

6.5.4.7 Parameters:

Parameters	Description	Value	Unit
v_{cl}^{TAS}	matrix of type $\mathbb{R}^{26\times2}$ representing true air speed during climb at specific altitude h		$m, m \cdot s^{-1}$
v_{cr}^{TAS}	matrix of type $\mathbb{R}^{21\times2}$ representing true air speed during cruise at specific altitude h		$m, m \cdot s^{-1}$
v_{de}^{TAS}	matrix of type $\mathbb{R}^{26\times2}$ representing true air speed during descent at specific altitude h		$m, m \cdot s^{-1}$
v_{ROC}	matrix of type $\mathbb{R}^{26\times2}$ representing rate of climb at specific altitude h		$m, m \cdot s^{-1}$
v_{ROD}	matrix of type $\mathbb{R}^{26\times2}$ representing rate of descent at specific altitude h		$m, m \cdot s^{-1}$
$d_{de,h}$	matrix of type $\mathbb{R}^{26\times2}$ representing distance the aircraft will travel before descending to ground from altitude h		$m, m \cdot s^{-1}$

6.5.4.8 Parameter specification:

In table below, we specify the matrices v_{cl}^{TAS}, v_{cr}^{TAS}, v_{de}^{TAS}, v_{ROC}, v_{ROD} and $d_{de,h}$. Values in the first column of the matrix are different altitudes (in kilometres). The values in second column of matrices v_{cl}^{TAS}, v_{cr}^{TAS}, v_{de}^{TAS}, v_{ROC} and v_{ROD} are the corresponding speeds (in meters per second). The values in second column of matrix $d_{de,h}$ are the corresponding distances (in kilometres).

h	$v_{cl}^{TAS}(h)$	$v_{cr}^{TAS}(h)$	$v_{de}^{TAS}(h)$	$v_{ROC}(h)$	$v_{ROD}(h)$	$d_{de,h}(h)$
0	80.76	-	74.07	12.11	3.83	0
152.4	81.28	-	74.59	12.03	3.91	2.928
304.8	81.8	-	77.67	11.95	4.13	5.868
457.21	85.4	-	83.85	12.34	3.97	8.792
609.61	85.91	-	100.31	12.26	4.79	11.83
914.41	97.74	118.31	118.31	13.76	5.09	18.57
1219.2	115.74	119.86	119.86	15.5	5.18	25.61
1828.8	139.92	139.92	139.92	16.43	6.74	38.84
2438.4	144.03	144.03	144.03	15.8	6.96	51.54
3048	177.47	148.66	171.81	15.12	9.56	63.16
3657.6	183.13	152.78	176.95	14.25	9.83	74.06
4267.3	188.27	175.92	182.1	13.35	10.09	85.06
4876.9	193.93	181.58	187.76	12.41	10.36	96.09
5486.5	199.59	186.73	193.41	11.44	10.62	107.2
6096.1	205.76	192.9	199.07	10.44	10.89	118.3
6705.7	211.93	198.56	205.25	9.41	11.14	129.5
7315.3	218.62	204.73	211.93	8.35	11.4	140.8
7924.9	225.31	211.42	218.62	7.25	11.64	152.2
8534.5	232.51	218.11	225.31	6.14	11.88	163.7
8839.3	236.11	221.71	228.91	5.57	11.99	169.2
9448.9	235.6	228.39	235.6	7.23	17.06	178.9
10059	233.54	233.54	233.54	6.22	16.09	187.4
10668	231.48	231.48	231.48	5.11	15.27	196.4
11278	229.94	229.94	229.94	3.59	13.48	206.3
11887	229.94	229.94	229.94	2.38	13.12	216.8
12497	229.94	229.94	229.94	1.04	12.91	227.5

6.5.4.9 Incoming arcs from different LPNs

There are incoming arcs from EN, AP, AO, AC_FS, IPN_AC_EV and CR_SA:

- Environment affects the ground speed and trajectory.
- Position of airports and their parameters are used in evolution.
- AOC prepares the flight plan used in evolution.
- Aircraft continues in its trajectory only if there is enough fuel in tanks.
- Interaction Petri net has the information about updated flight route.
- Crew makes decisions that affect the evolution.

6.5.4.10 Outgoing arcs to different LPNs

There are outgoing arcs to IPN1_AP, IPN2_AP, AC_FS, CR_SA and IPN_CR_SA:

- Airport interaction Petri net 1 sends the information whether the aircraft started to descent to airport which determines holding time.
- Airport interaction Petri net 2 sends the information whether the aircraft is already holding to airport.
- Information from evolution is used to determine fuel consumption.

- Crew regularly checks the state of the aircraft and uses the information to make decisions.
- Crew interaction Petri net sends the information whether the aircraft started to descent to crew which decides whether they will perform missed approach.

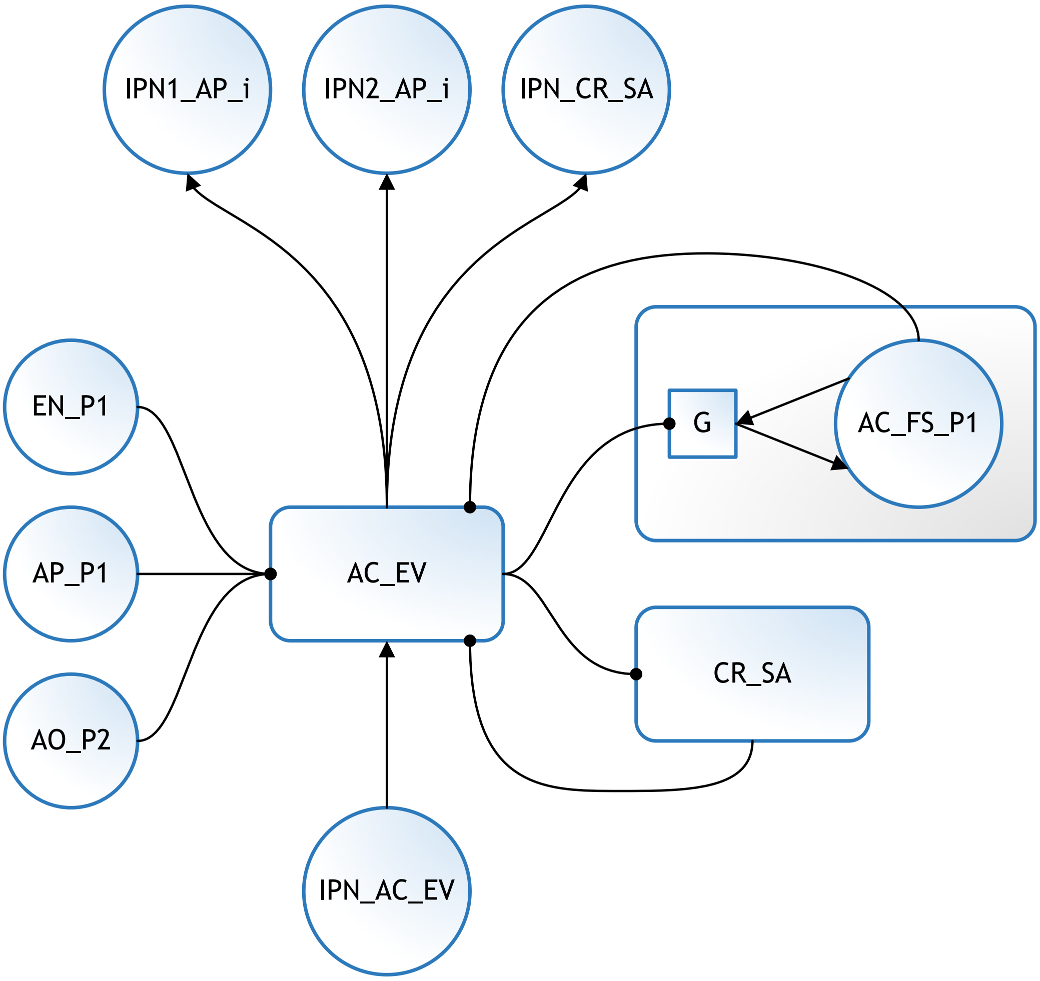


Figure 6.19: interactions between AC_EV and other LPNs

6.5.5 Interaction Petri net (IPN_AC_EV)

Figure 6.20: aircraft evolution interaction Petri net

6.5.5.1 Colour type: IPN_EC_EV

Place	State space	Description
$W_{IPN}^{j} = \begin{pmatrix} W_{IPNx}^{j} \\ W_{IPNy}^{j} \end{pmatrix}$	$\mathbb{R}^2$	Waypoints that form the route updated by the crew
N_{IPN}	$\mathbb{Z}$	number of waypoints of the route
Index follows this set: $j = 0, \ldots, N_{IPN} - 1$		

6.5.5.2 Colour function:

Place	Colour type	Colour function
IPN_AC_EV	IPN_AC_EV	constant

6.5.5.3 Initial marking:

Place	Initial colour
IPN_AC_EV	no token

6.5.5.4 Incoming arcs

There are incoming arcs from CR_SA_I1 and CR_SA_G5:

- The IPN gets the information about new route.

6.5.5.5 Outgoing arcs

There are outgoing arcs to AC_EV_I29 – AC_EV_I34:

- Aircraft takes the information about updated flight route.

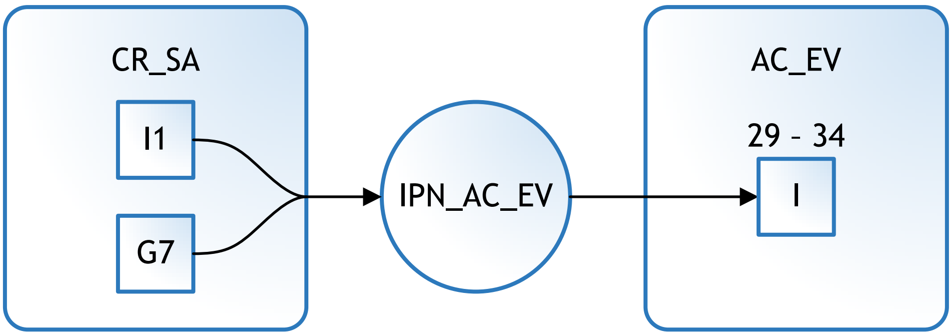


Figure 6.21: interactions between IPN_AC_EV and other LPNs

6.6 Crew agent

Crew agent consists of two local Petri nets:

- **Planning CR_PL:** Planning refers to decisions made by crew before the actual flight like asking for extra fuel after receiving the flight plan from AOC.
- **Situation awareness CR_SA:** This LPN contains the Situation Awareness (SA) and the intensions of the crew. Calculations performed in this LPN are very similar to those made in AO (pre-flight planning). SA includes the information about current position and fuel level of the aircraft. In-flight fuel management is also implemented in this LPN. The crew regularly updates the predictions made before and checks whether they have enough fuel. Replanning of the route occurs when an airspace sector along the route is closed or when, according to predictions, the destination airport cannot be reached with at least FRF left.

6.6.1 Crew planning LPN (CR_PL)

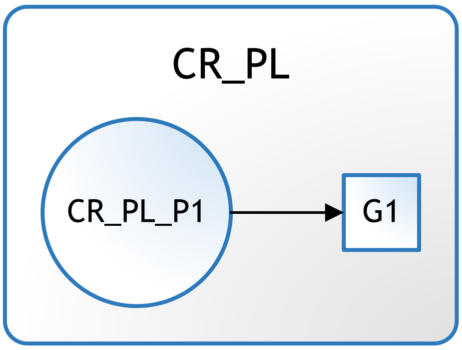


Figure 6.22: local Petri net of crew planning

6.6.1.1 Colour type: PL

Notation	State space	Description
L_{fEX}	$\{true,\ false\}$	determines whether the crew will ask for extra (discretionary) fuel
m_{fEX}	$\mathbb{R}_+$	amount of extra fuel asked

6.6.1.2 Colour function:

Place	Colour type	Colour function
CR_PL_P1	PL	constant

6.6.1.3 Initial marking:

Place	Initial colour
CR_PL_P1	a token with colour PL: $L_{fEX} = \begin{cases} true & \text{with probability } p_{fEX} \\ false & \text{with probability } 1-p_{fEX} \end{cases}$ $m_{fEX} = 0$

6.6.1.4 Guard transitions:

Transition	Firing condition	Firing function
CR_PL_G1: CR_PL_P1 ∧ [AO_P2 ∧ AC_CH_P1] → AC_FS_P1	$m_{fSUM} > 0$	a token with colour FS: $m_{fEX} = \begin{cases} r_{fEX} \cdot m_{fTR} & \text{if } L_{fEX} = true \\ 0 & \text{if } L_{fEX} = false \end{cases}$ $m_f = m_{fSUM} + m_{fEX}$ $f_f = C_{tx}$
	Used variables from other LPNs: • AO: m_{fTR}, m_{fSUM} • AC_CH: C_{tx}	

6.6.1.5 Parameters:

Parameters	Description	Value	Unit
p_{fEX}	probability of asking for extra fuel by crew	0.1	dimensionless
r_{fEX}	percentage or trip fuel asked as extra fuel	0.01	dimensionless

6.6.1.6 Incoming arcs from different LPNs

There are incoming arcs from AO, AC_CH:

- AOC sends the flight plan to the flight crew before the flight.
- Information from AC_CH is needed to determine initial fuel flow.

6.6.1.7 Outgoing arcs to different LPNs

There are outgoing arcs to AC_FS:

- Based on plan from AOC, crew determines the initial fuel intake.

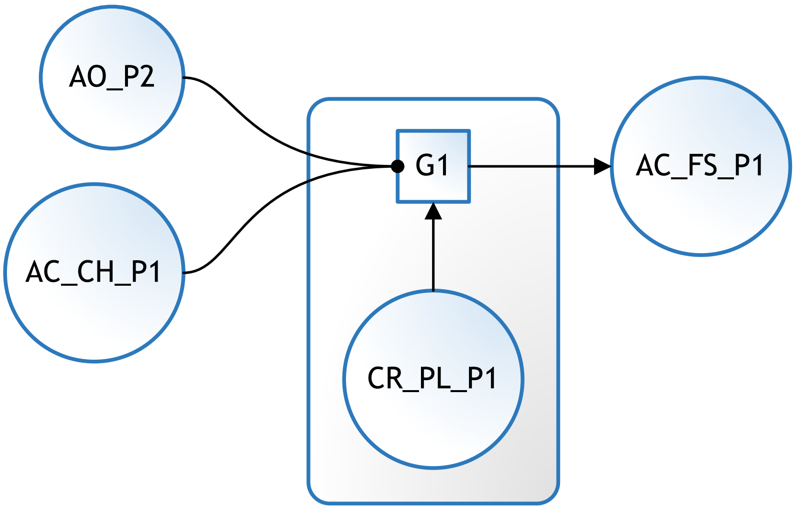


Figure 6.23: interactions between CR_PL and other LPNs

6.6.2 Crew situation awareness LPN (CR_SA)

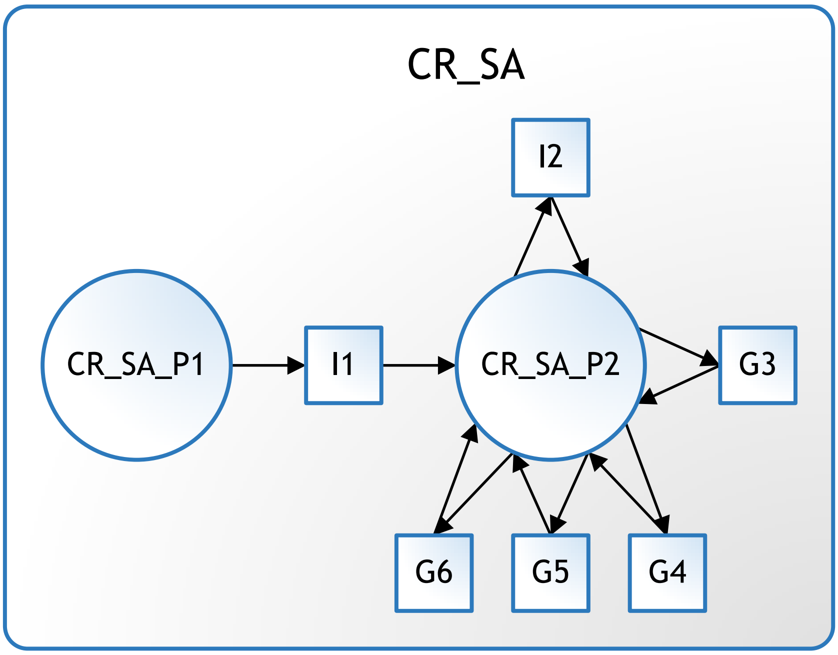


Figure 6.24: local Petri net of crew situation awareness

6.6.2.1 Colour type: SA

Notation	State space	Description
x_{CR}, y_{CR}, h_{CR}	$\mathbb{R}, \mathbb{R}, \mathbb{R}_+$	coordinates that are identified as the position and the altitude of the aircraft by the crew
$s_{CR} = \begin{pmatrix} s_{CRx} \\ s_{CRy} \end{pmatrix}$	$\mathbb{R}^2$	coordinates that are identified as the direction of the aircraft by the crew
S_{fCR}	$\{taxi, climb, cruise, descent\}$	mode of the flight as identified by the crew
$w_{CR}^{nm}(h) = \begin{pmatrix} w_{CRx}^{nm}(h) \\ w_{CRy}^{nm}(h) \end{pmatrix}$	$\mathbb{R}^2$	prediction of the speed of the wind in sector n, m at altitude h made by crew
A_{CR}^{nm}	$\{true, false\}$	prediction of the availability of the sector n, m made by crew
$W_{CR}^{j} = \begin{pmatrix} W_{CRx}^{j} \\ W_{CRy}^{j} \end{pmatrix}$	$\mathbb{R}^2$	waypoints of the current flight route
N_{WCR}	$\mathbb{Z}$	number of waypoints
H_{cr}	$\mathbb{R}_+$	planned cruising altitude
I_S, I_D, I_A	$\mathbb{Z}, \mathbb{Z}, \mathbb{Z}$	indices of the airports from which the crew departed, at which it is intending to land and which is planned as the alternate airport
L_{MA}	$\{true, false\}$	determines whether the crew is going to perform a missed approach or not
L_{route}	$\{true, false\}$	determines whether the crew is going to recalculate the flight route or not

m_{fCR}	$\mathbb{R}_+$	value that is identified as the amount of fuel in tanks
m_{fCR}^D	$\mathbb{R}_+$	amount of fuel that crew computed as the amount needed to get from current position to destination with at least FRF left
m_{fCR}^A	$\mathbb{R}_+$	amount of fuel that crew computed as the amount needed to get from current position to alternate with at least FRF left
$t_{timerInfo}$	$\mathbb{R}_+$	timer for updating the information of the crew
Indices follow these sets: $j = 0, \ldots, N_{WCR} - 1$ $n = 0, \ldots, N - 1$ $m = 0, \ldots, M - 1$		

6.6.2.2 Colour function:

Place	Colour type	Colour function
CR_SA_P1	no colour	none
CR_SA_P2	SA	$dt_{timerInfo} = -dt$

6.6.2.3 Initial marking:

Place	Initial colour
CR_SA_P1	a token with no colour
CR_SA_P2	no token

6.6.2.4 Immediate transitions:

Place	Initial colour
CR_SA_I1: CR_SA_P1 ∧ [EN_P1 ∧ AO_P2] → CR_SA_P2	a token with colour SA: $N_{WCR} = N_{WAO}$ $j = 0, \ldots, N_{WCR} - 1:$ $W_{CR}^j = W_{AO}^j$ $x_{CR} = W_{CRx}^0$ $y_{CR} = W_{CRy}^0$ $h_{CR} = 0$ $s_{CR} = \left\| \begin{pmatrix} W_{CRx}^1 - x \\ W_{CRy}^1 - y \end{pmatrix} \right\|^{-1} \cdot \begin{pmatrix} W_{CRx}^1 - x \\ W_{CRy}^1 - y \end{pmatrix}$ $n = 0, \ldots, N - 1$ $m = 0, \ldots, M - 1:$ $w_{CR}^{nm} = w^{nm}(h_{CR})$ $A_{CR}^{nm} = A^{nm}$ $H_{cr} = H_{crAO}$ $I_S = I_{SAO}$ $I_D = I_{EAO}$ $I_A = I_{AAO}$ $L_{MA} = false$ $L_{route} = false$ $m_{fCR} = 0$ $m_{fCR}^D = 0$ $m_{fCR}^A = 0$ $t_{timerInfo} = \mathrm{Exp}(\Delta t_{info})$
	used variables from other LPNs: • EN: $w^{nm}(h)$, A^{nm}

	• AO: $N_{WAO}, W^{j}_{AO}, H_{crAO}, I_{SAO}, I_{EAO}, I_{AAO}$
CR_SA_I2: CR_SA_P2 ∧ IPN_CR_SA → CR_SA_P2	a token with colour SA: $L_{MA} = \begin{cases} true & \text{with probability } p_{MA} \\ false & \text{with probability } 1-p_{MA} \end{cases}$ other variables retain their values from CR_SA_P2

6.6.2.5 Guard transitions:

Transition	Firing condition	Firing function
CR_SA_G3: CR_SA_P2 ∧ AP_P1 → CR_SA_P2	$\left\lfloor \frac{x_{CR}}{d_{sec}} + \frac{1}{2} \right\rfloor = \left\lfloor \frac{W^{1}_{CRx}}{d_{sec}} + \frac{1}{2} \right\rfloor$ and $\left\lfloor \frac{y_{CR}}{d_{sec}} + \frac{1}{2} \right\rfloor = \left\lfloor \frac{W^{1}_{CRy}}{d_{sec}} + \frac{1}{2} \right\rfloor$	a token with colour SA: $j = 0, \ldots, N_{WCR} - 2$ $W^{j}_{CR} = W^{j+1}_{CR}$ $N_{WCR} = N_{WCR} - 1$ other variables retain their values from CR_SA_P2
	Used variables from other LPNs: • AP: d_{sec} • AC_EV: x, y	
CR_SA_G4: CR_SA_P2 ∧ [AP_P1∧ AO_P2] → CR_SA_P2	$A^{n^{I_D}_{A} m^{I_D}_{A}}_{CR} = false$	a token with colour SA: if $A^{n^{I_A}_{A} m^{I_A}_{A}}_{CR} = true$ I_D is set to value of I_A and vice versa $L_{route} = false$ if $A^{n^{I_A}_{A} m^{I_A}_{A}}_{CR} = false$ $I_D = \arg\min_{i \in O_{AP}} \left\| \begin{pmatrix} x_{CR} - x^{i}_{A} \\ y_{CR} - y^{i}_{A} \end{pmatrix} \right\|$ $I_A = I_{AAO}$ where O_{AP} is set of all available airports $L_{route} = false$ other variables retain their values from CR_SA_P2
	Used variables from other LPNs: • AP: $n^{I_D}_{A}, m^{I_D}_{A}, x^{i}_{A}, y^{i}_{A}$ for $\forall i$ • AO: I_{EAO}	
CR_SA_G5: CR_SA_P2 ∧ AP_P1 → CR_SA_P2	$L_{route} = true$	a token with colour SA: Variables $W^{j}_{CR}, N_{WCR}, m^{D}_{fCR}, m^{A}_{fCR}$ are defined as described in section 6.6.3. $s_{CR} = \left\| \begin{pmatrix} W^{1}_{CRx} - x_{CR} \\ W^{1}_{CRy} - y_{CR} \end{pmatrix} \right\|^{-1} \cdot \begin{pmatrix} W^{1}_{CRx} - x_{CR} \\ W^{1}_{CRy} - y_{CR} \end{pmatrix}$ if $m_{fCR} < m^{D}_{fCR}$ and $m_{fCR} \geq m^{A}_{fCR}$ I_D is set to value of I_A and vice versa

		$L_{route} = true$ if $m_{fCR} < m_{fCR}^D$ and $m_{fCR} < m_{fCR}^A$ $I_D = \arg\min_{i \in O_{AP}} \left\| \begin{pmatrix} x_{CR} - x_A^i \\ y_{CR} - y_A^i \end{pmatrix} \right\|$ $I_A = I_{AAO}$ $L_{route} = true$ where O_{AP} is set of all available airports other variables retain their values from CR_SA_P2
	used variables from other LPNs: • EN: $\rho(h)$ • AP: $x_A^i, y_A^i, n_A^i, m_A^i, d_{sec}$ for $\forall i$ • AO: I_{AAO} • AC_EV: $v_{cl}^{TAS}, v_{cr}^{TAS}, v_{de}^{TAS}, v_{ROC}, v_{ROD}$ • AC_CH: all parameters regarding fuel flow	
CR_SA_G6: CR_SA_P2 ∧ AP_P1 → CR_SA_P2	$t_{timerInfo} \leq 0$	a token with colour SA: $x_{CR} = x$ $y_{CR} = y$ $h_{CR} = h$ $s_{CR} = \left\| \begin{pmatrix} W_{CRx}^1 - x \\ W_{CRy}^1 - y \end{pmatrix} \right\|^{-1} \cdot \begin{pmatrix} W_{CRx}^1 - x \\ W_{CRy}^1 - y \end{pmatrix}$ $n = 0, \ldots, N-1$ $m = 0, \ldots, M-1$: $w_{CR}^{nm} = w^{nm}(h_{CR}) + \varepsilon_{CR}^w$ $A_{CR}^{nm} = A^{nm}$ $m_{fCR} = m_f$ $t_{timerInfo} = \mathrm{Exp}(\Delta t_{info})$ $L_{route} = true$ where $\varepsilon_{CR}^w \sim N(0, \sigma_{wCR}^2)$
	used variables from other LPNs: • EN: $w^{nm}(h), A^{nm}$ • AC_FS: m_f • AC_EV: x, y, h	

6.6.2.6 Parameters:

Parameters	Description	Value	Unit
p_{MA}	probability of performing missed approach	0.00176	dimensionless

σ_{wCR}	standard deviation of error in wind speed prediction	1	$m \cdot s^{-1}$
Δt_{info}	mean of exponentially distributed time interval between two information updates	300	s

6.6.2.7 Incoming arcs from different LPNs

There are incoming arcs from EN, AP, AO, AC_CH, AC_FS, AC_EV and IPN_CR_SA:

- The current state of the airspace is part of crew`s Situation Awareness and they use the information to plan the rest of the flight and estimate the amount of fuel needed for it.
- CR_SA knows the position of the airports to make decisions during the flight.
- Information from the flight plan determines the intensions of the crew during the flight.
- Crew uses the information about aircraft to make decisions during flight.
- Crew regularly checks the fuel level.
- Crew regularly checks the state of the flight and uses the information to make decisions during flight.
- Interaction Petri net has the information that the aircraft started to descent. This triggers the decision about missed approach.

6.6.2.8 Outgoing arcs to different LPNs

There are outgoing arcs to AC_EV and IPN_AC_EV:

- Crew makes the decisions that affect the aircraft evolution.
- Crew sends the information about updated flight route to the interaction Petri net that forwards them to the aircraft evolution.

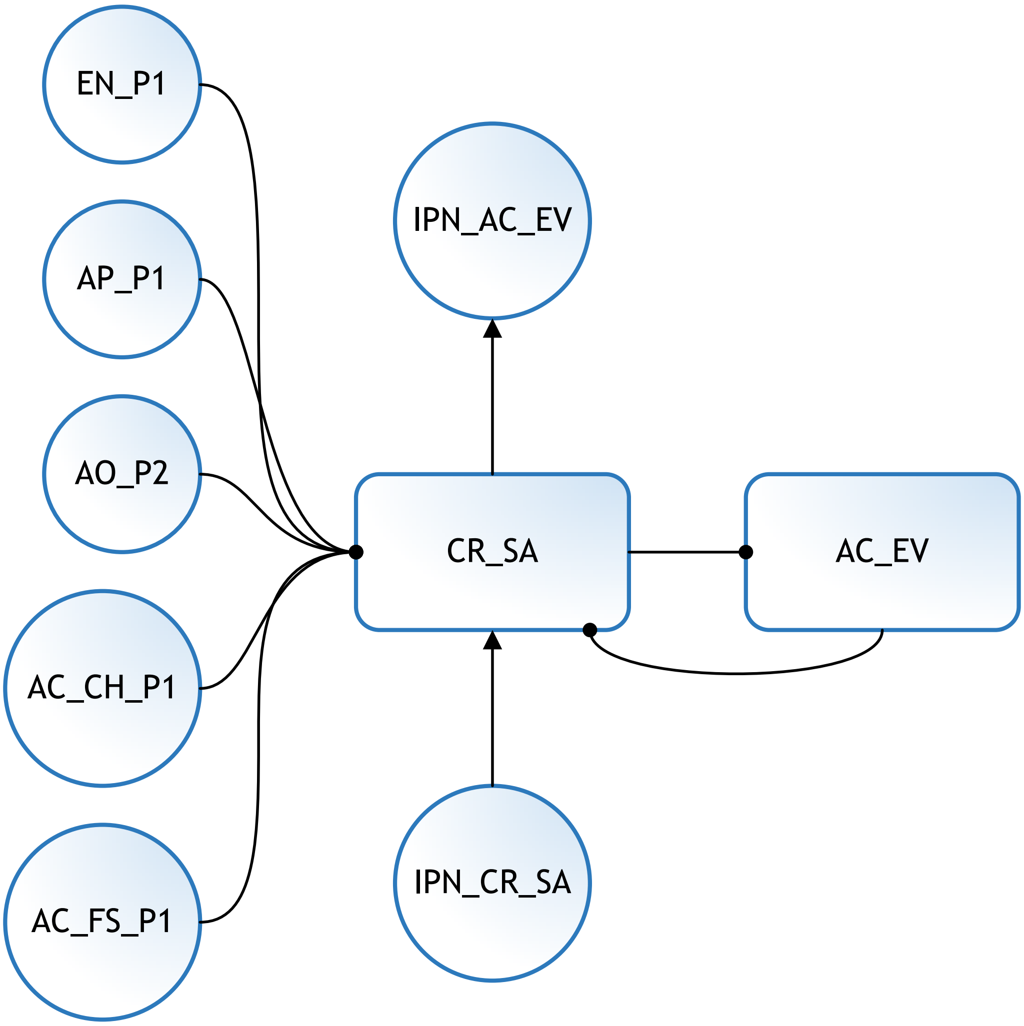


Figure 6.25: interactions between CR_SA and other LPNs

6.6.3 Recalculation of the route and fuel check by crew

In this section, we explain how the variables W_{CR}^{j}, N_{WCR}, m_{fCR}^{D} and m_{fCR}^{A} are computed in transition CR_SA_G5 (section 6.6.2.5). We will use following variables:

- $A_{CR}^{nm}, w_{CR}^{nm}(h), x_{CR}, y_{CR}, h_{CR}, S_{fCR}, H_{cr}, I_D, I_A$ where $n = 0, \ldots, N-1$ and $m = 0, \ldots, M-1$ from CR_SA
- $\rho(h)$ from EN
- n_A^i, m_A^i, d_{sec} where $i \in \{I_D, I_A\}$ from AP
- $v_{cl}^{TAS}, v_{cr}^{TAS}, v_{de}^{TAS}, v_{ROC}, v_{ROD}$ from AC_EV
- all parameters and formulas regarding fuel flow from AC_CH and AC_FS

Most of the computations done in transition CR_SA_G5 are the same as in transition AO_I1 in AOC agent. So we are not going to describe all the calculations in detail here, but rather point out the differences.

Variables W_{CR}^{j} and N_{WCR} are determined using the algorithm described in section 6.4.2.1. The difference is that the starting point is the current position of the aircraft, not the departure airport. Using these variables, we calculate the amount of fuel m_{fCR}^{D} needed to get to the destination with at least FRF remaining and the amount of fuel m_{fCR}^{A} needed to get to the alternate airport with at least FRF remaining.

First, we focus on m_{fCR}^{D}. Now following the steps described in section 6.4.2.2, we get to function $m_{f0,t}(t)$. In this situation, the function determines the amount of fuel needed for the remainder of flight at specific time t since the current time. Then we have

$$m_{fCR}^{D} = m_{f0,t}(0) + m_{fFR}$$

Following the same calculations for the alternate airport, we get to m_{fCR}^{A}.

There are few other differences in these calculations compared to the ones in section 6.4.2.2. One is that in this situation we do not have the same wind in each sector. It would be very complicated to include all the different values of wind speed and wind direction for every sector along the route. Also it is not realistic to assume that the crew of the aircraft has perfect information about the wind at every point of the route. So at this point we simplify it by taking the average of wind speed and direction at current position and on location of the destination.

We also have to adjust the computation based on the current flight mode and altitude. For example if the aircraft is cruising at the optimal altitude, we do not include the climbing part in the calculations. Or if the aircraft is climbing, we have to account for the current altitude h_{CR} that was already climbed.

6.6.4 Interaction Petri net (IPN_CR_SA)

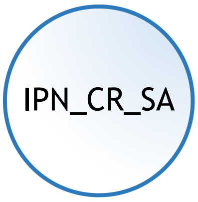


Figure 6.26: crew situation awareness interaction Petri net

6.6.4.1 Colour function:

Place	Colour type	Colour function
IPN_CR_SA	no colour	none

6.6.4.2 Initial marking:

Place	Initial colour
IPN_CR_SA	no token

6.6.4.3 Incoming arcs

There are incoming arcs from AC_EV_G3, AC_EV_G4, AC_EV_G8 and AC_EV_G11:

- The IPN gets the information that the aircraft started to descent.

6.6.4.4 Outgoing arcs

There are outgoing arcs to CR_SA_I2:

- The CR_SA gets the information that the aircraft started to descent.

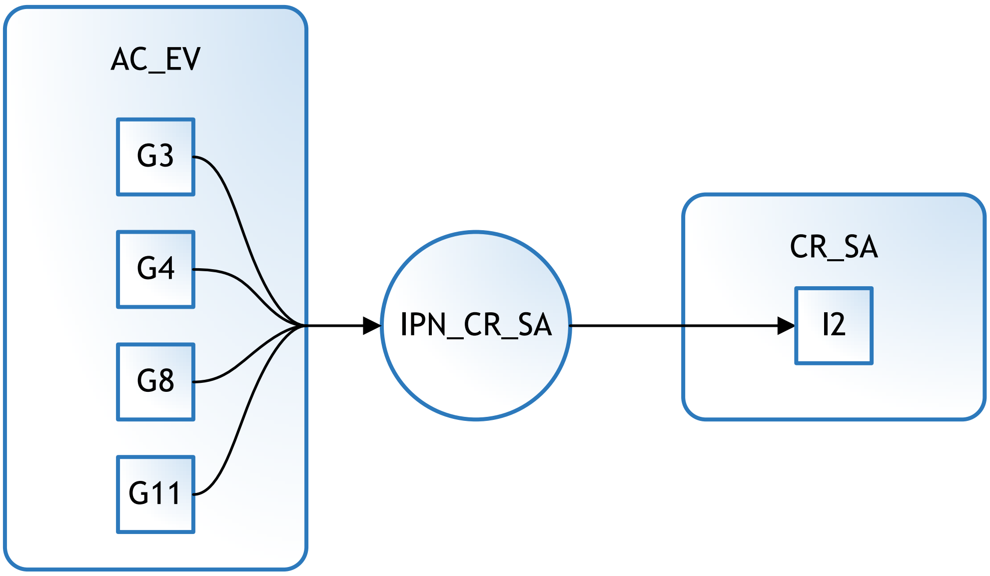


Figure 6.27: interactions between IPN_CR_SA and other LPNs

7 Model implementation

After designing the model, we implemented it using Java programming language. Since Petri nets have general structure very similar to object-oriented programs, the implementation of the model was quite intuitive maintaining the structure of the model explained in previous chapter.
The instructions on how to use the program are written in file *readme.docx* included together with the program.

7.1 Structure of the Java program

Using the terminology from Java, we implemented each local Petri net as a separate class. Then the whole model is made of 12 child classes extending one parent class (`LocalPetriNet`):

- `EN`
- `AP`
- `IPN_AP`
- `IPN_AP2`
- `AO`
- `AC_CH`
- `AC_FS`
- `AC_EV`
- `IPN_AC_EV`
- `CR_PL`
- `CR_SA`
- `IPN_CR_SA`

Within the class, all the parameters and colour variables are private variables of the class. Derived variables are public methods. Initial markings, Colour functions and transitions are implemented as public void methods. Each transition has a firing condition implemented as a public boolean method.
Objects of these 12 classes are then used all together to run one simulation which is equal to one flight. This is done by an object of class `Simulation`.
After each simulation, the results are written into special object of class `Result`. This object contains information like planned fuel consumption, planned trip time and distance, actual fuel consumption, actual trip time and distance, number of performed missed approaches, total holding time during the flight and important parameters used during this specific simulation.
Finally we use two more classes to calculate the aimed results:

- `MonteCarloRegular` – Using this class we perform the regular Monte Carlo simulation. Upon finishing, this program creates two output text files of comma-separated values format.

- `MonteCarloSplitting` – Using this class we perform the Splitting method simulation. Upon finishing, this program creates five output text files of comma-separated values format.

We further analyse these files using mathematical software MATLAB.

7.2 Output of the model

During the implementation of the model, we used one more class for testing the program. The class is called `TestFlight` and its function is to print values of various variables to the console along the flight so we could observe whether program is working properly. Here is one example of such output:

Time [min]	x [km]	y [km]	h [km]	S_f -	m_f [kg]	f_f [kg/s]
0	0	0	0.00	Start	0	0.000
5	100	200	0.00	Taxi out	9775	0.199
10	106	200	0.88	Climb	9611	1.918
15	153	197	5.22	Climb	9067	1.649
20	214	194	8.02	Climb	8631	1.313
25	282	191	9.95	Climb	8276	1.068
30	348	188	11.39	Climb	7994	0.860
35	415	184	12.15	Climb	7755	0.756
40	481	181	12.49	Cruise	7543	0.504
45	544	170	12.49	Cruise	7392	0.503
50	610	167	12.49	Cruise	7242	0.503
55	677	164	12.49	Cruise	7091	0.502
60	743	162	12.49	Cruise	6940	0.501
65	807	162	12.49	Cruise	6790	0.500
70	874	158	12.49	Cruise	6640	0.499
75	940	155	12.49	Cruise	6491	0.499
80	1006	151	12.49	Cruise	6341	0.498
85	1073	148	12.49	Cruise	6192	0.497
90	1138	145	12.49	Cruise	6043	0.496
95	1206	141	12.49	Cruise	5895	0.495
100	1272	138	12.49	Cruise	5746	0.495
105	1339	134	12.49	Cruise	5598	0.494
110	1406	131	12.49	Cruise	5450	0.493
115	1471	127	12.49	Cruise	5302	0.492
120	1538	124	12.49	Cruise	5154	0.491
125	1604	120	12.49	Cruise	5007	0.491
130	1670	117	12.49	Cruise	4860	0.490
135	1737	114	12.49	Cruise	4713	0.489
140	1804	110	10.65	Descent	4613	0.106
145	1869	107	6.74	Descent	4573	0.153
150	1923	104	3.66	Descent	4521	0.190
155	1966	102	1.47	Descent	4459	0.217
160	1981	104	1.00	Hold	4356	0.425
165	1981	102	1.00	Hold	4229	0.424
170	1982	102	1.00	Hold	4102	0.424
175	1994	100	0.22	Descent	3992	0.373
180	2000	100	0.00	Taxi in	3897	0.199

In this specific flight, we printed current values of seven variables every 5 minutes from the beginning of the flight until the end. The variables are:

- `Time` – time in minutes since the start of the flight
- `x` – current position of the aircraft on the x-axis in kilometres
- `y` – current position of the aircraft on the y-axis in kilometres
- `h` – current altitude of the aircraft in kilometres
- `S_f` – current flight mode
- `m_f` – amount of usable fuel left in tanks in kilograms
- `f_f` – current fuel flow in kilograms per second

7.3 Plots illustrating the flight

We used the `TestFlight` class to produce several plots to illustrate how the model works. First plot (Figure 7.1) shows the altitude of the aircraft at a specific time for three different flights. The flight labelled by black line is a nominal flight i.e. nothing unexpected happened during the flight. Green line displays a flight with holding. As we see on the plot, the aircraft was sent to holding at altitude 1 km above the airport before landing. During the third flight, labelled by blue line, the pilot decided to perform a missed approach.

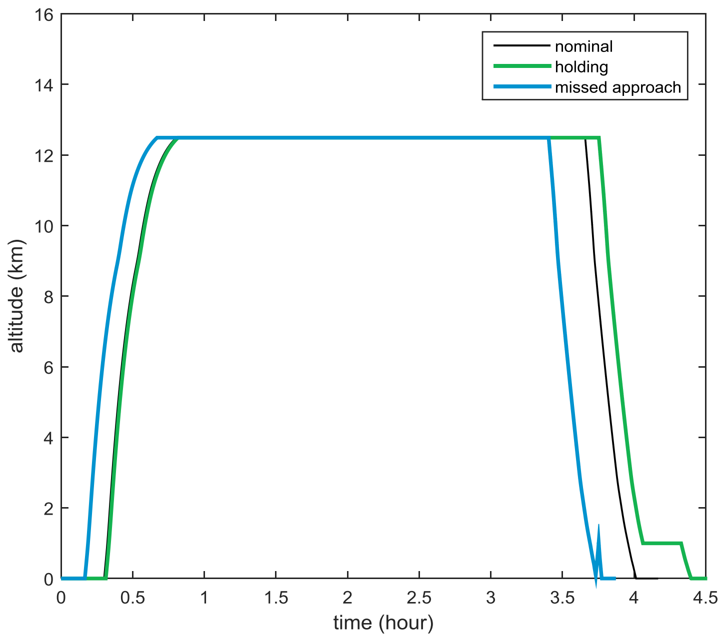


Figure 7.1: Altitude of the aircraft at a specific time for three different flights.

On the second plot (Figure 7.2), we have the amount of fuel left in tanks throughout the same three flights. Here, we see how the fuel flow changes along the flight. During taxiing, the fuel flow is very low and constant. Fuel flow is highest during the climb, especially during the start of climb. This is because the rate of climb is highest at lower altitudes and gradually decreases when aircraft reaches higher levels of the atmosphere where the drag is lower. Fuel flow during cruise is almost constant. It slightly decreases as the weight of the aircraft decreases due to burnt fuel, but that effect is too small to be visible on the figure. The descent of the flight is divided into three parts. The first is the idle thrust part, where the fuel flow is lowest because the engines are running with minimal thrust. During the approach and landing phases, the fuel flow is higher.

We can also notice that each flight starts with a different fuel intake. This reflects the fact that each flight starts with different airspace conditions. Different airspace sectors are closed and the wind speed and direction is also different every time. All these information are considered by airline operations control agent during the planning. If all flights would be planned perfectly and there would be nothing unexpected happening, then all flights would end up with approximately the same fuel level. But that is not the case in our model. We see that even though flight with holding, labelled by green line, starts with the most fuel, it ends with the least, followed by the flight with missed approach and nominal flight. This implies that holding burns the most unexpected fuel.

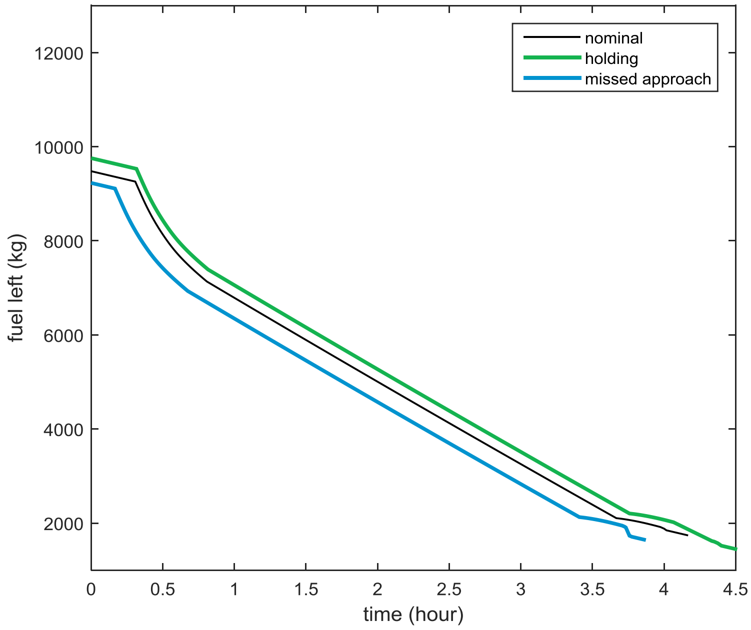


Figure 7.2: Amount of fuel left at a specific time for three different flights.

7.4 Plots illustrating the routing algorithm

In this section, we are going to illustrate how the computing of the optimal flight route works in the model. These computations are done by the airline operations control agent, specifically in the transition AO_I1 (section 6.4.1.5), and also by the crew agent in transition CR_SA_G5 (section 6.6.2.5). In this section, we consider a very small example of the airspace. The airspace is described by grid of 9x9 sectors (see Figure 7.3). In three of these sectors, there are departure airport, destination airport and alternate airport labelled *A*, *B* and *C* respectively. Four of the sectors are closed. Based on the position of the airports and the availability of the sectors, the algorithm calculates the planned route of the flight as seen on Figure 7.3.

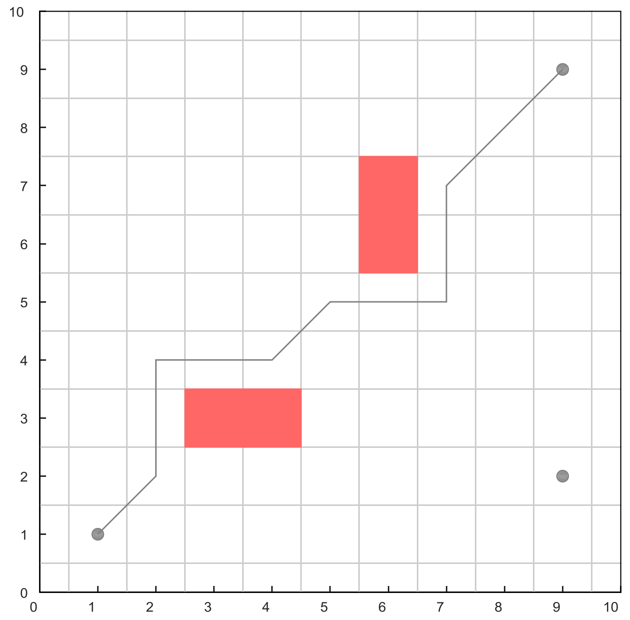


Figure 7.3: flight route planned by AOC considering a specific example of airspace consisting of 9x9 sectors

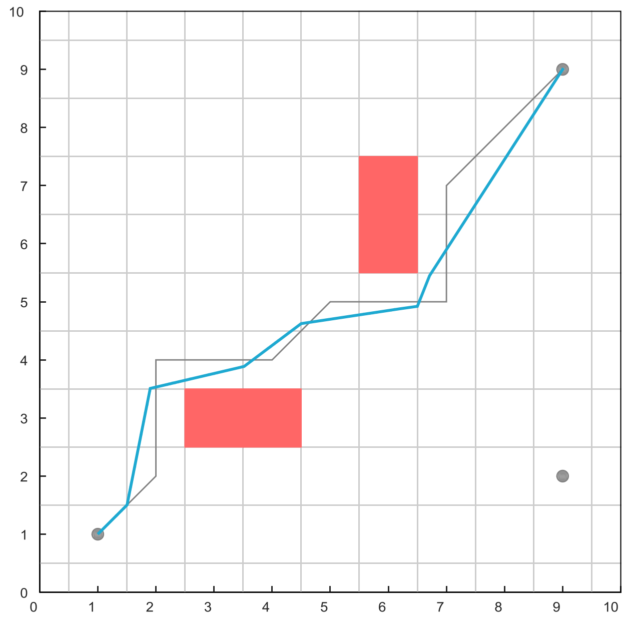


Figure 7.4: planned and actual flight route considering a specific example of airspace consisting of 9x9 sectors

On Figure 7.4, we see the actual trajectory of the flight. The difference between the planned and the actual trajectory is caused by two things. First one is a simplification of the model. We see that the planned route is defined by several points, where the direction of flight is changing. These waypoints are always in the middle of the sector. The way the evolution of the aircraft is designed is that the aircraft flies in the direction of the next waypoint until it reaches the sector in which the waypoint is located. When the aircraft is in the same sector as is the next waypoint, it considers this waypoint as already reached without travelling to the middle of the sector and changes direction to the next waypoint. This can be seen on Figure 7.4 for first five waypoints. In case of the last waypoint before the destination airport, we notice that the aircraft does not even reach the sector of the waypoint, but flies directly to the destination. This happened because the crew of the aircraft updates their information about the airspace at exponential time intervals and every time they do it, they also recalculate the route. This is what happened close after reaching the sector [7, 5]. Since all the sectors between current position and the destination were available, the crew updated the route and headed directly to the destination airport.

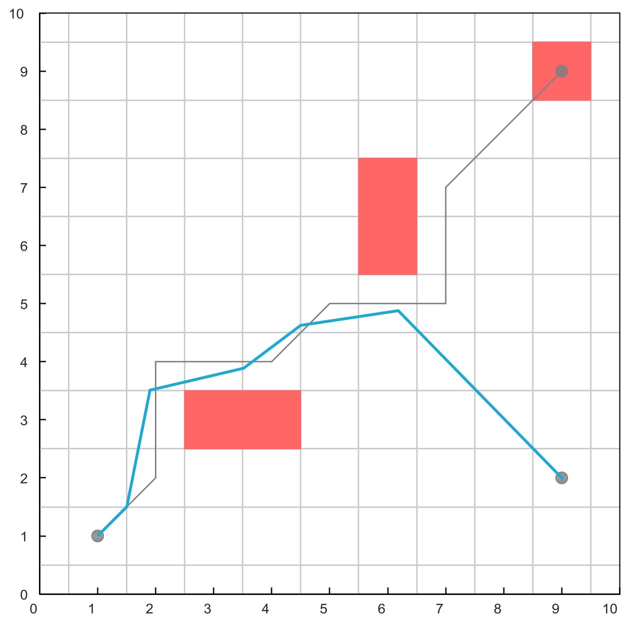


Figure 7.5: Flight route with diversion after the destination airport was closed.

We can see slightly different situation on Figure 7.5. The situation was the same at start of the flight, but approximately in the middle of flight, the sector where the destination airport is located became closed and thus unavailable for the aircraft. So next time the crew updated their information, they realized that they will not be able to land at the destination. In this situation, they make a decision to divert to the alternate airport *C*. After that they update the route and continue the flight until they land at the alternate airport.

8 Simulation results

The objective of this project is to assess risk probabilities of fuel-related events. More precisely, we are going to assess the probability of an aircraft landing with less than FRF and the probability of an aircraft running out of all usable fuel. It is expected that these probabilities are too small to be estimated by regular Monte Carlo simulations. Nevertheless, before implementing the splitting method algorithm discussed in section 5.2, we also performed regular Monte Carlo simulations. The reason was to observe the distribution of the fuel left after the flight and to get an idea of the order of the probabilities we are aiming to estimate. We can also analyse what happened during the flights that had the least fuel left which is important for understanding the results better.

8.1 Regular Monte Carlo simulations

Using regular Monte Carlo method, we calculated 6 simulation runs, each with 50 000 trajectories. By one trajectory, we mean one execution of the model described in chapter 6, in other words, one flight from the planning part, through the whole flight until the end of the flight, where we observed how much fuel was left in tanks of the aircraft. Calculation of one trajectory, using the implementation in Java programming language (as discussed in previous chapter, section 7.1), takes approximately 2/3 of a second. This time highly depends on length of the flight and time step of the simulation. For the simulations, we chose the distance between the departure airport and the destination to be 3000 kilometres as this is the usual trip distance for considered type of aircraft. We used time step equal to 1 second. Then one run of the simulation (50 000 trajectories) takes more than 8 hours to complete.

On the Figure 8.1, we have the histogram of the remaining fuel after the flight from all 6 simulation runs, which is 300 000 results in total. We see that the most common result is a bit less than 2 000 kilograms of fuel. The FRF in this situation is 786.6 kilograms.

We used these results to estimate the probability of reaching less than FRF. The estimates are summarized in Table 8.1. The probability estimates are not always calculated using all 50 000 trajectories. Sometimes, few trajectories are not considered because, for example, the flight was cancelled during the planning phase. This can happen if the algorithm for finding the route is not capable of finding any available route. The algorithm and situations when this can happen are discussed in section 6.4.2.1. That is the reason why in the case of four observations, we have estimate $8.02 \cdot 10^{-5}$ instead of $8.00 \cdot 10^{-5}$. The relative error of the estimator in the table is defined as

$$RE = \frac{\sqrt{\mathrm{Var}(\tilde{\gamma})}}{\tilde{\gamma}}.$$

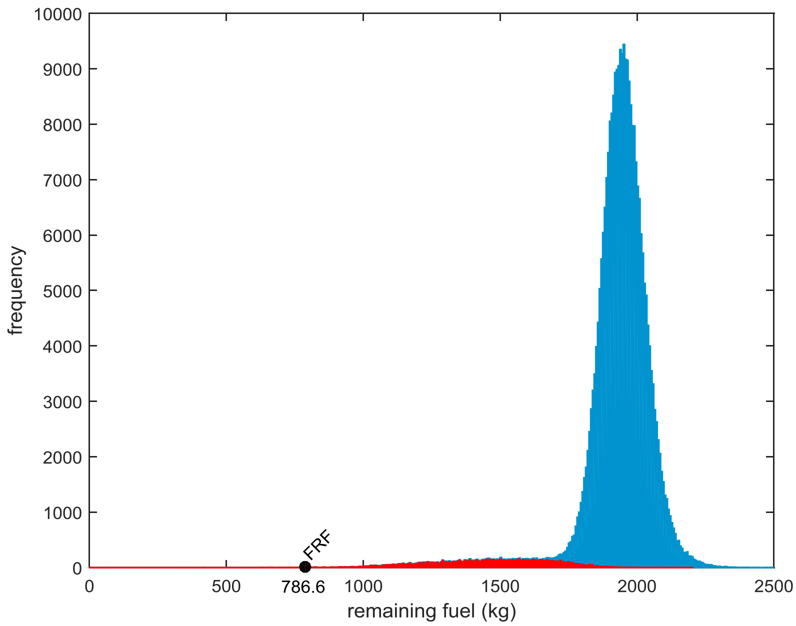


Figure 8.1: Histogram of the remaining fuel left from 300 000 simulated flights. Red highlighted observations are the flights in which the aircraft was sent to holding before landing.

Table 8.1: results of 6 runs of regular Monte Carlo simulations. In this case by observation, we mean the event of having less than FRF.

Results of 6 simulation runs						
simulation ID	1	2	3	4	5	6
number of observations	5	12	8	9	4	4
probability estimate	$1.00\cdot10^{-4}$	$2.41\cdot10^{-4}$	$1.60\cdot10^{-4}$	$1.80\cdot10^{-4}$	$8.02\cdot10^{-5}$	$8.02\cdot10^{-5}$

Total result			
number of observations	probability estimate	relative error	confidence interval
42	$1.40\cdot10^{-4}$	0.215	$\left(1.01\cdot10^{-4}, 1.90\cdot10^{-4}\right)$

We see that for each simulation, we have only few observations of events where the level of FRF was reached. 50 000 trajectories is not enough to get reliable results as can be seen by comparing them with the confidence interval in Table 8.1. If we have an abundance of computational time available, it is possible to avoid using more sophisticated accelerating simulation methods, like splitting method, and use only regular Monte Carlo for estimating this probability. But this is definitely not feasible for estimating the probability of fuel starvation i.e. burning all usable fuel before finishing the flight, since there was not even one observation of this rare event among all 300 000 flights. However we can still get a rough estimate of the order of the probability by extrapolation. To do that, we analyse the dependence between the probability of reaching a certain fuel level and the value of this level. In Table 8.1, we have the probability in case we choose the fuel level threshold equal to FRF. When we choose lower threshold value, the probability of reaching these values is also lower. The dependence is shown on Figure 8.2 as the blue line. We can now approximate this dependence by a proper fitted function. Since we have a relatively high number of observations, the central limit theorem implies that the decline of the probability could be exponential. The red curve on Figure 8.2 is the exponential fit of the blue curve. Assuming this exponential extrapolation, we get that the probability of fuel starvation is of the order 10^{-7}. The numbers on Figure 8.2 are different, because that plot is calculated from the results of simulations with higher parameter for wind variability (discussed in section 8.1.1.2), but the idea is the same. We will later see that this extrapolation is not far from the actual estimate computed by splitting method further reinforcing the hypothesis that the probability decline is exponential. We have not tested this hypothesis. Its purpose is only to give us an idea about the expected order of the probability of fuel starvation.

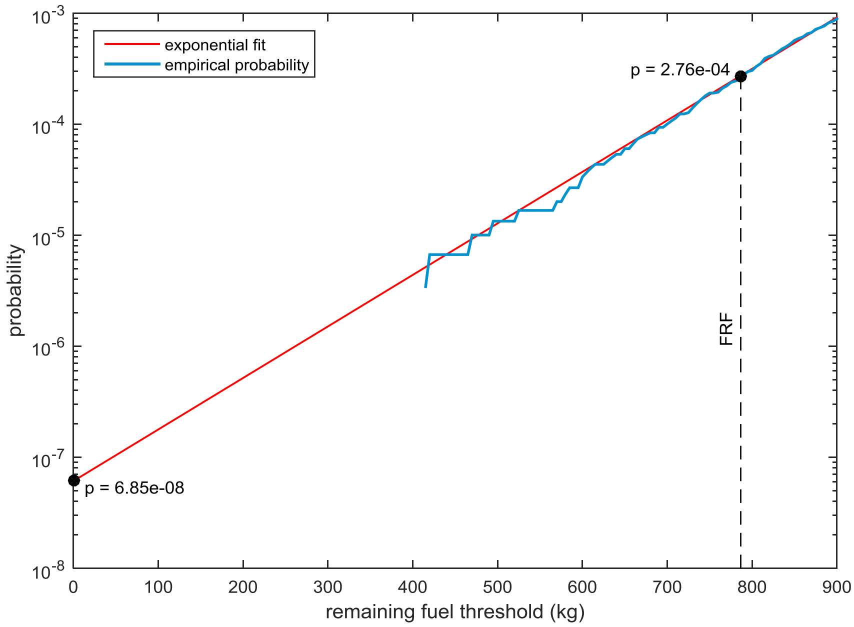


Figure 8.2: Exponential extrapolation (red line) of the dependence between the probability of reaching a certain fuel level and value of this level. This extrapolation is calculated from the results with higher parameter determining the variability of wind direction.

Another important step in analysing the Monte Carlo result is to find out why some flights have considerably less fuel at the end. We can observe that most of the results are concentrated in the approximate interval $[1700, 2200]$. There are very few results that are higher than the upper bound of this interval. These are either flights that had considerably higher tailwind than expected or flight where the aircraft had to divert to closer airport. But the left tail of the distribution is obviously heavier than the right one. This is caused by holding. In our model, there is a probability equal to 0.05 that the aircraft will be sent to holding before landing at the airport. The observations in which the holding occurred are highlighted in red on Figure 8.1. Variance of the holding time is quite high and it can be long from 5 minutes to 1 hour. This explains the high spread of the highlighted results. Then a very simplified explanation of the histogram would be that we have a random variable that is with probability 0.95 equal to a variable with higher mean and lower variance (the blue part of the histogram) and with probability 0.05 equal to a variable with lower mean and higher variance (the red part of the histogram).

8.1.1 Monte Carlo simulations with different parameters

In this section, we will introduce some more results of Monte Carlo simulations. We repeated the simulations using different values of parameters to observe how the distribution of the remaining fuel changes.

8.1.1.1 Monte Carlo simulations with higher probability of holding

We calculated 6 simulation runs, each with 50 000 realizations, using different parameter determining the probability of holding. The original probability is 0.05. Here, we increased it to 0.1. Such high probability is realistic only if we expect that there will be traffic congestion at the destination airport. The histogram of the remaining fuel from all 300 000 flights is on Figure 8.3. On Figure 8.4, we see enlarged version of the left tail of this distribution compared to the distribution with the original parameters. The frequency of results in this range is approximately twice as high. This could have been expected since we doubled the probability of holding. These results are in line with the simplified explanation of the histogram with two random variables described above.
The estimates for probability of reaching less than FRF are summarized in Table 8.2. As expected, the estimate is higher compared to previous simulations.

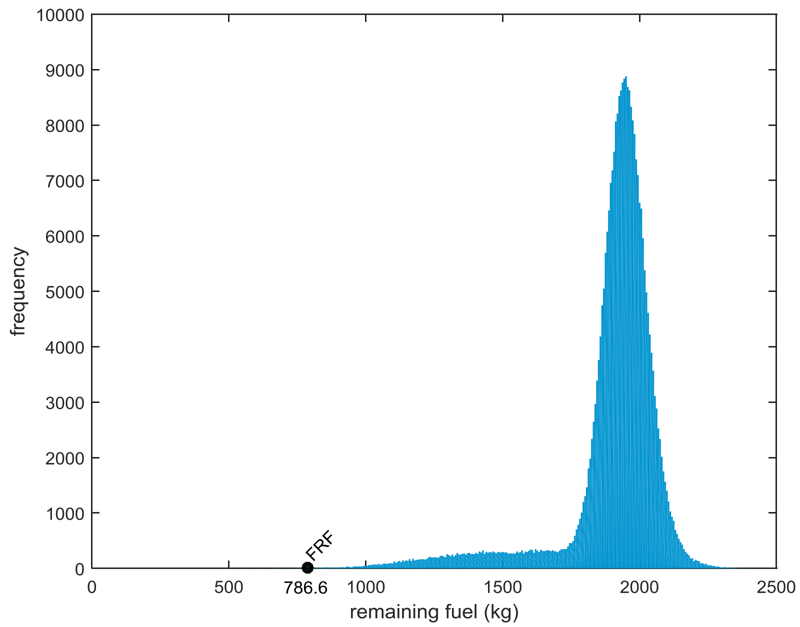


Figure 8.3 histogram of the remaining fuel left from 300 000 simulated flights with higher probability of holding

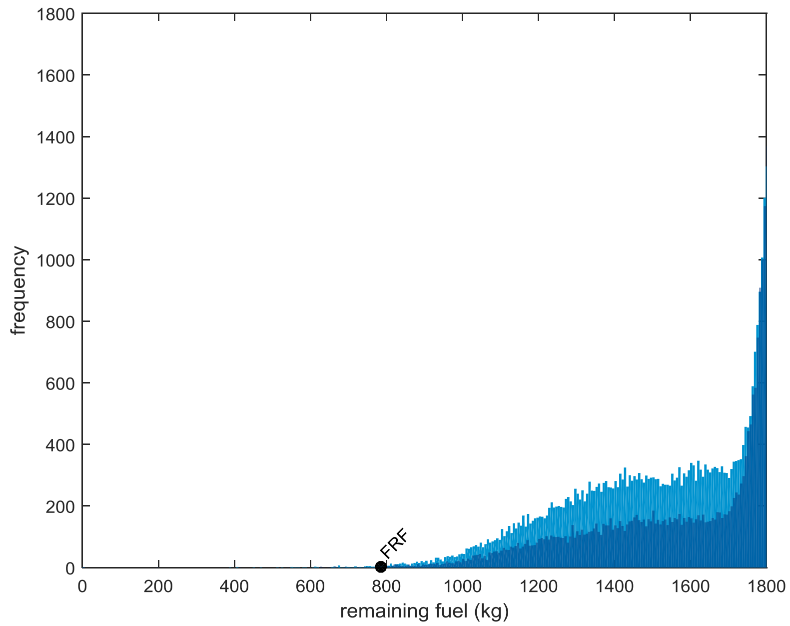


Figure 8.4: This is part of the previous histogram from Figure 8.3. Plot is limited to the left tail of the distribution.

Table 8.2: results of 6 runs of regular Monte Carlo simulations with higher probability of holding. In this case by observation, we mean the event of having less than FRF.

Results of 6 simulation runs						
simulation ID	1	2	3	4	5	6
number of observations	17	11	18	12	16	12
probability estimate	$3.41 \cdot 10^{-4}$	$2.21 \cdot 10^{-4}$	$3.61 \cdot 10^{-4}$	$2.41 \cdot 10^{-4}$	$3.21 \cdot 10^{-4}$	$2.41 \cdot 10^{-4}$

Total result			
number of observations	probability estimate	relative error	confidence interval
86	$2.87 \cdot 10^{-4}$	0.086	$\left(2.30 \cdot 10^{-4}, 3.55 \cdot 10^{-4}\right)$

8.1.1.2 Monte Carlo simulations with higher variability of wind direction

We calculated 6 simulation runs, each with 50 000 realizations, using different parameter determining the variability of wind direction. Original value of the parameter is equal to $\pi/32$. These simulations were done with higher value $\pi/8$. This parameter determines the variance of a normal random variable with zero mean that is added to the current wind direction every time the transition EN_D1 fires (as defined in section 6.2.1.5). This practically means that the prediction about the wind made by AOC before flight is less accurate which leads to higher uncertainty in fuel consumption. As we can see on Figure 8.5, this leads to higher variance of the remaining fuel. Also the estimates of probability of reaching less than FRF are higher as stated in Table 8.3.

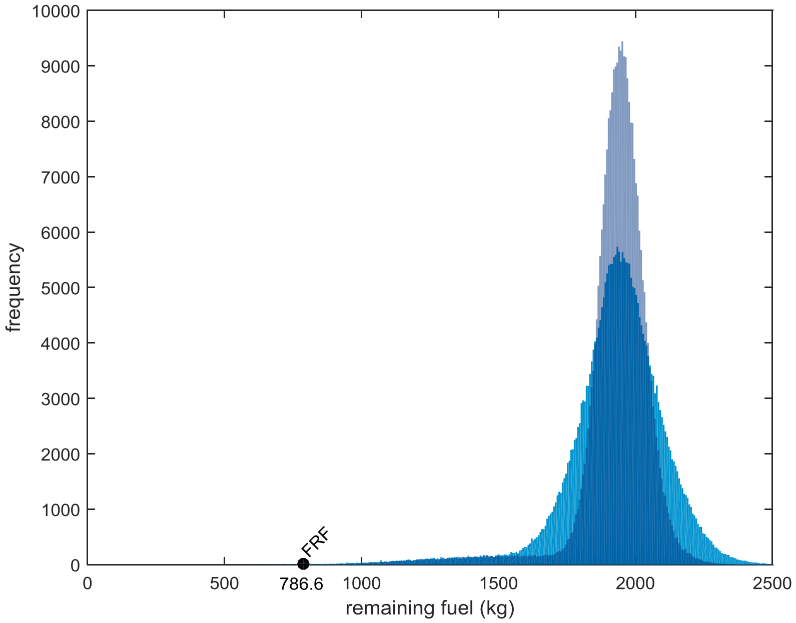


Figure 8.5: histogram of the remaining fuel left from 300 000 simulated flights with higher variability of wind direction (light blue) compared to the histogram calculated using original parameter values (transparent dark blue)

Table 8.3: results of 6 runs of regular Monte Carlo simulations with higher variability of wind direction. In this case by observation, we mean the event of having less than FRF.

Results of 6 simulation runs						
simulation ID	1	2	3	4	5	6
number of observations	18	15	11	13	8	10
probability estimate	$3.61 \cdot 10^{-4}$	$3.01 \cdot 10^{-4}$	$2.21 \cdot 10^{-4}$	$2.61 \cdot 10^{-4}$	$1.60 \cdot 10^{-4}$	$2.41 \cdot 10^{-4}$

Total result			
number of observations	probability estimate	relative error	confidence interval
75	$2.51 \cdot 10^{-4}$	0.118	$\left(1.97 \cdot 10^{-4}, 3.14 \cdot 10^{-4}\right)$

8.2 Splitting method simulations

In this section, we will present the results of simulations done using the splitting method as well as the procedure how we got the results. As discussed in chapter 5, we use this method to get estimates of low probabilities that would be impossible to get with regular Monte Carlo simulation within reasonable computing time. Following the definitions and notation from chapter 5, we have a piecewise deterministic strong Markov process $X = \{X(t) \in S, t \geq 0\}$ equivalent to dynamically coloured Petri net model described in chapter 6. We aim to estimate the probability $\gamma = \mathrm{P}(T_B \leq T)$ which is the probability of reaching certain low fuel level before stopping time T. In our situation, the stopping time T is defined as the time when the flight ends, more specifically when the transition AC_EV_G6 (defined in section 6.5.4.5) from aircraft evolution LPN fires.

To use the splitting method, we first have to determine the importance function $h: S \to \mathbb{R}$ and levels $L_0 \leq L_1 \leq \ldots \leq L_n$.

8.2.1 Determining the importance function and levels

Defining an importance function for the splitting method that provides an efficient algorithm is not a straightforward process. It is quite trivial to define levels that are theoretically correct, but it often proves to be a challenge to set levels in such a way that they produce desired results after reasonable time without huge variance of the estimator or without estimate being equal to zero, because there was no hit to one of the levels. Here, we will describe the steps that led to final version of the levels used in the simulations. We are going to describe four different approaches on how to define the importance function and levels. First two are theoretically correct, but not efficient. Third approach seems to solve the problems of two previous attempts, but is incorrect. We will explain what is wrong with the approach and why is it important to understand. The final fourth approach is correct and efficient. We used it for the simulations.

8.2.1.1 First approach – naive importance function

First idea on how to define the importance function is the following. Using the splitting method, we want to estimate probability of reaching very low fuel levels. So we set the importance function

$$h_0(x) = m_f,$$

where $m_f = m_f(x)$ determines the amount of remaining fuel at a specific state $x \in S$. This is a very naive approach, because it does not take into account the typical behaviour of the process X and the function $m_f(x)$. On Figure 7.2, we can see the amount of remaining fuel at a specific time of flight, in other words, function $m_f(X(t))$, where $t \in [0, T]$. This function is always decreasing in time. Let $\tilde{m}_f(T)$ denote the average amount of remaining fuel at the end of the flight. Now consider that we set the first level L_0 to value higher than $\tilde{m}_f(T)$. Then almost every trajectory we simulate will hit the first level. That means that this level is useless because it does bring us significantly closer to the rare event. Now, on the other hand, we consider the first level being set to value lower than $\tilde{m}_f(T)$. Then all the hits we get for this level would be very close to the end of the flight which means that the probability of reaching next levels would be too small. Also by choosing the level like this, we ignore what happens during the flight. For example if fuel consumption is significantly higher than average in the beginning of the flight, there is still no chance that this trajectory hits the first levels until much later in the flight. This is inefficient because if we would be able to identify the higher fuel consumption as soon as it occurs, we could save the state of the flight as the entrance value of the first level. Then by resampling this state there would be higher chance of something like this happening again later in the flight, bringing us even closer to the rare event.

Using this importance function is inefficient because we cannot distinguish whether the low fuel has been reached because of some interesting event that increased the fuel consumption above average or just because we are at a very late stage of flight in which the fuel level is always this low. This leads to the idea of incorporating the current position of the aircraft into the importance function.

8.2.1.2 Second approach – inefficient importance function

To incorporate the current position of the aircraft, we developed a more sophisticated function that can be used to determine the importance function. We define function

$$f(x) = \frac{m_{fAO}(d) - m_f}{m_{fAO}(d_0) - m_{fAO}(0)}.$$

Function $d = d(x)$ determines the current distance of the aircraft from the destination airport at state $x \in S$ and d_0 is equal to distance of the departure airport from the destination airport. Function $m_{fAO}(d)$ specifies the amount of remaining fuel at a specific distance d from the destination according to the flight plan calculated before the flight. This function is calculated in transition AO_I1 and is represented by a matrix $m_{fAO,d}$ (described in section 6.4.2.2). Then $f(x)$ is the difference of the planned remaining fuel and the actual remaining fuel relative to the planned fuel consumption for whole trip at state $x \in S$. If $f(x) > 0$, it means that the aircraft is burning more fuel than expected. On the other hand, if $f(x) < 0$, fuel consumption is lower than planned. Unlike function h_0, function f enables us to identify the cases where the fuel consumption is above the average at any point of the flight, which we can use to determine the importance function and levels more efficiently.

The target while choosing levels was to have the transition probabilities $\hat{p}_k$ of reaching a level from the previous level as close to 0.2 as possible. This number was chosen based on a theoretical analysis done in (Amrein & Künsch, 2010). The proposition 5.2 in this paper states that $p_{opt}^a \approx 0.2032$ is the optimal transition probability. This came out as a result of an optimization problem to minimize the computational effort while having a constraint to relative error of the estimator. To get this and other theoretical results, they made a simplifying assumption that

$$P\left[A_k \mid A_{k-1}, \left(t_{k-1}, x_{k-1}\right)\right] \equiv p_k \qquad \left(\forall \left(t_{k-1}, x_{k-1}\right), \forall k\right),$$

where A_k is the event of reaching k -th level and $\left(t_k, x_k\right)$ are the entrance time and entrance value for this level. This assumption says that the probability of reaching level k given that we are currently on level $k-1$ is equal to constant p_k regardless of the specific position and time within the level $k-1$. And that this holds for all levels. In other words, the levels are defined such that it does not matter in what situation we reach a level, the probability of reaching the next level is always the same. Unfortunately this is not true in case of our model. Take, for example, these two situations of reaching the same level. In first situation, the aircraft was just sent to holding above the destination airport for next 30 minutes. In the second situation, aircraft has reached level while landing 10 meters above the ground. It is clear that in the first situation, the aircraft is going to fly for at least next half an hour during which it will burn even more fuel, while in the second situation, the flight is going to end in next few seconds having basically no chance to burn extra unplanned fuel. To satisfy this assumption, we would have to take into account every possible situation while setting the levels and even then it is not guaranteed whether the assumption can be satisfied because of the existence of discrete modes in our model (jumps of the piecewise deterministic Markov process). One example of the discrete mode is that every time the aircraft starts to descend, it is decided whether the crew will perform a missed approach or not with fixed probabilities corresponding to each decision.

The fact that this assumption does not hold in our situation does not mean that the splitting method does not work. It only means that it is not always optimal (or even possible) to have the transition probability for reaching the next level equal to 0.2. Also the upper and lower bounds for relative error that are derived in chapter 5 of (Amrein & Künsch, 2010) do not hold.

Getting back to determining the importance function and levels within our model, one possible way is to set the importance function $h_1(x) = f(x)$ and then setting levels $L_0 \leq L_1 \leq \ldots \leq L_n$ accordingly such that the transition probabilities are close to 0.2 and that we reach the wanted rare event by hitting the last level.

The process of determining the specific values of levels is a process of trial and error. We start by setting the first level L_0 and if the value repeatedly produces desired transition probability, we move on to setting the value of next level L_1.

The example of setting the levels using function h_1 can be seen on Figure 8.6. On the x-axis, we have the distance $d(x)$ between the aircraft and the destination airport and we have the values of function $f(x)$ on y-axis. The black line marks the value of function f along the flight. Flight starts at point $\left[d_0, 0\right]$. The distance from destination is $d_0 = 3000$ kilometres and the value of function

$f(x)=0$, because the actual fuel consumption at the start of flight is equal to the planned fuel consumption. The flight ends with $d(x)=0$. It is important to realize that the black line is not a plot of a function, but a curve on a two dimensional plane. It can happen that the distance $d(x)$ is equal to the same value at several times of the flight. It is visible on Figure 8.6 that the aircraft was repeatedly flying back and forth getting closer to the airport and then again farther just before landing. In this situation, the aircraft was flying in the holding pattern. The colourful horizontal lines are five levels. So in this example, the trajectory hit the first level at the beginning of the flight and then second, third and fourth just before the end.

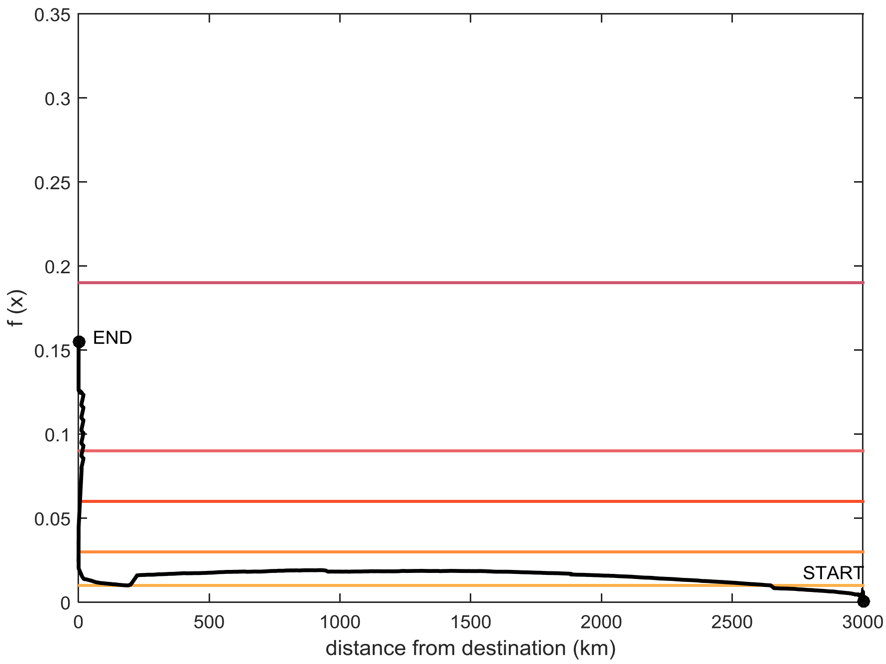


Figure 8.6: Example of one possible way of setting levels. The black line labels the development of values of function f. The horizontal lines are the levels.

Using h_1 as the importance function is essentially correct provided we simulate enough trajectories. But even after a lot of experiments with different values of levels, we did not get satisfactory results. Either we did not get to low enough fuel level or the variance of our estimator was too high. The problem with very high variance was that even when we set the levels in a way that we got reasonable transition probabilities for all levels in most of the simulations, it still often happened that the simulation ended with zero hits to the last level resulting in useless estimator.

As we see on Figure 8.6, the value of function $f(x)$ does not change much during the flight, but increases rapidly just before the end of the flight. This is a very usual development. The rapid increase happens when the total fuel consumption exceeds the total planned fuel consumption $m_{fAO}(d_0)-m_{fAO}(0)$. This characteristic is not reflected by h_1 as the importance function and this leads to ineffective algorithm. The reason why current levels are not efficient enough is similar to the reason why using the importance function h_0 does not work. We want the first level to be hit at earlier part of the flight. So it cannot be set too high. But then we often have trajectories with below-the-average fuel consumption along the whole flight with short holding period at the end that manage to hit the first level just before the end of the flight. These trajectories are saved and resampled at later stage of the simulation even though they are useless because the probability of this trajectory reaching higher levels is almost zero. Then even if we set the number of successes to 200, only a fraction of those trajectories have realistic chance of ever reaching the rare event. And low variability of the trajectories hitting the rare event naturally results in high variance of the estimator.

8.2.1.3 Third approach – incorrect importance function

In this section, we will introduce one example how to deal with this complication. This example is actually incorrect and the splitting method does not work when implemented. But we will introduce it anyway, because it is insightful to understand why the example does not work and it can prove helpful in further research.

In this example, we will follow up on the levels described above using the function h_1. The idea is to set a co-called *distance condition* for each level, meaning that each level can be reached only before reaching certain specified distance from the destination. So, for example, the first level L_0 can be reached only if the aircraft is farther from the destination than $2500\ km$. The second level L_1 can be reached only if the aircraft is farther from the destination than $2000\ km$ and so on. This solves the problem when lower levels were reached too close to the end of the flight. Now all the trajectories that hit the first level have reasonable chance to hit the rare event.

But if we implement this algorithm and run simulations, we see that the probability estimates are very low. For example the probability of reaching less than FRF can be as low as 10^{-8} which is extremely low compared to result from regular Monte Carlo which is of the order 10^{-4}.

The reason for this inconsistency is that by setting the "distance condition", we restrict the simulation to consider only a subset of all trajectories that can reach the rare event. This leads to a significant underestimate of the final probability.

We can also look at it from a different point of view. When we say that the first level is reached only if the function $f(x) \geq L_0$ before getting closer to the destination than $2500\ km$, it is equivalent to saying that the first level is reached only if $f(x) \geq L_0$ for $d(x) \geq 2500\ km$ or $f(x) \geq M$ for $d(x) < 2500\ km$, where M is very high value that cannot be reached by $f(x)$. We can see the example of this on Figure 8.7. It is quite obvious what is wrong with this example from the figure. The sets $B_0, B_1, \ldots, B_n$ defined by these levels are not a decreasing sequence of sets. This was the fundamental assumption used in section 5.2.1. Without this assumption, the whole idea of splitting method falls apart and the method provides incorrect results.

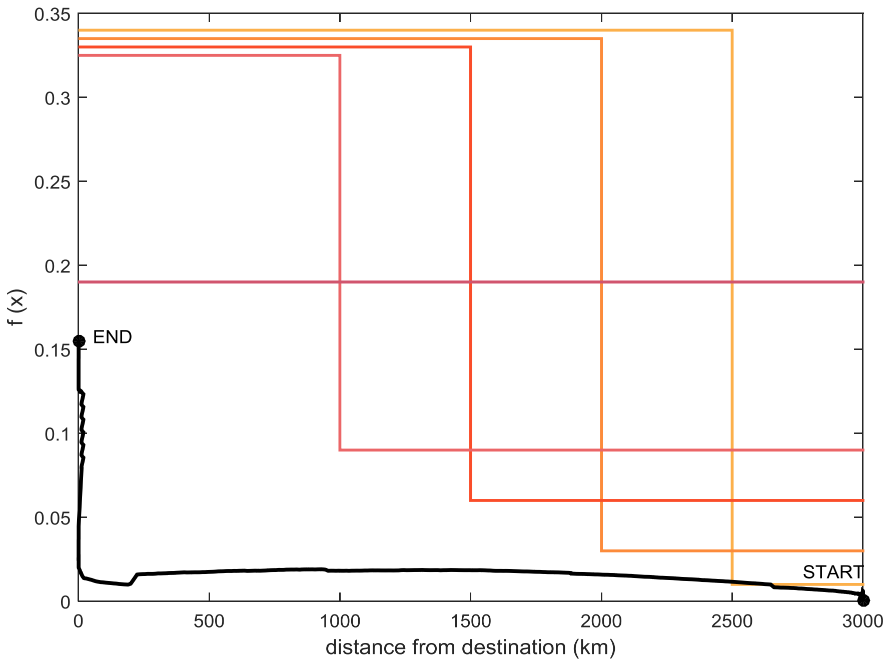


Figure 8.7: Example of setting levels using the distance condition. The black line labels the development of values of function f. The colourful lines are the levels.

8.2.1.4 Fourth approach – correct and efficient importance function

Now based on previous sections, we know that we have to define an importance function and levels that will produce a decreasing sequence of sets $B_0 \supset B_1 \supset \ldots \supset B_n$ and also reflect usual behaviour of function f including the steep increase before the end of the flight. The correct form of the importance function that we finally used to get results is following.

We define function $l_k : \mathbb{R} \to \mathbb{R}$ corresponding to each level. We say that the k-th level is reached if $f(x) \geq l_k(d(x))$. That means that functions l_k are contour lines of the importance function $h: S \to \mathbb{R}$ corresponding to level values L_k for $k = 0, 1, \ldots, n$. The explicit expression of the importance function is not important so we define function h implicitly by defining the contour lines l_k. These are defined as piecewise linear functions of the distance d of the aircraft from the destination (in meters) with two linear sections for $k = 0, 1, \ldots, n-1$:

$$l_k(d) = \begin{cases} a_{k1} + b_{k1} \cdot d , & d \in [0, D_1) \\ a_{k2} + b_{k2} \cdot d , & d \in [D_1, D_2] \end{cases}$$

The maximum of the function is in 0, followed by rapid decline of the first linear part corresponding to $b_{k1} < 0$. The second linear part has also negative slope $b_{k2} < 0$, but it is very close to zero. This definition reflects the behaviour the function f as can be seen on Figure 8.8. The last contour line is defined as a horizontal line:

$$l_n(x) = L_n \quad \forall x \in \mathbb{R} .$$

Purpose of this is to know the lower bound of the value of f for all trajectories that hit the last level. Since each function l_k consists of two linear segments, it can be unambiguously defined by three points. These points are $[0, A_k]$, $[D_1, B_k]$ and $[D_2, C_k]$. We get the definition of the function from these points using the following formulas:

$$a_{k1} = A_k \qquad b_{k1} = \frac{B_k - A_k}{D_1}$$

$$a_{k2} = \frac{B_k D_2 - C_k D_1}{D_2 - D_1} \qquad b_{k2} = \frac{C_k - B_k}{D_2 - D_1}$$

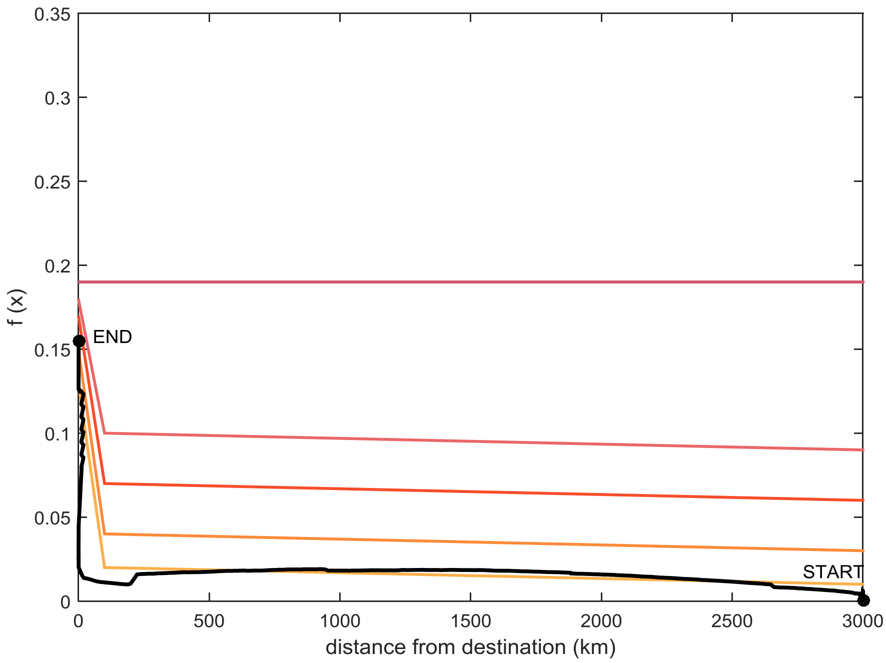


Figure 8.8: Example of setting levels using contour lines of the importance function. The contour lines are defined as piecewise linear functions with two linear segments.

This importance function provides correct results with smaller variance. But we still need to make a small adjustment to get to probability of having less than FRF remaining and the probability of fuel starvation. Using the importance function h described above, we know that all trajectories that hit the last level satisfy that $f(x) \geq L_n$. But this fact does not gives us the actual amount of remaining fuel, because the function m_{fAO} used to define f can be different for each flight. To get the desired probability, we adjust the condition of hitting the levels. For simulations where the result is the estimate of probability of having less than FRF remaining, we define that k-th level is reached if

$$f(x) \geq l_k(d(x)) \text{ or } m_f(x) \leq m_{fFR} \text{ for } \forall k,$$

where m_{fFR} is the FRF calculated by AOC agent. This value is constant for all simulations.

For simulations where the result is the estimate of probability of fuel starvation, we define that k -th level is reached if

$$f(x) \geq l_k(d(x)) \text{ or } m_f(x) \leq 0 \text{ for } \forall k .$$

Specific definitions of functions l_k used for the simulations and the results are presented in the following section.

8.2.2 Results of the splitting method simulations for probability of having less than FRF remaining

For estimating the probability of reaching fuel level lower than FRF, we used the variation of the splitting method called the fixed number of successes. We set the number of successes equal to 200 . As we discussed in previous section, determining the specific values of the levels $L_0 \leq L_1 \leq \ldots \leq L_n$ is not a trivial process. After a lot of simulations, we implicitly defined the importance function h by its contour lines $l_k(d)$ at the values of the levels. The contour lines are defined by these three points: $[0, A_k]$, $[D_1, B_k]$ and $[D_2, C_k]$. We set $D_1 = 5 \cdot 10^4$ and $D_2 = 3 \cdot 10^6$, which corresponds to the points where the aircraft is $50\,km$ and $3000\,km$ from the destination. The specific values of A_k, B_k and C_k are listed in Table 8.4.

Table 8.4: specification of the coefficients defining the contour lines of the importance function used for estimating the probability of having less than FRF remaining

k	$A_k = l_k(0)$	$B_k = l_k(5 \cdot 10^4)$	$C_k = l_k(3 \cdot 10^6)$
0	0.15	0.017	0.007
1	0.151	0.023	0.013
2	0.152	0.026	0.016
3	0.16	0.03	0.02
4	0.161	0.04	0.03
5	0.17	0.05	0.04
6	0.18	0.18	0.18

Using these parameters, one simulation run takes approximately 1 hour to finish and simulates 200 trajectories that hit the rare event. That is a significant improvement from regular Monte Carlo simulation that takes approximately 8 hours to finish and produces, on average, only 7 trajectories that hit the rare event.

In Table 8.5, we have the final results of 60 simulation runs and the transition probabilities for each level for 5 out of 60 simulation runs.

Table 8.5: Results of 60 splitting method simulations for estimating the probability of having less than FRF remaining. The transition probabilities for each level are the mean probabilities based on all 60 simulations.

simulation ID	1	2	3	4	5
k	$\hat{p}_k$	$\hat{p}_k$	$\hat{p}_k$	$\hat{p}_k$	$\hat{p}_k$
0	0.195	0.200	0.202	0.203	0.194
1	0.173	0.177	0.176	0.181	0.201
2	0.478	0.241	0.452	0.318	0.356
3	0.276	0.355	0.489	0.198	0.245
4	0.198	0.060	0.299	0.392	0.027
5	0.427	0.977	0.649	0.683	0.728
6	0.465	0.374	0.040	0.706	0.424
$\hat{p}$	$1.747\cdot10^{-4}$	$6.501\cdot10^{-5}$	$6.154\cdot10^{-5}$	$4.373\cdot10^{-4}$	$2.892\cdot10^{-5}$

Total result	
probability estimate	relative error
$1.599\cdot10^{-4}$	0.152

The result lies in the confidence interval made by the regular Monte Carlo simulations in section 8.1 which is $\left(1.01\cdot10^{-4}, 1.90\cdot10^{-4}\right)$.

8.2.3 Results of the splitting method simulations for probability of fuel starvation

For estimating the probability of fuel starvation, we used a combination of the fixed number of successes and fixed number of effort. That means that we set the number of successes to 500 , but we also set high upper limit for the computational effort. This limit was set to maximum of $60\,000$ simulated trajectories for each level. For most of the simulation runs, the maximum number of simulated trajectories was by far lower than this limit, but for some simulations we did not achieve the targeted number of successes within this limit.

As in previous section, the importance function h is implicitly defined by its contour lines $l_k\left(d\right)$ at the values of the levels. The contour lines are defined the same way as in previous section by these three points: $\left[0, A_k\right]$, $\left[D_1, B_k\right]$ and $\left[D_2, C_k\right]$, where $D_1 = 5\cdot10^4$ and $D_2 = 3\cdot10^6$. The specific values of A_k, B_k and C_k are listed in Table 8.6.

Using these parameters, one simulation run takes approximately 6 to 8 hours to finish. As we mentioned, most of the simulations end with 500 hits to the rare event. The probability of this rare event is too low to even try estimating it with regular Monte Carlo. If we consider that one trajectory of regular Monte Carlo simulation was calculated in approximately 2/3 of a second, the expected computation time to get one hit to the rare event of fuel starvation is about 7 months.

In Table 8.7, we have the final results of 43 simulation runs and the transition probabilities for each level for 5 out of 43 simulation runs.

Table 8.6: specification of the coefficients defining the contour lines of the importance function used for estimating the probability of fuel starvation

k	$A_k = l_k(0)$	$B_k = l_k(5\cdot10^4)$	$C_k = l_k(3\cdot10^6)$
0	0.15	0.017	0.007
1	0.151	0.023	0.013
2	0.152	0.026	0.016
3	0.16	0.04	0.03
4	0.161	0.06	0.05
5	0.18	0.09	0.07
6	0.21	0.11	0.09
7	0.24	0.13	0.11
8	0.3	0.3	0.3

Table 8.7: Results of 43 splitting method simulations for estimating the probability of fuel starvation. The transition probabilities for each level are the mean probabilities based on all 43 simulations.

simulation ID	1	2	3	4	5
k	$\hat{p}_k$	$\hat{p}_k$	$\hat{p}_k$	$\hat{p}_k$	$\hat{p}_k$
0	0.197	0.200	0.199	0.202	0.201
1	0.194	0.198	0.200	0.200	0.202
2	0.376	0.398	0.396	0.366	0.400
3	0.096	0.039	0.092	0.134	0.052
4	0.512	0.365	0.244	0.539	0.903
5	0.560	0.019	0.017	0.176	0.019
6	0.0005	0.597	0.080	0.075	0.618
7	0.074	0.674	0.654	0.290	0.388
8	0.718	0.0009	0.217	0.007	0.007
$\hat{p}$	$1.073\cdot10^{-8}$	$1.652\cdot10^{-9}$	$6.790\cdot10^{-8}$	$2.984\cdot10^{-8}$	$2.566\cdot10^{-8}$

Total result	
probability estimate	relative error
$3.647\cdot10^{-8}$	0.273

The result is not very far from the estimate made by extrapolation from the results of regular Monte Carlo simulation (see section 8.1) which is $1.88\cdot10^{-7}$. This does not imply that regular Monte Carlo is appropriate method to estimate these probabilities, but it supports the hypothesis that the decline of the probability discussed in 8.1 is exponential.

9 Recommendations for further research and discussion about the results

While developing the model, the goal was to develop the first version of the model that includes the most important parts and principles of the operation and the most significant hazards related to fuel management like unexpected wind, possibility of diversion or long holding time. This goal has been reached and the model provides reasonable results using the splitting method simulation. Current version of the model has not yet been validated by statistical data about fuel-related incidents or accidents. The model has to be extended significantly before a meaningful comparison to real operations can be made.

Purpose of this chapter is to describe these possible extensions and provide guideline for further research.

9.1 Recommendations for extending the Petri net model

In this section, we will describe the possible extension that can be made to the Petri net model described in chapter 6. We will start with the most important ones based on the experience gathered during this project.

9.1.1 Inclusion of the Air traffic control agent

First recommendation is to include the sixth agent for ATC. There are a lot of hazards in Table 4.1 related directly to ATC. Some of these hazards are already included in the model within different agents, but a separate agent would allow us to include much more of them. Another important thing is that ATC make important decisions that affect the flight. For example the trajectory of the flight has to be approved and can be changed by ATC. This agent would enable more complicated decision-making within the model.

One example is the decision to send the aircraft to holding before allowing it to land. Current implementation in the model is that when the aircraft approaches the airport $i \in \{1, \ldots, N_{AP}\}$, it is sent to holding with probability p^i_{hold}. This probability is a fixed parameter of the airport agent. In reality, this decision is affected by a lot of factors. The important one in our situation is the current fuel level of the aircraft. If the aircraft has low fuel level (close to FRF or lower), it is given a priority from the ATC. So there would have to be extremely high traffic in the situation where an aircraft with less than FRF left is sent to holding before being allowed to land. One possible way to improve this would be to make the probability of holding a colour variable of the ATC local Petri net and make it dependent on the current fuel level of the aircraft.

Second example is the unavailability of the airspace sectors. An airspace sector can be closed for various reasons, but it is often the case that this also a decision made by ATC.

9.1.2 Inclusion of hazards related to fuel system

The most serious accidents related to fuel management (some of the examples are mentioned in chapter 12.2 Appendix B) were often related to malfunction of the fuel system. Also the hazard clusters *C1* and *C2* (see Figure 4.1) are mainly related only to the fuel system. These hazards include, for example, fuel leakage, malfunction of the fuel distributing system making fraction of the fuel in tanks inaccessible or pre-maintenance error. This extension is especially important for making the estimate of probability of fuel starvation more realistic.

These and other hazards can be included by extending the fuel system local Petri net AC_FS or adding a separate local Petri net for fuel management system (FMS).

9.1.3 Implementation of more sophisticated wind model

Wind and the wind prediction is a crucial factor in this risk assessment. Most of the flight planning in reality is affected by the wind prediction, for example the optimal route is determined heavily by wind (jet streams). The accuracy of the wind prediction determines how well the airline operations control can calculate the required trip fuel. Meteorological models and accurate wind predictions are a very complex topic and the AOC uses sophisticated software to calculate the flight plan.

The wind model in current version of the model is very simple (defined in section 6.2.1) and so also the predictions are simple. The prediction in our model is equal to the current state of the airspace because both the wind direction and wind speed follow an independent random walk with zero drift. The most important improvement to current wind model is to introduce a correlation between adjacent airspace sectors. Now, the wind direction and speed has the same values for all sectors at start of the simulation, so the values are relatively close for all sectors through the whole simulation. But it can happen that there are two adjacent sectors that have very different values, so the aircraft experiences an unrealistic sudden change of the wind while crossing the border of these sectors.

9.1.4 Implementation of more sophisticated decision-making of the crew

As we can observe on the examples of fuel-related incidents and accidents listed in chapter 12.2 Appendix B, the decisions made by the crew during the flight were often the key factor that determined final outcome of the situation. For example in case of the Ryanair incident in 2010 (section 12.1), the investigation concluded that the decision to perform the second missed approach and then divert to Valencia (even though there was a different closer alternate available) were the main causes for landing with less than FRF remaining. Another example would be the Hapag-Lloyd accident in 2000 (section 12.2.5), where after realizing that there is less than 1.3 metric tons of fuel left in tanks, the crew decided to head toward Vienna (220 km away) even though the Zagreb airport was only 75 km away. As a result, the aircraft ran out of fuel 20 km before Vienna and landed on the grass 500m before the runway.

One of the things that can be improved is the decision to perform a missed approach. In current version of the model, it works similarly to the decision whether the aircraft will hold. Before the aircraft will land, it performs a missed approach with probability p_{MA}. It can perform more than one missed approach. In reality, the pilot takes into account several factors when deciding whether to perform missed approach including the current fuel level.

9.1.5 Improving the routing algorithm

The routing algorithm that is currently implemented in the model can find a flight route if the airspace has only a small fraction of the sectors closed. We performed simulations where approximately 1% of the sectors were closed. In this setting approximately 0.2% of the simulations were stopped due the insufficient routing algorithm. As I discussed in section 6.4.2.1, it is unlikely that this has significant effect on the overall probability estimate. But it is something that has a lot of possibilities for improvement and it can become more important in future versions of the model. Especially, if there will be more complicated structure of the airspace and closed sectors implemented.

9.1.6 Considering more than one aircraft in the model

Currently the whole model simulates the flight of only one aircraft. Natural extension is to include several aircraft flying different flight routes at the same time. Among other things, this would enable us to model high traffic including traffic congestions. One of the situations that can be modelled is that when a big airport hub gets closed for some time (e.g. because of strong storm) a lot flights are diverted to adjacent smaller airports. This can lead traffic congestion and long holding times above these airports. Also, in reality, each airspace sector has a limit of how many aircrafts can fly through it at one time. This aspect can be also included using more than one aircraft in the simulation.
In this case it is important to consider the added value of more aircraft, because it significantly affects the computation time of the simulation since most of the computations have to be done separately for each aircraft.

9.2 Discussion of the results

Before developing the model itself, we constructed a list of 150 hazards related to fuel management. We included the most important hazards in the model, but there are still some hazards from the list that are not included and a lot of hazards are included only implicitly (for example hazards like H022, H032, H058, H061, H080 are all related to ATC and they are implicitly included in the model by the possibility of long holding time). This fact can lead to conclusion that the overall risk probability is underestimated. But the situation is not that easy so we cannot jump to a conclusion like this before considering other aspects that should be included in future versions.
As we can observe by analysing the simulation results, one of the probable scenarios of reaching very low fuel level within the model is following. Before the aircraft lands at the airport, it is sent to holding (the decision is made in the airport agent AP). The holding time can be very short (only few

minutes), but can be also very long (up to one hour). Such long holding times are extremely improbable, but even if the aircraft holds for 30 minutes, it burns a lot of unexpected fuel. After holding, there is a chance of performing missed approach and if the aircraft performs missed approach, then it can be sent to holding again with the same probability as before. As mentioned in previous section, these decisions are not affected by the amount of remaining fuel. So if we implement more complex decision-making algorithms that will exclude situations, where the aircraft is sent to holding even though it does have enough remaining fuel for it, we will significantly reduce the computed probability estimate.

Consider the result for reaching fuel level lower than FRF. The result of the simulations is that this probability is $1.599 \cdot 10^{-4}$. So according to this, approximately one out of 6250 flights ends with less than FRF. We described the incident where this happened, the Ryanair incident in 2010 (section 12.1). After this event, there was an extensive investigation. The report (Comisión de Investigación de Accidentes e Incidentes de Aviación Civil, 2013) mentioned other cases of this happening after the incident (e.g. the incident in July 2012), but there was only a few of them. So considering this and the amount of flights that take place every day, the result of our simulations is unrealistically high.

As a conclusion, the risk estimates of future versions of the model with the extensions proposed in previous sections are expected to be lower than the current results.

10 Conclusion

The main objective of this study was to develop a model that can be used to estimate the probabilities of fuel-related events. This objective was reached. We developed the first version of the model together with all the aspects necessary to perform a risk assessment analysis. The current version of the model simulates the main steps of the operation within the context of fuel management including the planning and the actual flight in a general setting. The model includes a lot of parameters that can be adjusted to a specific situation. For example, we can use any arrangement of any number of airports or we can run the simulations for different aircraft type with turbine engines, provided we have the corresponding parameters from Base of Aircraft Data (Eurocontrol Experimental Centre, 2011). We can even analyse specific scenarios by setting certain events (like airport closure, unexpected sudden change of wind or fuel system malfunction) to happen at fixed or stochastic times during the flight.

This model was developed using the TOPAZ safety risk assessment cycle (described in section 1.3). This methodology provides steps to perform a complete safety risk assessment. Our goal was to complete first five steps of the cycle. This among other things includes completing the following tasks:

- We made a list of 150 hazards related to fuel management and categorized them based on their effects on the safety of the operation.
- Using the dynamically coloured Petri nets, we constructed the model.
- We implemented the model using the Java programming language.
- We implemented the algorithms for regular Monte Carlo simulation and the splitting method.
- We performed the simulations to estimate the probability of two specific fuel-related events and analysed the results.

Altogether this provides a substantial basis for further research of agent-based dynamic models of safety risk assessment in the context of fuel management.

11 Bibliography

Amrein, M., & Künsch, H. R. (2010). *A Variant of Importance Splitting for Rare Event Estimation: Fixed Number of Successes.* Seminar für Statistik. Zürich: ETH Zentrum CH-8092.

Blom, H. A., Stroeve, S. H., & de Jong, H. H. (2006). *Safety Risk Assessment by Monte Carlo Simulation of Complex Safety Critical Operations.* Amsterdam: National Aerospace Laboratory NLR.

Comisión de Investigación de Accidentes e Incidentes de Aviación Civil. (2013). *Report IN-010/2010.* CIAIAC.

Eurocontrol Experimental Centre. (2011, April). *User Manual for the Base of Aircraft Data (BADA) Revision 3.9.* European Organisation for the Safety of Air Navigation.

European Aviation Safety Agency. (2014). *Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes CS-25.* EASA.

European Commission. (2008). Commission Regulation (EC) No 859/2008. *Official Journal of European Union*.

Everdij, M. (2010). *Compositional modelling using Petri nets with the analysis power of stochastic hybrid processes.* Enschede: University of Twente.

Grandeau, S. C. (1995). *The Processes of Airline Operational Control.* Massachusetts Institute of Technology, Department of Aeronautics and Astronautics.

International Air Transport Association. (2009). *Guidance Material and Best Practices for Fuel and Enviromental Management.* Geneva: IATA.

International Civil Aviation Organization. (2010). *ICAO Standards and Recommended Practises, Annex 6, Part 1, Ninth Edition, Amendment 38.* ICAO.

International Civil Aviation Organization. (2012). *Flight Planning and Fuel Management Manual.* ICAO.

Robinson, D. P., & Murphy, D. J. (2013). *Taxi Time Variation at Major U.S. Airports.* Washington, DC: Federal Aviation Administration.

Rubino, G., & Tuffin, B. (2009). *Rare Event Simulation using Monte Carlo Methods.* Rennes, France: John Wiley & Sons.

Safety, Eurocontrol/FAA AP15. (2014). *Agent-based Dynamic Risk Modelling for ATM: A White Paper.* Eurocontrol.

Stroeve, S. H., & Bakker, G. J. (2007). *Mathematical model of longitudinal separation evolution during arrivals with continuous descent approach segments.* Amsterdam: National Aerospace Laboratory NLR, Memorandum ATSI-2007-110.

Stroeve, S. H., Doorn, B. V., & Everdij, M. (2013). D1.2: The Human Contribution - Analysis of the Human Role in Resilience in ATM. *New design principles fostering safety, agility and resilience for ATM*.

The Boeing Company. (2005). *Flight Planning and Performance Manual 737-800 CFM56-7B26*. Seattle, Washington U.S.A: Flight Operations Engineering Boeing Commercial Airplane Group.

van Es, G. W. (2002). *Assessment of Standard probabilities in Support of FAA AC 25.1309.* Amsterdam: National Aerospace Laboratory, NLR-CR-2002-601.

12 Appendices

12.1 Appendix A: Fuel-related safety incident of Ryanair in May 2010

In May 2010, a fuel-related incident occurred in which the Ryanair aircraft landed with less than FRF. In 2013, a detailed report was written about this incident (Comisión de Investigación de Accidentes e Incidentes de Aviación Civil, 2013). We present this case, because it is an example of one of the events whose probability we assess with the model. In this section, we present main observations and conclusions of the report.

12.1.1 Summary of the event

On 14 May 2010, a Ryanair aircraft Boeing 737-800 was flying from Stansted Airport (London) to Alicante Airport (Spain). It was cleared to land at runway 10, but performed a missed approach due to wind conditions. The crew tried a second approach at runway 28, but the wind conditions were similar so they decided to divert to alternate airport in Valencia. Before Valencia the crew issued an urgency declaration (PAN-PAN) due to low fuel. Eight minutes later, during final approach, they declared an emergency (MAYDAY). They landed at runway 12 in Valencia Airport uneventful. The amount of fuel in aircraft after landing was 956 kg which is 183 kg less than calculated FRF. The aircraft was refuelled and continued to Alicante Airport.

12.1.2 Calculation of the fuel

According to OFP used by the crew these were the amounts calculated based on company`s Operational Manual:

Fuel category	Calculated amount of fuel
Taxi fuel	150 kg
Trip fuel	5 442 kg
Contingency fuel	272 kg
Alternate fuel	1 120 kg
Final reserve fuel	1 139 kg
Additional fuel	0 kg
Extra fuel	0 kg
Total fuel	8 123 kg
Rounded up total fuel	**8 200 kg**

150 kg is the average taxi fuel equivalent to 12 minutes of taxiing for Boeing 737-800 series. The contingency fuel was 272 kg corresponding to 5% of the trip fuel to Alicante. It is equivalent to 7 minutes of flight. The first alternate destination was Valencia Airport, for which the required fuel amount was 1 120 kg. The second alternate destination was San Javier-Murcia Airport for which the

calculated fuel amount was 1 100 kg. This amount was established by company policy as the minimum alternate fuel. So the actual fuel amount required to get to San Javier-Murcia Airport was lower. The crew did not know the actual amount required.

The calculation was in accordance with EU OPS 1.255 (European Commission, 2008). The captain decided not to add any extra fuel based on positive meteorological forecast. Additional fuel (as defined in paragraph 3.2.1.6) was not mentioned in the report. According to company`s operational manual, Ryanair`s fuel policy aims to minimize fuel amount during take-off to minimize the fuel consumption and therefore also fuel costs. It is common for Ryanair aircrafts to land with just FRF. This policy forces also other competitive airlines to adopt this policy. This could lead to situations that several airplanes with not enough fuel reserves approach the same airport at one time.

12.1.3 Main observations and causes

The crew decided to try a second approach in Alicante for runway 28 even though the wind condition was similar as during the first approach due to lack of accurate wind information. When cleared for the second approach, the amount of fuel was already 41 kg below the fuel required to divert to Valencia and arrive with FRF. This was due to long time spent manoeuvring over Alicante Airport. It was 22 minutes between first and second missed approach because of weather conditions and traffic situation. By the time the crew started diverting to Valencia, the fuel dropped to 327 kg below the required minimum. According to EU OPS 1.375 (European Commission, 2008), they should have declared an emergency by that time. The crew declared urgency (PAN PAN) based on Ryanair regulations, which was not understood by Valencia control. When the fuel fell below FRF, the crew declared emergency (MAYDAY). The use of MAYDAY and PAN-PAN phraseology is not considered in EU OPS (European Commission, 2008) or ICAO (International Civil Aviation Organization, 2010).

The first alternate destination (Valencia) is farther from Alicante than the second (Murcia-San Javier) so it would took less fuel and time to reach second alternate destination. Because of the characteristics of Ryanair`s OFP, the crew did not know this.

12.1.4 Conclusion

The incident was caused by the crew`s inadequate decision-making in opting to make a second approach in Alicante, in the choice of alternate destination and in the flight parameters used en route to Valencia. Ryanair`s fuel saving policy complies with the legal requirements, but does not leave enough for contingencies. These resulted in the fuel being below the required final reserve upon landing. Another contributing factor was the wind information provided by ATC in Alicante before the second approach. They did not give the crew a clear picture that would help them making more suitable decisions.

12.2 Appendix B: Other fuel-related safety incidents

12.2.1 Air Transat 2001

On 24 August 2001, an Air Transat Airbus A330-200 eastbound across the North Atlantic at night experienced a double-engine flameout after which Lajes on Terceira Island in the Azores was identified as the best diversion. A successful glide approach and landing were subsequently achieved there.

The Investigation found that the flameouts had been the result of fuel exhaustion after a fuel leak from the right engine caused by a pre-flight maintenance error. Fuel exhaustion was found to have occurred because the flight crew did not perform the Quick Reference Handbook procedure applicable to an in-flight fuel leak.

Source: Database of accidents and incidents on SKYbrary

http://www.skybrary.aero/index.php/A332,_en-route,_North_Atlantic_Ocean,_2001_(HF_LOC_AW_FIRE)

12.2.2 Virgin Atlantic Airways 2005

On 8 February 2005, a Virgin Atlantic Airways A340-600 experienced in-flight fuel management problem during flight from Hong Kong to London. This led to loss of power of No. 1 engine and temporary power loss of No. 4 engine. The captain decided to divert to Amsterdam where the aircraft landed safely on three engines.

The cause of the starvation of the inner fuel tanks was a failure of the discrete outputs of the master Fuel Control and Monitoring Computer which caused that the automatic transfer of fuel within the aircraft stopped functioning.

Source: Database of accidents and incidents on SKYbrary

http://www.skybrary.aero/index.php/A346,_en-route,_Amsterdam_Netherlands,_2005_(LOC_AW)

12.2.3 Tuninter 2005

On 6 August 2005, a Tuninter ATR 72-210 was ditched near Palermo after fuel was unexpectedly exhausted en route. The aircraft broke into three sections on impact and 16 of the 39 occupants died.

The Investigation found that insufficient fuel had been loaded prior to flight because the flight crew relied exclusively upon the fuel quantity gauges which had been fitted incorrectly by maintenance personnel. It was also found that the pilots had not fully followed appropriate procedures after the engine run down and that if they had, it was at least possible that a ditching could have been avoided.

Source: Database of accidents and incidents on SKYbrary

http://www.skybrary.aero/index.php/AT72,_en-route,_Mediterranean_Sea_near_Palermo_Italy,_2005_(AW_LOC_HF)

12.2.4 Delta Airlines 2008

On 26 November 2008, a Boeing 777-200 operated by Delta Air Lines on a scheduled passenger flight from Shanghai to Atlanta was in the cruise at flight level FL390 in the vicinity of Bozeman, Montana when there was an uncommanded thrust reduction or 'rollback' of the right engine. Prescribed drills were followed and a descent to FL310 was made after which normal thrust control was regained and the flight continued to the planned destination. None of the 263 occupants were injured.

The accumulation of ice in the fuel system, which formed from the water normally present in jet fuel, was determined as a probable cause of the incident. Contributing to the incident were certification requirements (with which the aircraft and engine fuel systems were in compliance), which did not account for the possibility of ice accumulating and subsequently releasing in the aircraft and engine fuel feed system.

Source: Database of accidents and incidents on SKYbrary

http://www.skybrary.aero/index.php/B772,_en-route_Bozeman_MT_USA,_2008_(AW_LOC)

12.2.5 Hapag-Lloyd 2000

On 12 July 2000, Airbus A310-304 departed from Chania, Greece to Hannover, Germany. After take-off, the crew noticed that it was not possible to fully retract the landing gear. Using the FMS, the crew estimated the aircraft's fuel consumption and the captain decided to shorten the flight and land in Munich. There was insufficient fuel to reach Munich, however, as the Flight management system was not designed to take into account the extra drag produced by a half-raised landing gear. After an alarm informed the crew that there was only 1.3 metric tons of jet fuel left in the tanks, the pilot quickly redirected the flight to Vienna, 220 km away, although Zagreb was closer at 75 km. The aircraft ran out of fuel 20 kilometres before Vienna and the crew was left to glide the jet towards the runway. The aircraft landed on the grass about 500 m from the runway. The left main gear broke off and the No. 1 engine and wing sustained substantial damage as the aircraft slid for 600 m before coming to rest. Some of the passengers were injured, although none seriously. The aircraft was written off due to the severe damage caused to the underside of the fuselage.

Source: Database of Aviation Safety Network

http://aviation-safety.net/database/record.php?id=20000712-0

12.3 Appendix C: Sources for parameters

In Table 12.1, we list sources for most of the parameters used in the model. The rest of the parameters were determined by consulting the operation experts from NLR. These were mostly parameters that are highly dependent on the specific situation for which the model is being used. Since we simulated only a general setting, the aim was to determine a realistic value for these parameters.

Table 12.1: sources for some of the parameters used in the model

Environment agent (section 6.2)	
α	The dependence of wind on altitude is based on formulas from chapter 6 of (Stroeve & Bakker, 2007).
h_{ref}	
C_{w1}	
C_{w2}	
C_{w3}	
C_{w4}	
μ_w	Parameters are based on empirical data from Schiphol airport in Amsterdam from (Stroeve & Bakker, 2007).
σ_w	
w_{min}, w_{max}	
h_{trop}	These are the parameters of the International Standard Atmosphere model, defined in chapter 3 of (Eurocontrol Experimental Centre, 2011).
τ_0	
β	
p_0	
g	
R	
Airports agent (section 6.3)	
t^i_{tx}, σ^i_{tx}	Taxiing parameters are from (Robinson & Murphy, 2013).
$t^i_{tx,min}, t^i_{tx,max}$	
d^i_{hold}	Parameter is based on an article from Virtual Air Traffic Simulation Network. *http://www.vatsim.net/pilot-resource-centre/ifr-specific-lessons/introduction-holding-patterns*
h^i_{hold}	Parameter is based on an article from SKYbrary. *http://www.skybrary.aero/index.php/AT72,_vicinity_Stockholm_Bromma_Sweden,_2010_(LOS_AGC_HF)*
Airline operation control agent (section 6.4)	
C_{H1}	Parameters are from section 2.2.1 of Flight Planning and Performance Manual for Boeing 737-800 (The Boeing Company, 2005).
C_{H2}	
H_{max}	
h_{fAO}	Parameters are based on regulation from (European Commission, 2008).
t_{fCG}, t_{fFR}	

Aircraft agent (section 6.5)	
C_{tx}	Parameter is equal to the average fuel flow during taxiing based on data from more than 8500 flights of aircraft type Boeing 737-800.
m_0	These parameters are from Base of Aircraft Data, revision 3.9 (Eurocontrol Experimental Centre, 2011), specifically for aircraft type Boeing 737-800.
m_{max}	
C_{f1}	
C_{f2}	
C_{f3}	
C_{f4}	
C_{fcr}	
C_{T1}	
C_{T2}	
C_{T3}	
C_{Tcr}	
C_{Tapp}	
C_{Tld}	
C_{D1}	
C_{D2}	
S	
h_{app}	
h_{ld}	
v_{cl}^{TAS}	
v_{cr}^{TAS}	
v_{de}^{TAS}	
v_{ROC}	
v_{ROD}	
Crew agent (section 6.6)	
p_{MA}	Parameter is from report (van Es, 2002)

12.4 Appendix D: Original list of fuel-related hazards

Table 12.2: list of hazards identified in (International Civil Aviation Organization, 2012)

ID	Description of hazard	Source
H001	Tropical storm, winter storm, tornado, cyclone	page 5-28
H002	Icing, freezing precipitation, snow	page 5-28
H003	Heavy rain	page 5-28
H004	Strong winds	page 5-28
H005	Thunderstorms	page 5-28
H006	Wind shear	page 5-28
H007	Fog	page 5-28
H008	Dust or sand storms	page 5-28
H009	Lightning	page 5-28
H010	Volcanic eruption	page 5-28
H011	Geophysical event on the ground, e.g. earthquake or tsunami	page 5-28
H012	Space weather (e.g. solar activity variations) affecting satellite communication or navigation	page 5-28
H013	ATM congestion	page 5-28
H014	Mechanical failure of an aeroplane system	page 5-28
H015	Adverse terrain or large bodies of water along the route	page 5-28
H016	Isolated aerodrome	page 5-28
H017	Runway closure	page 5-28
H018	Airspace closure	page 5-28
H019	Political unrest or terrorism	page 5-28
H020	Organization changes, e.g. changes to key personnel, rapid growth, rapid contraction, corporate mergers	page 5-28
H021	Operational changes, e.g. new equipment, adapted procedures	page 5-28
H022	Hazards affecting ATC capabilities	page 5-29
H023	Hazards affecting aerodromes	page 5-29
H024	Hazards affecting field condition reporting	page 5-29
H025	Hazards affecting meteorological reporting or forecasting	page 5-29
H026	Hazards affecting airline operational control, flight following and flight monitoring	page 5-29
H027	Longer taxi time than planned	page 5-26
H028	Taxi and ground delay	page 5-27
H029	En-route speed restriction	page 5-27
H030	En-route deviation	page 5-27
H031	Air traffic delay	page 5-27

H032	ATC flow management and aerodrome congestion	page 5-27
H033	Long time spent in holding	page 5-27
H034	Missed approaches	page 5-27
H035	Additional approaches	page 5-27
H036	Insufficient aircraft type specific fuel planning experience of flight crew	page 5-34
H037	Flight crew unfamiliar with route	page 5-34
H038	Route near maximum range of aeroplane	page 5-34

Table 12.3 list of hazards identified in the hazard database of (Stroeve, Doorn, & Everdij, 2013)

ID	Description of hazard	Source
H039	Lack of routing accuracy of flight management system	page 60
H041	Aircraft not equipped with technical system, e.g. auto-landing system	page 60
H042	Error in routing of flight management system, e.g. wrong waypoints in database, or outdated FMS plan	page 60
H043	Airborne systems not working, e.g. cockpit display, flight management system, or large electronic failure	page 61
H044	Problem with instrument landing system	page 61
H045	Problem with landing gear	page 61
H046	Degradation of aircraft structure	page 61
H047	Problem with the positioning system, e.g. failure of GPS, navigation error in own position	page 61
H048	Degradation of one or multiple engines	page 61
H049	Problem with approach or runway lights	page 62
H050	Bird strike	page 62
H051	No ATC on an airport	page 62
H052	Runway blocked or contaminated	page 62
H053	Restricted airspace	page 62
H054	Complex standard arrival route	page 62
H055	Controller does not inform other controllers about an emergency situation	page 62
H056	Poor coordination between civil and military ATC	page 62
H057	Poor coordination between ATC centres	page 63
H058	Misidentification of an aircraft by ATC	page 63
H059	ATIS does not provide correct information to pilots	page 63
H060	Flight plans of ATC system and FMS differ	page 64
H061	Malfunctioning of ATC systems, e.g. radar	page 64
H062	Controller makes a wrong decision	page 66
H063	Controller makes a mistake in aircraft identity	page 66

Table 12.4: list of hazards identified in (European Commission, 2008)

ID	Description of hazard	Source
H130	Jet stream	page 254/213
H131	Mountain waves	page 254/213
H132	Significant temperature inversions	page 254/213
H133	Bird hazards and strikes	page 254/49

Table 12.5: list of additional hazards from various sources, mostly consultations with experts from NLR

ID	Description of hazard	Source
H134	Pilots feel pressed by management to reduce fuel intake	NLR
H135	Pilots plan a nearby alternate destination, which is in practice not a feasible option (e.g. for political reasons)	NLR
H136	Fuel quantity indicator is malfunctioning	NLR
H137	Pilots do not check fuel quantity	NLR
H138	Failure in the fuel system such that part of the fuel cannot be used	NLR
H139	Fuel leakage	NLR
H140	Crew does not follow the applicable procedures correctly	NLR
H141	Pre-flight maintenance error	NLR
H142	Fuel management not working properly, e.g. automatic transfer of fuel	NLR
H143	Inadequate certification requirements	NLR
H144	Fuel imbalance	SKYbrary
H145	Fuel freezing	SKYbrary
H146	Electrical failure	SKYbrary
H147	inability to fully retract flaps after missed approach	NLR
H148	plane is flying in lower altitude than expected	NLR
H149	pilot is not flying in optimal mode	NLR
H150	Malfunctioning of AOC systems	NLR
H151	Incorrect fuel bias	NLR