Diss. ETH No. 19043

LHC Machine Protection System: Method for Balancing Machine Safety and Beam Availability

A dissertation submitted to
ETH ZURICH

for the degree of
Doctor of Sciences

presented by
SIGRID CLAUDIA WAGNER
Dipl. Masch.-Ing., ETH Zurich
born 23 April 1978
citizen of Untersiggenthal

accepted on the recommendation of
Prof. Dr. Wolfgang Kröger, examiner
Dr. Rüdiger Schmidt, co-examiner

2010

für Bopi

Acknowledgements

I gratefully acknowledge the commitment and support of those who made this thesis possible.

First, I would like to thank my supervisor, Wolfgang Kröger, for providing the opportunity for this thesis and the working environment at LSA. I owe many thanks also to my supervisors at CERN, Rüdiger Schmidt and Jörg Wenninger, for their confidence in and enthusiasm for my work. I thank Ralf Mock, who initially opened the door to the world of particle accelerators to me, and Irene Eusgeld, who introduced me to the art of research.

I am grateful for the continuous support I received from my colleagues at CERN, in particular Bernd Dehning, Benjamin Todd, Markus Zerlauth and Jan Uythoven, who never got tired of sharing their system knowledge with me. Likewise, I owe thanks to Gianluca Guaglio for his introduction to his work and genuine Italian kitchen.

Jan Hauschild and Marcel Held I thank for the inspiring discussions we had. Special thanks go to Daniel Alai for the Polysnack sessions, where the foundation of the analytical model description was laid. The implementation of the latter I owe to Roberto Nibali, with whom I sailed through some legendary programming sessions.

I thank my colleagues at the lab, in particular Markus Schläpfer and Manuel Kaegi, who looked after me when I first arrived, supported me with their knowledge and experience and with whom I shared many great moments. Also, I am very grateful to Patrick Probst for the IT support and to Monika Mortimer for cross-reading my work.

Finally, I would like to thank my friends and my family for being a source of inspiration and supporting me in the realisation of ideas and dreams. Special thanks go to my parents and brothers for being my ultimate backup at all times. Thank you.

Zurich, April 2010 Sigrid Wagner

Abstract

The Large Hadron Collider (LHC) at CERN in Geneva, Switzerland, exceeds existing particle accelerators in terms of size and complexity. The most remarkable machine damage potential is held by the amount of stored energy.

This thesis introduces a quantitative method for the reliability analysis of the LHC Machine Protection System (MPS) in terms of machine safety and beam availability. It is based on object-oriented modelling of the primary signal path, where the components' behaviour is described by a simple Markov model with two failure states. The explicit inclusion of machine failure allows for the quantification of five scenarios. They include the safety-relevant scenario of a missed emergency shutdown and the scenario of a preventive shutdown, which is crucial with regard to beam availability.

The presented MPS model covers two of the main MPS subsystems, namely the Beam Loss Monitor System and the Beam Interlock System. The model includes almost 5000 individually modelled components. It is implemented twice, first based on Monte Carlo simulation and second based on an analytical model description.

The implementation of the analytical description is used for a series of case studies. The case studies yield scenario probabilities that confirm expectations based on operational experience and previous studies. The results are verified by the results of related Monte Carlo simulations, which show good accordance. Component importances are also extracted.

A series of sensitivity analysis studies based on input parameter variation are presented. The studies do not reveal any unexpected dependence and as such confirm the stability of the model implementation. In addition, a set of model extensions are introduced which overcome some of the limiting assumptions and simplifications of the initial model.

Zusammenfassung

Der LHC (Large Hadron Collider) am CERN in Genf, Schweiz, ist grösser und komplexer als alle bisherigen Teilchenbeschleuniger. Das grösste Gefahrenpotential für den LHC birgt die in der Maschine gespeicherte Energie.

Die vorliegende Dissertation stellt eine quantitative Methode vor für die Zuverlässigkeitsanalyse des Maschinenschutzsystems (MPS) des LHC in Bezug auf Maschinensicherheit und Strahlverfügbarkeit. Die Methode basiert auf der objekt-orientierten Modellierung des primären Signalpfads. Die Komponenten werden durch ein einfaches Markoff-Modell mit zwei Fehlerzuständen beschrieben.

Der Einbezug von Maschinenstörungen ermöglicht die Quantifizierung von fünf Szenarien. Dazu gehören das sicherheitsrelevante Szenario einer verpassten Notabschaltung und die vorsorgliche Abschaltung, welche bezüglich Strahlverfügbarkeit entscheidend ist.

Das vorliegende MPS-Modell umfasst zwei der Hauptsubsysteme des MPS, das *Beam Loss Monitor System* und das *Beam Interlock System*. Das Modell beinhaltet annähernd 5000 einzeln modellierte Komponenten. Es ist zweifach implementiert, einmal basierend auf einer Monte-Carlo-Simulation und einmal basierend auf einem analytischen Modellbeschrieb.

Die analytische Implementierung wird für eine Reihe von Fallbeispielen verwendet. Sie ergeben Szenario-Wahrscheinlichkeiten, die Betriebserfahrungen und vorgängige Analysen bestätigen. Die Ergebnisse der Fallbeispiele werden durch Monte-Carlo-Simulationen verifiziert. Sie zeigen gute Übereinstimmung. Die Importanzen der Komponenten werden zusätzlich bestimmt.

Eine Reihe von Sensitivitätsanalysen basierend auf Parametervariation wird ebenfalls vorgestellt. Sie zeigen kein unerwartetes Verhalten und bestätigen somit die Stabilität der Implementierung. Zusätzlich werden einige Modellerweiterungen präsentiert, welche gewisse Einschränkungen des Anfangsmodells beheben.

Contents

Chapter 1

Introduction

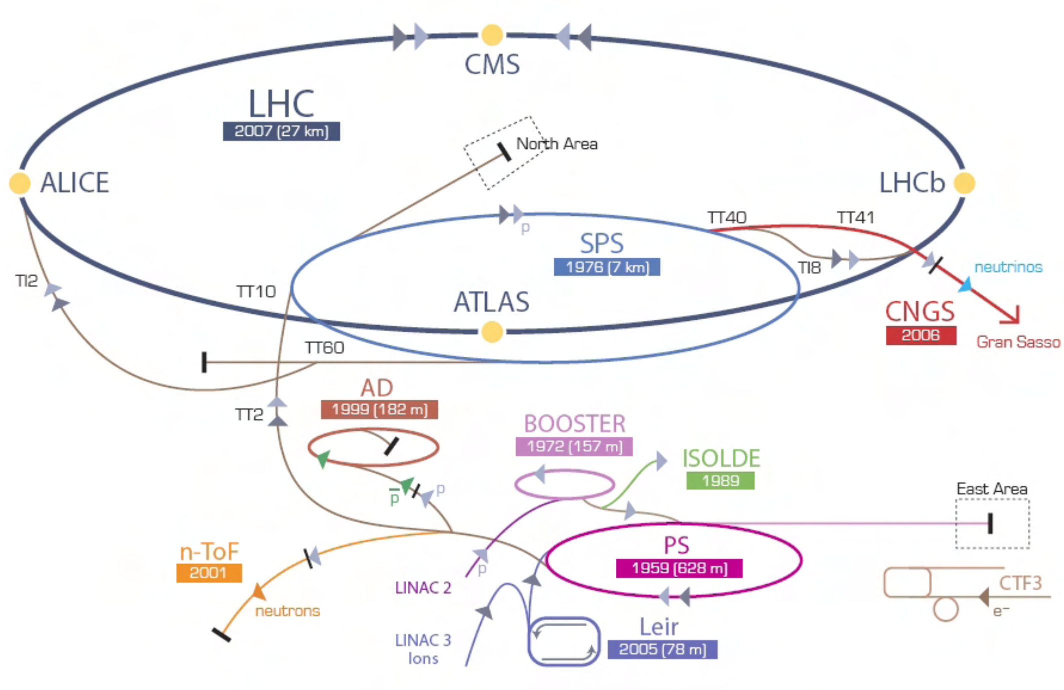


Figure 1.1: *CERN accelerator complex [1]*

The European Organization for Nuclear Research (CERN) in Geneva, Switzerland, operates a succession of particle accelerators of different types (Fig. 1.1). The accelerators provide particle beams in conditions specific to fundamental research experiments in the field of particle physics.

The latest accelerator designed and built at CERN and the organisation's flagship is the Large Hadron Collider (LHC). *First LHC beam* was celebrated on 10 September 2008.

After an incident caused by a fault in an electrical connection [2], the LHC was shut down for repair and system optimisation. It was successfully recommissioned in late 2009, and a new world record was set in terms of beam energy [3].

Once fully operational, the LHC is expected to deliver entirely new insight into the laws of nature [4]. The most popular ambition is the discovery of the Higgs boson. Its detection would confirm the standard model of particle physics [5].

This thesis deals with the reliability of the LHC Machine Protection System (MPS). The MPS monitors the accelerator and triggers an emergency shutdown in case of dangerous conditions. Its reliability is crucial for a successful operating of the LHC.

1.1 Motivation

The concern of reliability in the field of particle accelerators manifests itself in conferences and workshops. Some key aspects are illustrated on the basis of the second Accelerator Reliability Workshop (ARW), which took place from 26 to 30 January 2009 in Vancouver, Canada [6]. The workshop attendees represented more than 30 institutes in 15 countries mainly covering North America, Europe and Asia.

The study of the contributed papers reveals the following outlines:

- The figure of merit is high accelerator availability (uptime), which is achieved by maximising the system reliability and minimising the downtime periods [7–11].
- In modern accelerators, due to their size and complexity, high system availability is difficult to attain, requiring highly reliable subsystems [8].
- Manifold effort is taken to improve the availability of existing accelerators in order to obtain more efficient operation and higher productivity [8, 12].

The taken measures mainly base upon the direct analysis of operational data and the related identification of weak points such as unreliable components or time-consuming procedures [9, 10, 12–15].

If operational and testing experience is unavailable or if more complex aspects and interrelations have to be studied, more formal reliability analysis methods and their models are used. Walter and Schneeweiss [16] distinguish five main sets of models:

- Failure mode, effect and criticality analysis (FMECA) models

- Boolean models (using fault trees (FTs) or reliability block diagrams (RBDs))
- Markov models (using state space with exponentially distributed times to failure)
- Petri-net-based models
- Statistical simulation (Monte Carlo) models

A well-established quantitative methodology in the nuclear power industry is the Probabilistic Safety Assessment (PSA) [17, 18], which includes FTs. Podofilli et al. [18] discuss the application of PSA to an accelerator-driven experimental facility with the goal of evaluating the adequacy of its safety systems. The extensive use of programmable electronic and prototype components in the systems is pointed out to entail notable challenges, particularly with regard to the determination of event frequencies and failure probabilities, which are required as model inputs. Other assessments of accelerator reliability base upon RBDs, as for example used in the study by Burgazzi and Pierini [19].

The considerations regarding programmable electronic and prototype components applies to the LHC MPS likewise. Additionally, the MPS exceeds the protection systems of existing particle accelerators in terms of size and functional and operational requirements. Nevertheless, a series of studies have been performed to address the reliability of the MPS. Each focusses on a specific MPS subsystem, using different model approaches. FMECA models were used by Todd [1], FTs by Guaglio [20] and Markov models by Filippini [21]. Vergara [22] developed an analytical model and performed analyses on the MPS subsystem for quench protection.

The presented studies focus on subsystems and specific issues of those. The performance and reliability of the MPS however depends on all of its subsystems. In continuation of the previous work, the objective of this thesis therefore is targeted on a more global reliability analysis of the MPS.

1.2 Objectives

The core objective of the thesis is the development of an integrated quantitative method suitable for a global reliability analysis of the MPS. The method should allow for addressing one of the main issues related to the MPS and its operation, namely the trade-off between machine safety and beam availability.

This trade-off is defined by the missing of an emergency shutdown in case of dangerous accelerator conditions and by the triggering of preventive shutdowns in case of detected failures within the MPS. The first scenario is referred to as *missed emergency beam dump*. It interferes with machine safety due to the entailed damage of the LHC. The latter is called *false beam dump*. It interferes with beam availability due to the inclusion of possibly unnecessary shutdowns.

Based on the general objective stated above, the requirements regarding the method have been specified. The method needs to yield the following quantitative results:

- Probability of missed emergency beam dump (interfering with machine safety)
- Probability of false beam dump (interfering with beam availability)
- Importance of the system components with regard to the above scenarios

These system figures depend on the MPS layout and the reliability of the different components and subsystems respectively. Accordingly, the method and the related modelling approach need to take into account the following system characteristics:

Size The MPS includes thousands of components (even on a very high system abstraction level), whose reliabilities have to be included in a model

Redundancy The MPS includes a wide range of redundancy of different types and on different levels

System demand Scenario *missed emergency beam dump* involves the demand of the MPS (defined by the reliability of the accelerator)

Fail-safe measures Scenario *false beam dump* is based on fail-safe measures included in the MPS design

The development of the method is targeted on the inclusion of the mentioned characteristics. It is to be noted that a range of other aspects which are also important with regard to MPS reliability and operation are not explicitly included. Such aspects are for example

- human failures and safety culture,
- personal safety or

- financial risk.

It is also to be noted that the method is geared on a reliability analysis based on the design function of the MPS, i.e. the triggering and performance of a shutdown upon detection of a dangerous condition. This particularly excludes the question whether all of the possible dangerous conditions in the accelerator are covered by the MPS and its detectors. Furthermore, the reliability analysis focuses on operation with circulating beams. Beam injection and extraction are not covered in detail.

In contrast to the established methods used in the previous studies on the MPS, the approach chosen for this work is based on object-oriented modelling, i.e. the component-by-component representation of the system. It includes simple Markov models as the basic modules and Monte Carlo technique for implementation.

1.3 Outline

This thesis is structured as follows:

Chapter 2 provides an overview on the LHC and the essential aspects with regard to LHC machine protection. It also introduces the terminology frame of this work.

Chapter 3 describes the LHC MPS, i.e. the system to be analysed.

Chapter 4 presents the developed modelling approach for the analysis in general and the related model of the MPS in particular.

Chapter 5 describes the implementation of the MPS model based on Monte Carlo simulation, the first out of two implementations presented in this work.

Chapter 6 describes the second implementation of the MPS model, based on an analytical description.

Chapter 7 presents the results of case studies performed on the MPS model, using the implementation of the analytical description.

Chapter 8 provides the verification of the case study results, as well as a series of sensitivity analyses.

Chapter 9 introduces model extensions which allow to include some additional features to the analysis.

The work is completed by conclusions and outlook.

Chapter 2

Overview on the LHC and its Machine Protection

This chapter introduces the LHC and its machine protection. It provides a brief overview on structure, function and operation and includes the most relevant terminology underlying the reliability analysis presented in this thesis.

2.1 The LHC

The LHC is located in a ring tunnel at a depth of 45 to 170 m underground [23]. Its main characteristics are implied in the name:

Large With a circumference of 26.7 km, the LHC is the world's largest particle accelerator [4]. Many of its subsystems set new standards in terms of size, complexity and performance compared to existing facilities.

Hadron This is a group of subatomic particles that are built of quarks and antiquarks [5]. Protons and neutrons are hadrons, beside others.

Collider This is a type of accelerator that makes particles crash into one another [24].

The LHC is designed to collide beams of either protons or lead ions [4]. Proton beams are used in the current starting phase of operation. Lead ions (i.e. lead nuclei) will be employed in a later stage.

Structure, function and operation are further discussed on the basis of the schematic LHC layout presented in Figure 2.1.

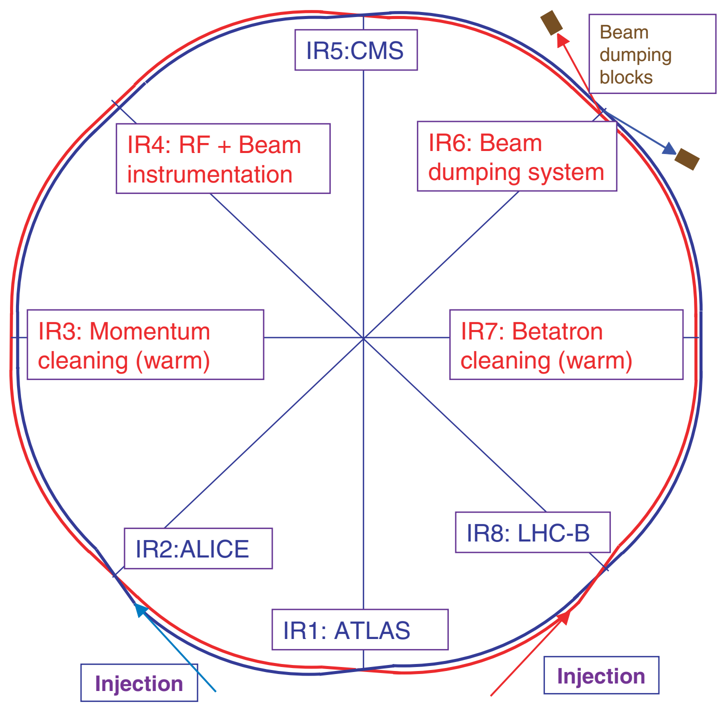


Figure 2.1: *Schematic layout of the LHC [25]. IR: Insertion region.*

2.1.1 Structure

The LHC ring is divided into eight sectors (Fig. 2.1) by eight *interaction points* (IPs, [26, 27]). Seen from the centre of the ring, a position in the ring can be referred to as *left* or *right* of an IP [27]. Two beams are circulated in the ring in opposite directions in two separate vacuum chambers. Beam 1 runs clockwise (blue), beam 2 counter-clockwise (red). The beams' design orbit is not completely circular, but includes arcs and straight sections. The eight straight sections are named *insertion regions* (IRs). They span the area to the left and right of the IPs and thus extend over the boundary of two adjacent sectors, while the eight arcs are located within the sectors.

The LHC is basically built of consecutive magnets of different types and function. The magnets in the arcs are aligned in a succession of so-called FODO cells [28], which are composed of six dipole magnets (MB) and two quadrupole magnets (MQ) (Fig. 2.2). The dipole magnets bend the beams, while the quadrupole magnets focus them. The type of the magnets installed in the IRs depends on the specific requirements of the related IP regarding beam characteristics.

In four of the IPs, the beams are made cross each other. Around these collision points, four large detectors are installed, namely ATLAS (IR1, Fig. 2.1), ALICE (IR2), CMS

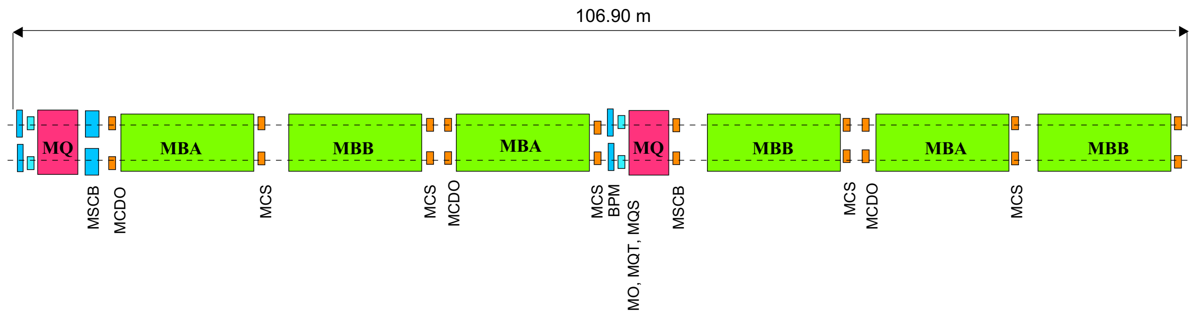


Figure 2.2: *Schematic layout of a LHC arc FODO cell [26]*

(IR5) and LHC-b (IR8). These detectors, also referred to as *experiments*, collect data on the collision events and the resulting new particles.

IR2 and IR8 additionaly contain the magnets required for the injection of the beams (*kicker* and *septum* magnets). IR4 includes the devices to accelerate the particles of the beams (so-called *RF cavities*). The same insertion also features beam instrumentation, i.e. devices measuring the beam parameters. The magnets for the extraction of the beams from the ring are located in IR6.

IR3 and IR7 are called *cleaning insertions* [27]. Cleaning refers to the absorption (i.e. removal) of beam particles whose parameters deviate from given specifications. The deviations potentially lead to the particles being lost in parts of the accelerator where they could cause damage. For reason of precaution, such particles are absorbed in the cleaning insertions by devices called *collimators*. In IR3, particles with high momentum (longitudinal oscillation) offset are removed. In IR7, particles with deviation in the transverse oscillation amplitude of their trajectory are absorbed (betatron cleaning).

2.1.2 Function

The primary function of the LHC is the colliding of the beams (under specified conditions) and the collection of data on the collision events and the emerging particles. The data will allow for new insight on the subatomic structure and the states of matter, as well as on underlying forces and energies.

One of the main goals is the discovery of the Higgs boson. It would contribute to answering the fundamental question why particles have different mass. As a possible explanation, the existence of a Higgs field has been postulated. The theory presumes that the particle mass depends on the interaction of the particle with the Higgs fields. According to the theory, the particle mass is the higher, the stronger its interaction with the Higgs field is. The discovery of the Higgs boson would be a strong indication of the existence of such a Higgs field.

Besides the discovery of the Higgs boson, the detectors at the collision points are designed for different other purposes. While ATLAS [29] and CMS [30] serve general purposes, ALICE [31] and LHC-b [32] are more specific detectors. Table 2.1 provides an overview.

Table 2.1: *The LHC detectors*

Detector	Purpose
ATLAS	Detection of any new particles emerging from the collisions, including Higgs boson [26].
ALICE	Study of the quark-gluon plasma generated in the collisions of lead-ions [26].
CMS	Detection of any new particles emerging from the collisions, including Higgs boson.
LHC-b	Detection of forward particles and recording of their decay (e.g. of 'B mesons'), in order to study the difference between matter and anti-matter [26, 32].

In addition to the large detectors mentioned above, there are two smaller detectors designed to study forward particles generated in the LHC (as does LHC-b). The TOTEM detector, among other things, ultimately measures the size of the proton [33]. It is placed near IP 5, in the insertion of the CMS detector. The LHC-f detector is designed to simulate cosmic rays in laboratory conditions, using forward particles as a source [34]. It is located close to ATLAS (IR1).

2.1.3 Operation

As pointed out above, the LHC is designed to study particles and related effects that have never been observed before. The LHC can be understood as a microscope with an unprecedented resolution, allowing to study matter on a very small scale. The observation of small-scale structures requires particles with small quantum-mechanical wavelength λ [26]. However, according to De Broglie's relation $\lambda = h/p$, the smaller the wavelenght λ, the higher the particle momentum p (h is the Planck constant). Therefore, the smaller the scale to observe, the higher a particle momentum (and energy) is required.

The LHC is designed to achieve a top particle energy of 7 TeV, resulting in a center-of-mass energy of 14 TeV for the proton-proton collisions. Particle energy is also referred to as beam energy. Besides the center-of-mass energy, the luminosity is the second parameter quantifying the performance of an accelerator [26]. The luminosity is defined by the interaction rate of the particles in the IPs divided by the interaction cross section [28]. The LHC is designed to reach a luminosity of 1E34 cm-2 s-1 [26, 27].

The (relativistic) energy of a particle is defined as the sum of its kinetic and its rest energy, the latter of which is a constant [5]. Accordingly, in order to achieve high particle energy, high particle velocity is required. The particles in the LHC are accelerated to almost the speed of light by the electric fields in the RF cavities (see above).

In order to keep the particles on their orbit, i.e. to bend and focus them, very high magnetic fields are required. These are achieved by superconductive (sc) magnets. At top beam energy of 7 TeV, the magnetic flux density (magnetic field) in the dipole magnets of the FODO cells in the arcs (see above) reaches 8.33 T. The sc magnets are operated at a temperature close to absolute zero. Their coils are cooled by superfluid helium at 1.9 K, provided by an extensive cryogenics system.

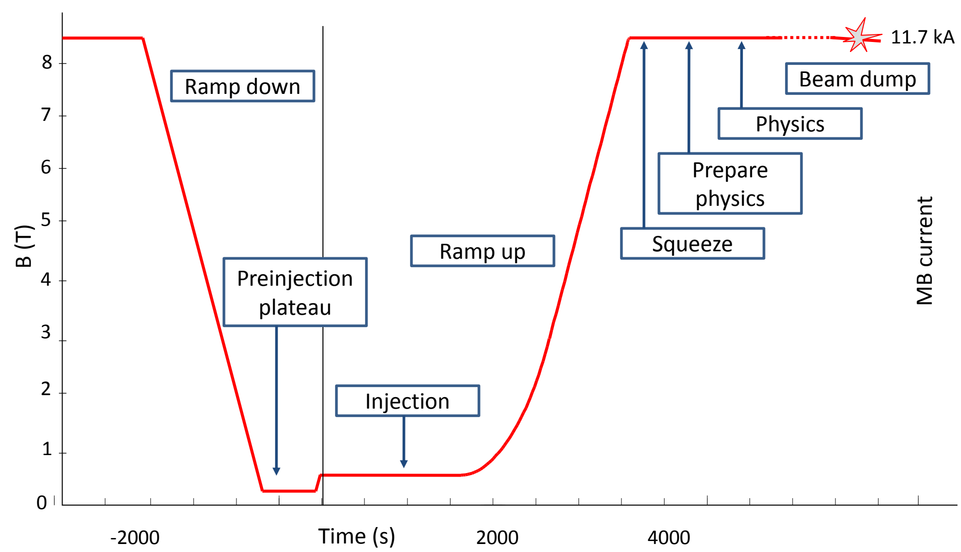


Figure 2.3: *LHC operation cycle [25], on the basis of magnetic field (B) and dipole magnet current*

Figure 2.3 illustrates the nominal LHC operation cycle on the basis of the current and magnetic field in the sc dipole magnets in the arcs. It includes five major phases.

Injection Bunches of 1.15E11 particles at 450 GeV (pre-accelerated in the CERN accelerator complex, Fig. 1.1) are injected from the Super Proton Synchroton (SPS) into the LHC. Twelve batches of either 216 or 288 bunches (2808 bunches in total) compose each of the two beams. The beam energy of 450 GeV corresponds to a magnetic field of about 0.54 T [25]. The beams are kept separated in the IPs, i.e. they are not collided. The injection phase lasts about 15 minutes.

Ramping up The two circulating beams at 450 GeV are accelerated until reaching the top energy of 7 TeV. This involves the ramping of the dipole magnets up to a magnetic field of 8.33 T, which is achieved by a magnet current of 11.7 kA. The ramping phase lasts about 20 minutes.

Squeezing The transverse size of the beams in the IPs is reduced (the beams are 'squeezed') and the beams are made collide. The squeezing lasts 5 to 10 minutes.

Physics The detectors, after short preparation, record data on the collisions. This lasts for several hours [26]. The phase is terminated by a beam dump (*end-of-mission beam dump*) triggered by operators in the control room, i.e. the beams are extracted from the ring in IR6 and lead into absorbers.

Ramping down The dipole magnets are ramped down to a magnetic field slightly below the level required for injection ('preinjection plateau'). They are ramped up again to injection level shortly before the start of the subsequent injection phase [26]. The ramping phase lasts about 20 minutes.

At nominal operation, two cycles are expected per day, relating to physics phases of about 11 hours. A total of 200 days of operation are assumed per year, relating to 400 (nominal) cycles. A cycle is also referred to as 'fill'.

2.2 The LHC Machine Protection

The material costs for the construction of the LHC and the detectors amounts to four billion Swiss francs [35]. In order to allow for an appropriate return of the investment, the LHC has to be protected against damage leading to an interruption of operation or, in the worst case, an early termination of the LHC project.

The imminent danger for the LHC emanates from the energy stored in its electric circuits and beams (Tab. 2.2). The more than 10 000 magnets in the LHC are powered in 1612 electrical circuits [25]. The circuits with the most stored energy are those of the main dipole magnets in the arcs. A main dipole circuit, including one power converter, powers the 154 dipole magnets of an arc in series. Each of the dipole magnetic fields corresponds to an energy of about 7 MJ [26], resulting in 1.1 GJ for the entire circuit. There are eight main dipole circuits in the LHC. Including all the remaining circuits, this results in a total of 10 GJ stored in the LHC magnet system. This energy is enough to heat up and melt about 15 tons of copper [25].

Table 2.2: *Energy stored in LHC magnets and beams at 7 TeV [25]*

Energy stored in one main dipole circuit	1.1 GJ
Energy stored in the whole magnet system	10 GJ
Energy stored in one beam	362 MJ

The energy stored in a beam results from the particle energy of 7 TeV and the number of particles in the beam. The relevant figures are given in Table 2.3.

Table 2.3: *Specifications of a nominal LHC proton beam [25]*

# Protons per bunch	1.15×10^{11}
# Bunches per batch from SPS	216 or 288
# Batches per fill	12
# Bunches per beam	2808
# Protons per beam	3×10^{14}

The proton energy of 7 TeV exceeds the highest proton energy ever achieved (in TEVATRON at Fermilab, USA) by factor 7 [26]. Due to the high number of protons (beam intensity), the resulting energy stored in the LHC beam exceeds TEVATRON by factor 200 (Fig. 2.4). The high intensity and small transverse dimension of the beam result in a beam energy density in the LHC that outreaches other accelerators (e.g. TEVATRON, SPS at CERN or HERA at Desy, Germany) by factor 1000. The energy density is the most relevant parameter for beam-induced damage [26].

The release of even a small fraction of the energy stored in the LHC electric circuits and beams into the LHC equipment may lead to damage. The energy of the beam is released through impact on material. In case of nominal operation, the energy of the beam is safely deposited in the absorbers (dump blocks), which are the only elements of the LHC able to absorb the energy without being damaged. The impact of protons on LHC equipment other than the absorbers is referred to as *beam loss*, i.e. particles leaving their nominal orbit and being 'lost' and absorbed in the surrounding equipment.

One of the most relevant mechanisms regarding the release of the energy of an electrical circuit is the quenching of a magnet (*magnet quench*). It relates to sc magnets and as such to the circuits with the most stored energy, i.e. the main dipole circuits.

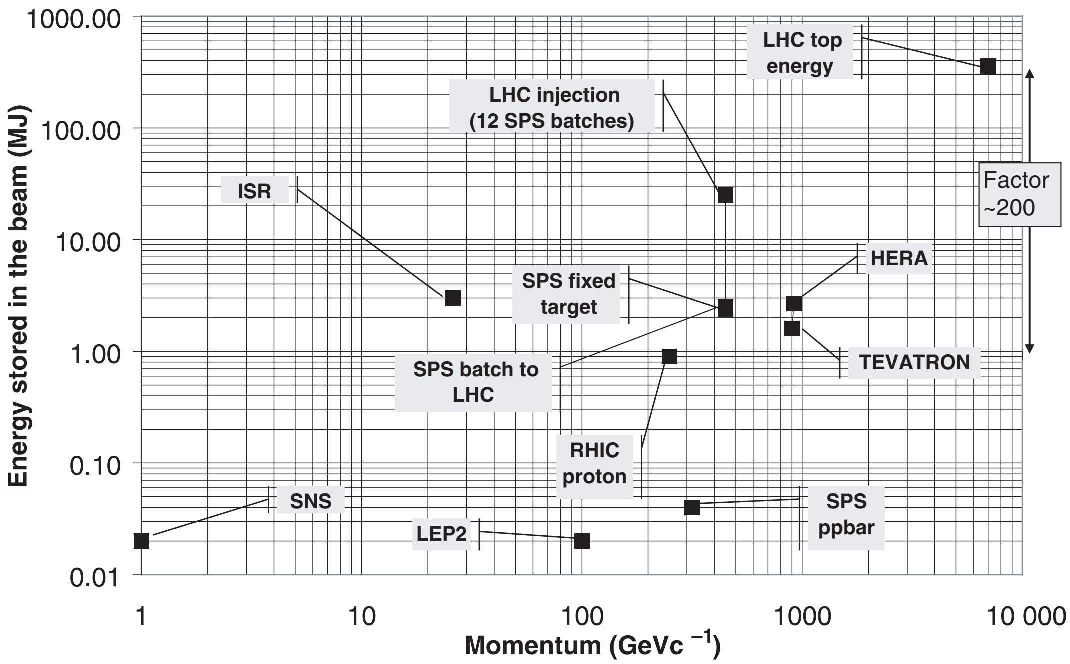


Figure 2.4: *Energy stored in the beams of different accelerators [25]*

A magnet quench refers to the transition from the superconducting to the normal conducting state of a sc cable. It entails the (local) release of the stored energy (due to the resulting rise of resistance).

In order to prevent damage of equipment, the basic tasks of machine protection with regard to safe LHC operation therefore are

- preventing (accidental) beam losses and
- preventing magnet quenches.

These tasks are performed by the LHC Machine Protection System (MPS). In case of beam or machine equipment conditions that could potentially lead to dangerous beam loss or a magnet quench (and related damage), an *emergency beam dump* has to be performed. The beams have to be extracted and safely deposited in the absorbers before damage is caused. In case of a magnet quench still happing (which from experience cannot be completely avoided), the MPS has to control the related energy release such that no damage occurs.

Accordingly, the goal in case of potentially dangerous conditions can be summarized as follows:

- Extract the beams (and their energy) from the machine
- Extract the energy from electric circuits (if necessary)
- Perform these tasks fast enough to prevent damage

The following sections further address the beam loss and magnet quench mechanisms. As will be shown, they are not completely independent but in a mutual causal relationship. The MPS and its subsystems are introduced in detail in Chapter 3.

2.2.1 Beam Loss

The trajectory of a particle in a circular accelerator is stable only if its transverse amplitude (betatron amplitude) stays within a certain limit [26]. The related amplitude limit is called *dynamic aperture* [27, 36]. A particle with an amplitude exceeding the dynamic aperture is eventually lost, i.e. its trajectory gets instable and eventually impacts on surrounding equipment where it gets absorbed. Besides losses due to instable particle trajectories, there are losses due to scattering with other particles, for example residual gas molecules in the vacuum chamber, particles of the same beam or particles of the other beam in the collision points [26, 36].

The mentioned phenomena cannot be avoided completely. Even during nominal LHC operation, there is a constant loss of particles. These losses are referred to as *steady losses*. Additionally to these inherent beam losses, failures during operation can cause scattering or instabilities that lead to more immediate losses of a part or all of the beam [36]. They are referred to as *accidental losses* [26].

Particle impact on solid material involves nuclear and electromagnetic interactions that result in particle cascades (secondary particle shower) [25]. The local energy deposition and related temperature increase (and the resulting damage) depend on the number, energy and spatial distribution of the impacting particles and on the material properties (e.g. heat capacity).

For a given number of particles at a given energy, the damage potential depends on the time period over which their loss occurs (besides the distribution). The shorter the time period, the higher is the impact power. Accordingly, beam losses are categorised based on a parameter called *beam lifetime* (τ). It defines the time after which the number of particles in the beam (i.e. beam intensity) is reduced to a fraction of 1/e of the initial number [26].

Table 2.4 shows the categorisation based on beam lifetime. Steady beam losses are kept at an acceptable (i.e. safe) level by the collimators in the cleaning insertions IP3 and IP7, which continuously absorb particles with potentially instable trajectories. Normal steady losses refer to conditions expected during early LHC operation [26]. High steady losses relate to particular short-time conditions due to operational adaptations, like change of optics, tuning, setting of collimator aperture or others. Low steady losses refer to optimised operation after some years of experience and a possible upgrade of the collimation system.

Table 2.4: *Categorisation of beam losses (adapted from [26])*

	Category	Beam Lifetime
Steady beam loss	low	$\approx$ 100 h
	normal	$\approx$ 10 h
	high	$\approx$ 12 min
Accidental beam loss	slow	$>$ 1 s
	fast	$<$ 1s
	very fast	$>$ 1ms
	ultra fast	$<$ 1 ms, in particular within 1 turn (89μs)

Accidental beam losses require an immediate emergency beam dump to prevent direct damage or magnet quenches. The available time for the MPS to detect the loss and perform a dump obviously depends on the loss category. The categories up to very fast losses are covered by the Beam Loss Monitor System, a subsystem of the MPS with a sufficiently low response time (Chapter 3). Ultra fast losses however, which are caused by failures during injection, extraction or by erroneous triggering of dedicated kicker magnets, go beyond the capabilities (i.e. time frame) of active protection (through controlled extraction of the beams). They depend on passive protection provided by the collimators and are likely to entail damage to the collimators or surrounding equipment.

2.2.1.1 Accidental Beam Loss

Figure 2.5 illustrates the causal chains of failures leading to accidental beam loss. Failures during LHC operation can lead to extended beam loss due to beam instabilities (deflection or defocusing) or scattering. Beam losses also result from a reduction of the physical aperture or from a debunched beam [26].

Beam deflection or defocusing Due to a change in a magnetic field (e.g. caused by

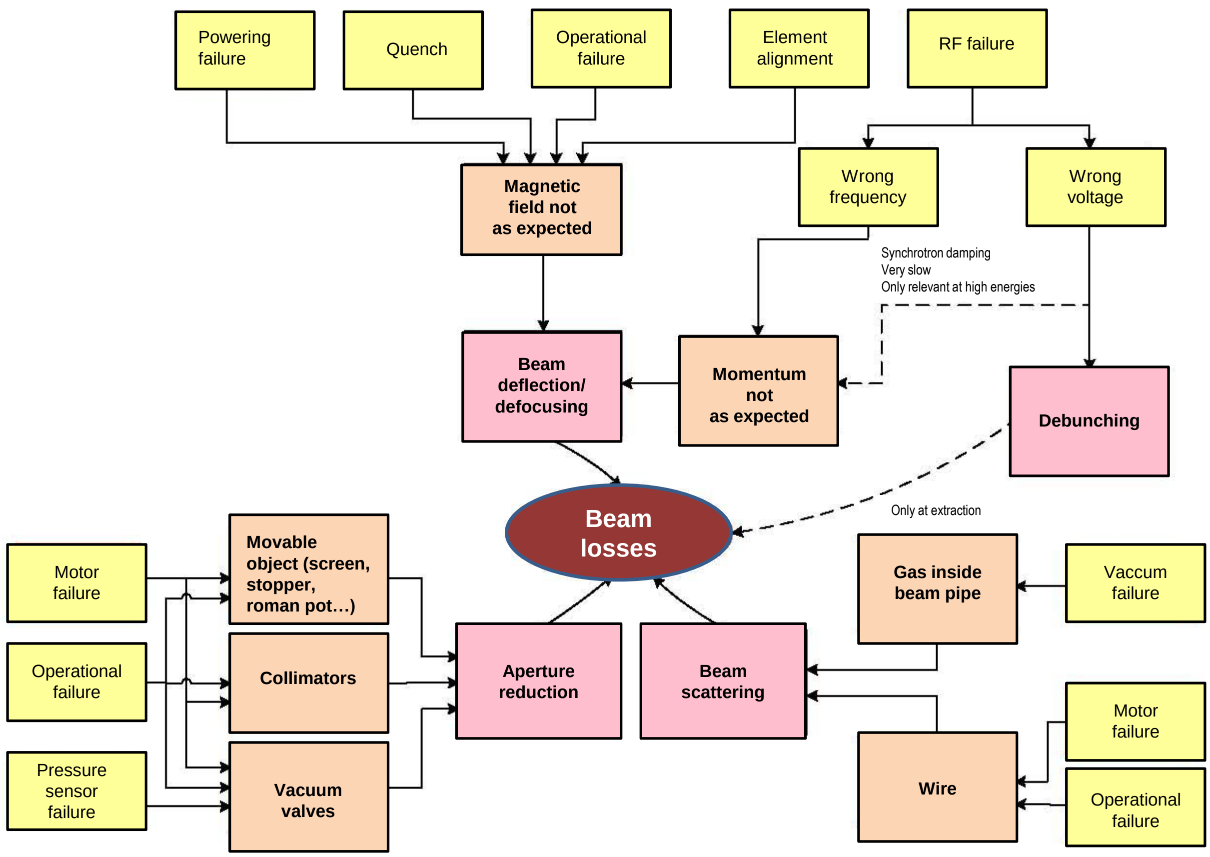


Figure 2.5: *Overview on the causal chains resulting in accidental beam loss [26]*

a magnet quench) or in the particle momentum (e.g. caused by a failure in the RF system).

Beam scattering Due to increased pressure in the vacuum chambers (e.g. caused by a leak) or due to wrongly positioned movable devices such as wire scanners.

Aperture reduction Due to wrongly positioned movable devices such as collimators, vacuum valves, injection and matching screens, RF and safety stoppers, roman pots and alignment mirrors. There are almost 500 movable devices that can reduce or completely close the physical aperture at LHC [26].

Debunching Due to a failure in the RF systems, the beam loses the bunched structure, i.e. particles start to fill the entire ring. This entails losses at the extraction of the beam from the machine if the beam abort gap is affected (Chapter 3).

Table 2.5 depicts some of the failure chains and the resulting beam loss category (with regard to beam lifetime).

Table 2.5: *Examples of failure chains (adapted from [26])*

Failure	Phenomenum	Loss category
Change in magnetic field	Deflection/defocusing	Slow to ultra fast
Change in momentum (caused by failure of RF system)	Deflection/defocusing	Fast
Loss of vacuum	Scattering	Slow
Wrong position of movable objects	Aperture reduction	Slow to fast

2.2.2 Magnet Quench

Quench refers to the resistive transition from the superconducting to the normal conducting state of sc material [37]. The quench of a magnet (i.e. of its powering cable) is caused by an increase of the temperature beyond the critical value of the cable material. The critical temperature depends on the magnetic field and the current density respectively.

A quench and the related building of a resistive zone in the cable is localised at first and then propagates [26]. At the beginning, the release of the energy stored in the circuit is therefore concentrated in the very small volume of the quench origin (hot spot [37]). A quench in most of the LHC sc magnets (without intervention by the MPS) quickly leads to severe damage of the cables and coils due to local overheating. Without intervention, the temperature in the hot spots could increase to 1000 K within less than 1 s.

Quenches can be divided into different types, depending on the causing mechanism. Sonnemann [37] distinguishes training quenches (caused by cable motion and the related friction and heat dissipation [25]), beam-induced quenches, heater-induced quenches and quenches caused by cryogenic failure. In view of the beam losses occurring even during nominal operation, the beam-incuded quenches are particulary relevant in terms of machine protection. Quenches are expected to be the most immediate undesired consequence of increased beam loss [26].

Table 2.6 provides the quench level and the level of direct damage of a main dipole magnet for fast beam loss (occurring within a few milliseconds in case of very specific failures). At injection energy of 450 GeV, a number of $5x10^{9}$ protons lost in the magnet is expected to cause a quench. At top energy of 7 TeV, $5x10^{6}$ protons (i.e. a fraction of 10^{-8} of the beam) are suffient to quench the magnet, due to lower critical temperature and higher particle momentum [25]. In case of an increased beam loss by three or four orders of magnitude, direct damage is caused.

Table 2.6: *Quench and damage levels of a main dipole magnet for fast beam loss [25]*

	at 450 GeV	at 7 GeV
Quench level	5x10^9 protons	5x10^6 protons
Damage level	some 10^{12} protons	some 10^{10} protons

For LHC operation, the beam losses have to be kept below the quench level of the magnets. This is mostly achieved by the collimators in the cleaning insertions IR3 and IR7. Additionally, in case of beam losses exceeding a predefined threshold (below quench level), the MPS triggers an emergency beam dump. The MPS subsystem in charge of the detection of fast beam losses is the Beam Loss Monitor System (Chapter 3).

Experience in existing accelerators with sc magnets shows that quenches still cannot be completely avoided [26]. The MPS therefore is designed to avoid damage resulting from a quench. The related MPS subsystem is the Quench Protection System (Chapter 3). It detects a quench and initiates protection measures (i.e. the extraction of the beams and the energy of quenched circuit) .

2.3 Terminology for Reliability Analysis

The reliability analysis of the MPS involves basic terms like *safety*, *reliability* and *availability*. Besides, it must take into account common design principles such as *redundancy*, *fault tolerance*, *fail-safe* and *self-monitoring*. These terms and in particular their interrelations easily cause confusion.

For the traceability of both the developed method and the resulting analysis, a consistent understanding of the underlying terminology is essential. This section provides the terminology framework of the thesis and specifies the most relevant terms and their interrelations. Where possible, standard definitions are used as a basis. The section also serves as a roundup of the previous sections.

The framework has been previously published as part of a conference paper [38].

2.3.1 Machine, Detector and MPS

From a reliability analysis point of view, it is helpful to divide the LHC systems into three functional groups, which are hereafter referred to as *Machine*, *Detectors* and *MPS*. The Machine includes all LHC components and systems that represent the essential equipment needed to provide colliding beams. The beams are regarded as part of the machine. The Detectors cover the equipment needed to collect the data on the collisions. The MPS spans the components and systems that, while not being directly involved in the machine function, ensure machine safety.

This thesis focuses on Machine and MPS, the Detectors are not within the scope of this work. However, their inclusion to the frame is straightforward.

2.3.2 Machine Safety

Machine safety implies machine operation under nominal conditions. In case of non-nominal machine equipment or beam conditions, the MPS performs an *emergency beam dump*. Without MPS intervention, non-nominal conditions may lead to machine damage. Machine safety thus depends on the MPS. It is compromised by a non-nominal condition of the MPS.

2.3.3 MPS Reliability and Beam Availability

Reliability refers to the ability 'to perform a required function under given conditions for a given time interval' [39]. For the MPS, the required function is the performing of emergency beam dumps at non-nominal machine conditions (see above), the related time interval starting at beam injection.

With regard to that function, the MPS can fail in two ways, 1) missing an emergency dump (*missed emergency beam dump*) or 2) performing a dump at nominal conditions (*false beam dump*). While both affect beam availability, only the former involves compromised machine safety thus leading to machine damage.

Availability refers to the ability 'to be in a state to perform a required function (...) at a given instant of time (...)' [39], depending on the combined aspects of reliability and maintenance. According to [40], an available system finds itself in an up-state as opposed to a down-state. *Beam availability* in the context of this thesis relates to the presence of colliding beams.

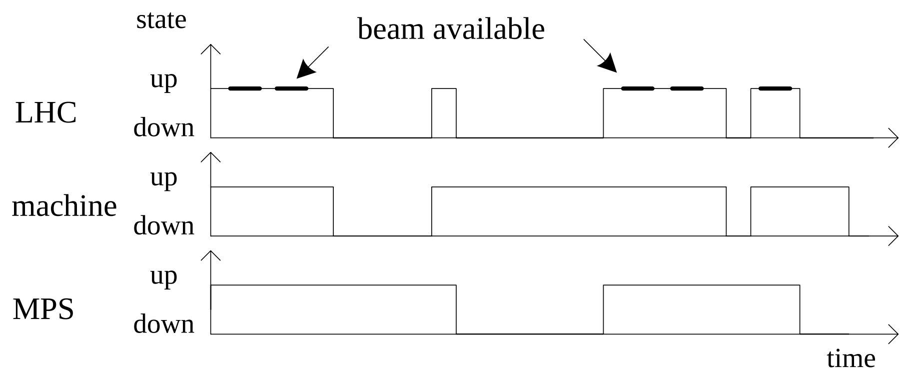


Figure 2.6: *Dependence of beam availability on machine and MPS availability [38]*

Figure 2.6 illustrates the dependence of the beam availability (as part of the overall LHC availability) on the availability of the machine and MPS. Their up-state corresponds to their finding themselves at a point of the nominal operational cycle. Downtime names the time beyond nominal cycles, i.e. where operation is interrupted due to repair or other measures to restore nominal operation conditions. The downtime of the machine and MPS sums up to the overall LHC downtime. Following this scheme, an availability of 100% is achieved by one nominal cycle following another. This ideal case also defines the denominator for beam availability as the sum of LHC uptime with beams available. By contrast, the total time is useful as denominator for comparison of different nominal cycles of an accelerator or of the performances of different accelerators of the same kind.

It is to be noted that the presented scheme does not cover machine safety. The up-states only relate to operation. Machine safety depends on the condition of the MPS during operation.

2.3.4 Machine Operation

The nominal machine operation cycle is scetched in Figure 2.7 (cp. Fig. 2.3). While *cycle* names the time period between two injections, *mission* (within this work) covers the time between injection and beam dump, i.e. the time where there are beams in the machine. The analysis assumes a nominal cycle of 12 hours.

In case of non-nominal machine conditions, a mission is ended early by an emergency beam dump performed by the MPS. Non-nominal conditions entail the risk of machine damage and downtime. While damage, besides costs for repair always includes down-

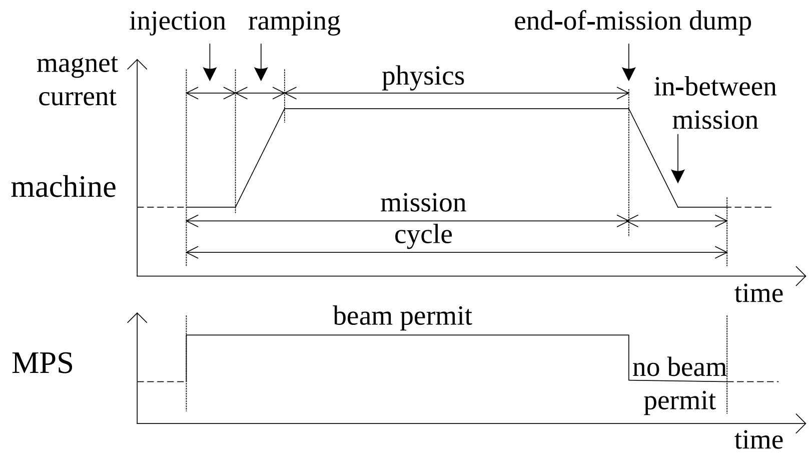


Figure 2.7: *Nominal operation cycle of machine and MPS. [38]. 'physics' corresponds to 'beam available' in Figure 2.6*

time, downtime does not necessarily imply damage.

One of the most relevant non-nominal machine condition is the quench of a sc magnet, which without intervention of the MPS leads to damage of the magnet. The replacement of the damaged magnet takes at least one month of downtime. If damage is prevented through the intervention of the MPS, the recovery of nominal magnet condition still results in a downtime of several hours [25]. Besides, each quench implicates a wearout. Quenches therefore have to be avoided.

2.3.5 MPS Operation

The presence of beams in the machine requires MPS *beam permit* status, indicating that it is ready and no non-nominal machine conditions are detected. In case of a nominal machine and MPS cycle (Fig. 2.7), beam permit is granted for beam injection and maintained until the end-of-mission dump is triggered in the control room, which involves the withdrawal of the MPS beam permit leading to the dump. After successful testing and diagnostics, beam permit is provided for the next cycle to begin.

Failures in the MPS can be grouped into three main categories, 1) undetected failures, 2) failures that generate a dump request signal and 3) failures that generate a warning. All of these failures leave the related component in a state that interferes with the fulfilling of the required component function.

Undetected failures (*blind failures*) compromise machine safety. On the MPS component level, they prevent the treatment and transmission of an incoming dump request signal. On the global system level, the concurrence of non-nominal machine conditions and undetected faults of MPS component can result in a missed emergency dump. The MPS design features a wide range of redundancy to avoid this scenario. *Redundancy* is defined as the existence of more technical means than necessary for the required function [41] and is one of the design principles rendering a system *fault tolerant*, i.e. able to continue functioning ('service') despite a ('hardware or a software') failure [42].

Failures generating beam dump request signals (*false failures*) base upon the *fail-safe* principle included in the MPS design. Following that principle, such failures trigger a beam dump (*false beam dump*), thus passing the machine into a safe state [43]. The generation of the related beam dump request signals emerges either directly from inherent fail-safe design or indirectly from extra *(self-)monitoring* features. Monitoring means activity intended to observe the actual state of a component ('item'), usually carried out during operation ('in the operating state') [39], in order to detect failures. These failures and the related beam dumps compromise beam availability.

The third category includes failures that, unlike the failures leading to false dumps, are considered as not critical with regard to the MPS functionality and machine safety respectively. Their detection leads to a warning, as a hint for the maintenance measures after a beam dump. Faulty components are detected after a dump by diagnostics and testing. The time for their repair may contribute to LHC downtime (if it cannot be done in the shadow of other repair or in-between-mission activities).

Chapter 3

The LHC Machine Protection System

After the overview given in the previous chapter, this chapter provides a more detailed description of the LHC Machine Protection System (MPS). It focuses on aspects that are relevant with regard to the performed reliability analysis. The purpose is to deliver the insight needed for the reader to trace the MPS model presented in Chapter 4.

3.1 General

It is to be noted that the system description provided in the following sections is based on the MPS reacting to non-nominal conditions of machine equipment (hardware) or beams (i.e. scenario *emergency beam dump*). The MPS in the context of nominal machine conditions and internal failing is discussed in Section 3.6.

3.1.1 Structure

Figure 3.1 illustrates the structure of the MPS with the different subsystems. The core of the MPS is the Beam Interlock System (BIS, green). It receives signals from a multitude of so-called User Systems (grey) and transmits them to the LHC Beam Dumping System (LBDS, blue). The primary signal path hence starts in the User Systems, spans the BIS and ends in the LBDS. With the exception of dedicated Beam Loss Monitors (BLMs), the LBDS receives signal from the BIS only. This in particular means that the User Systems are connected to the LBDS through the BIS only. There is no direct connection.

The MPS design includes a wide range of redundancy. There is a multitude of critera

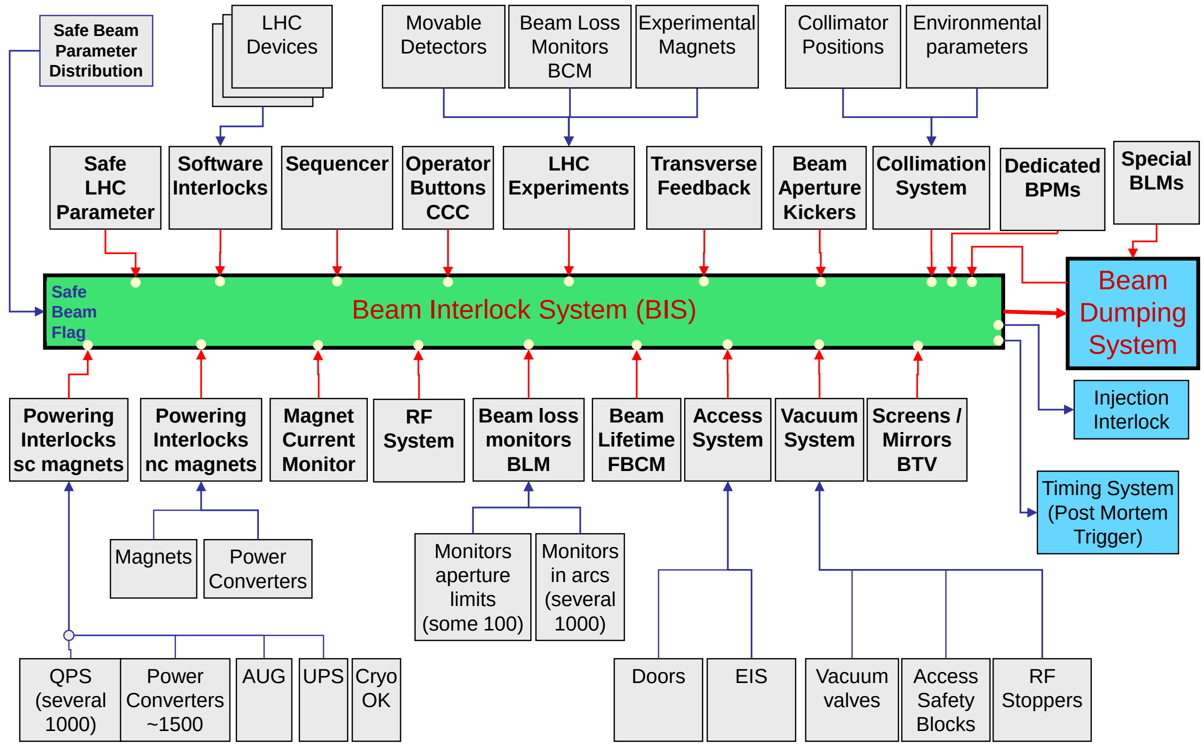


Figure 3.1: *MPS structure with arrows reflecting primary signal path. Grey: User Systems, green: Beam Interlock System, blue: Beam Dumping System. Courtesy of R. Schmidt*

to classify redundancy into different types. A possible classification of the redundancy included in the MPS is provided by Gomez Alonso [26].

The different subsystems depicted in Figure 3.1 are further introduced in the following section by means of their function.

3.1.2 Function

As pointed out in Chapter 2, the primary function of the LHC MPS is to perform a beam dump in case of non-nominal machine conditions. The function can be devided into three subfunctions, which can be assigned to the MPS subsystems (Fig. 3.1) according to Table 3.1.

The term *beam dump request* refers to a MPS subsystem (or component) communicating the need for a beam dump. It is an abstract term which does not completely reflect the real system design. The MPS is designed according to the active protection principle. It means that machine operation at any time requires clearing by the MPS.

Table 3.1: *Main MPS subsystems and their function*

System	Function
User Systems	Detection of non-nominal machine conditions, resulting in a beam dump request
Beam Interlock System (BIS)	Transmission of beam dump requests to the LBDS
Beam Dumping System (LBDS)	Extraction of the beams from the ring and their leading into absorbers (dump blocks) upon beam dump request

Clearing withdrawal corresponds to the triggering of a beam dump request. *Trigger a beam dump request* is also referred to simply as *trigger a beam dump*.

The clearing by a User System is referred to as *user permit*, while the clearing of the BIS is referred to as *beam permit*. Accordingly, the functional chain implied in Table 3.1 is described more precisely as follows. The detection of a non-nominal condition by a User System results in the withdrawal of its user permit. This in turn makes the BIS withdraw beam permit. The withdrawal of the beam permit leads to the extraction of the beams by the LBDS.

Table 3.2 lists the conditions leading to the triggering of a beam dump request by selected User Systems. The indicated abbreviations relate to the notation of Figure 3.1. The User Systems are grouped according to the monitored parameters, i.e. beam-related (upper part of the table), equipment-related (middle part, particularly magnet-related) and other (lower part).

3.1.3 Operation

As pointed out above, an emergency beam dump involves one (or more) User Systems, the BIS and the LBDS. The time between the occurrence of a non-nominal condition (detected by a User System) and the completion of the beam dump (i.e. the absorption of the beams by the dump blocks) is defined by the response time (also referred to as reaction time) of the involved subsystems. Table 3.3 shows the approximate minimum response time of the User Systems, the BIS and the LBDS. For the User Systems, it relates to the transmission time of the beam dump request from the location of non-nominal conditions (according to Tab. 3.2) to the BIS interface. For the BIS, the given value relates to the transmission time from the User System interface to the LBDS interface. For the LBDS, it relates to the time span between the arrival of the beam

Table 3.2: *User Systems and conditions resulting in the triggering of a beam dump request by the system (adapted from [26])*

User System	**Trigger condition**
Beam Loss Monitor System (BLMS)	Non-nominal beam losses
Fast Beam Current Change Monitor (FBCM)	Beam lifetime too small
Beam diagnosis invasive elements (BTV)	Unsafe position of element
Beam Position Monitor (BPM)	Beam position exceeding predefined values
Fast Magnet Current Change Monitor (FMCM)	Fast current change in normal conducting (nc) magnet circuit
Powering Interlock Controller (PIC)	Problem in sc magnet circuit
Quench Protection Systems (QPS)	Quench in sc magnet
Warm Magnet Interlock Controller (WIC)	Problem in nc magnet circuit
Collimation system	Unsafe position or temperature of collimator jaws
Vacuum System	Too high gas pressure in vacuum chamber or vacuum valve in beam
Experiment magnets	Powering failure
Experiment movable devices	Unsafe position of device
Safe LHC Parameters	Inconsistent parameters (e.g. Safe Beam Flag)
Access System	Access violation
CERN Control Room (Operators)	Operator decision

dump request at the interface and the completed beam dump.

The system with the fastest response on non-nominal beam condition is the BLMS with 80 μs. The fastest response with regard to non-nominal machine equipment condition is provided by the FMCM with about 60 μs. The transmission of the beam dump request through BIS and LBDS takes 200-320 μs. The total response time of the MPS on a non-nominal condition sums up according to Figure 3.2. Since most of the User Systems feature a response time in the range of milliseconds, they normally contribute the major part to the total response time.

The synchronisation as part of the Beam Dumping System (LBDS) process refers to the synchronisation of the kicker magnets with the gap in the beams. The deflection

Table 3.3: *Response time of the User Systems, the BIS and the LBDS (adapted from [26])*

System	**Approx. minimum response time**
Beam Loss Monitor System (BLMS)	80 μs
Fast Beam Current Change Monitor (FBCM)	$>$ 1 ms
Beam diagnosis invasive elements (BTV)	100 ms
Fast Magnet Current Change Monitor (FMCM)	60 μs
Powering Interlock Controller (PIC)	100 μs - 10 ms
Quench Protection Systems (QPS)	15 ms - 250 ms
Warm Magnet Interlock Controller (WIC)	0.1 - 100 ms
Collimation system	10 - 100 ms
Vacuum System	10 - 100 ms
Experiment magnets	10 - 100 ms
Experiment movable devices	10 - 100 ms
Safe LHC Parameters	100 ms
Access System	100 ms
CERN Control Room (Operators)	1 - 10 s
Beam Interlock System (BIS)	20 - 120 μs
Beam Dumping System (LBDS)	$<$ 180 μs

of a continuous beam by the kicker magnets from the ring orbit into the extraction lines would cause the scattering of the beam over the respective deflection angle across machine equipment. In order to allow for a 'clean' deflection, i.e. without beam loss, the beams feature a so-called abort gap [26]). It refers to a nearly particle-free section in the beam, which provides the time window for the kicker magnets to build up the magnetic field to deflect the beam. The beam takes 89 μs for one turn in the ring, i.e. the abort gap passes the extraction kicker magnets (in IR 6, Chapter 2) every 89 μs. The time delay related to synchronisation depends on the position of the gap in the ring at the moment of the beam dump request. In case of the gap just passed, the LBDS has to retard the triggering of the kicker magnets for another 89 μs, otherwise less. The extraction (abort) itself obviously takes the time of one turn. Therefore, an emergency beam dump cannot be performed in less than 170 microseconds.

It is important to note that the listed response times of the User Systems relate to the detection of non-nominal conditions with regard to the monitored parameters. In order to get the total response time upon the occurrence of a failure, the time span between failure occurrence and detection of resulting non-nominal conditions has additionally

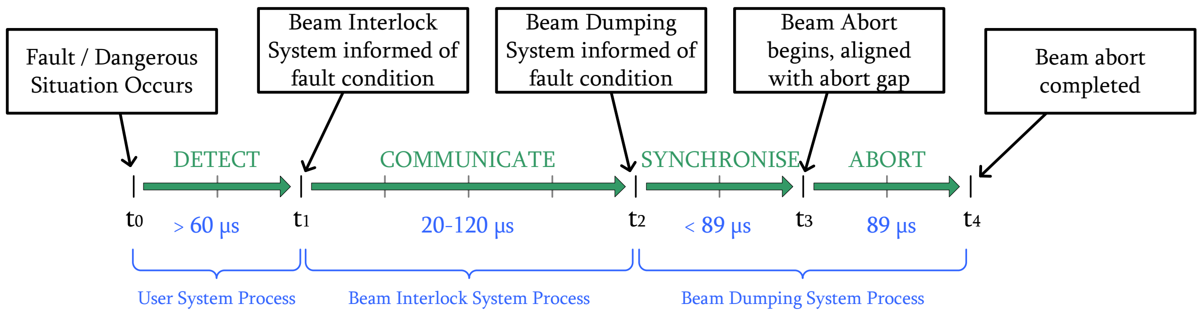


Figure 3.2: *Succession of events after a failure and response times of the MPS. Courtesy of B.Todd*

to be taken into account. In case of a failure with an immediate effect on monitored parameters, this time span is short, while for indirect effects it is longer. A failure in a magnet may lead to non-nominal conditions that are immediately detected by hardware monitoring devices such as FMCM, while the indirect effect of the failure on the beams may be detected by the BLMS later.

The following sections further introduce the BIS, the BLMS (as one of the largest User Systems, monitoring the beams) and the LBDS. The description focuses on the (primary) signal path of the beam dump requests and the components involved, which represents the basis of the developed modelling approach introduced later in this work (Chapter 4).

3.2 Beam Interlock System

A simplified structure representation of the BIS is illustrated in Figure 3.3. The illustration represents the basis for the MPS model (Section 4.2).

The BIS, as the MPS core subsystem connecting the User Systems to the LBDS, is distributed over the whole circumference of the LHC. Its ring structure is closely related to the LHC layout (Fig. 2.1). The bottom-mid part of the illustration corresponds to IP1, the top-mid part to IP5.

3.2.1 Structure

For the further description of the BIS structure, Figure 3.4 shows the bottom-mid part of the above illustration in detail. The BIS consists of Beam Interlock Controllers (BICs,

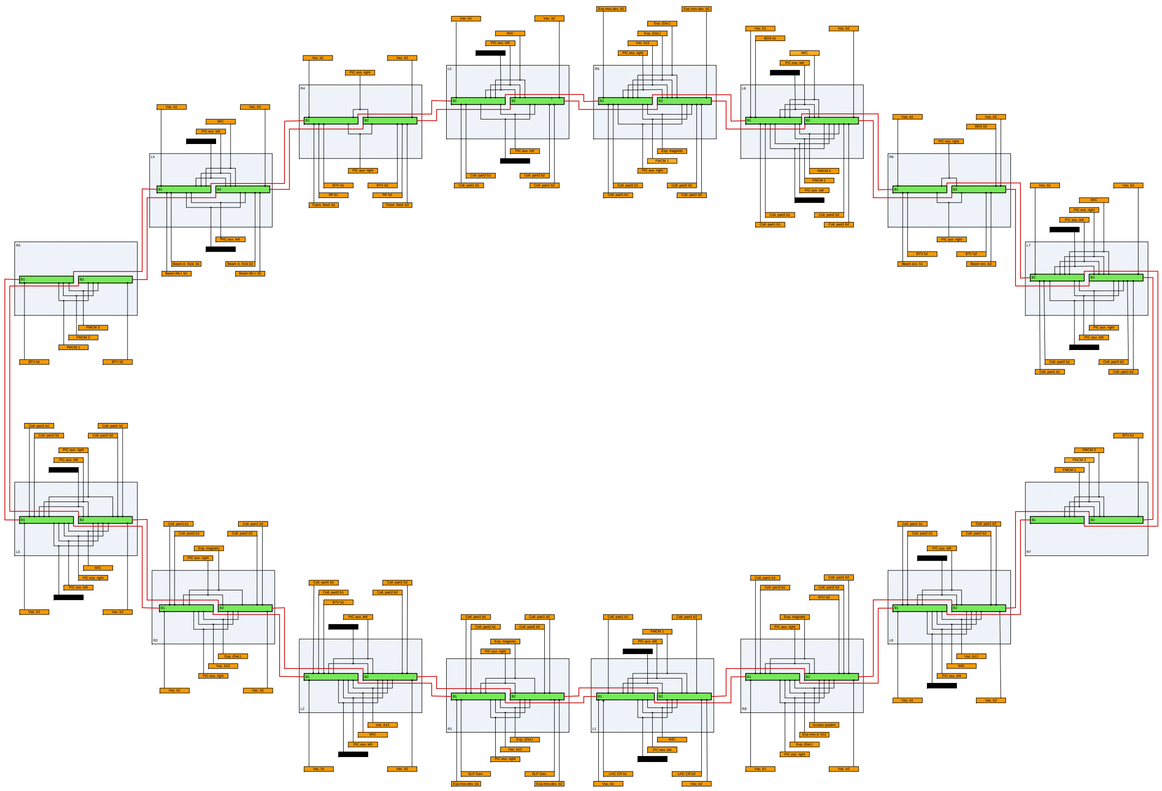


Figure 3.3: *Schematic structure of the BIS*

green) and User Interfaces (CIBUs, orange/black).

Two BICs, one for each beam, are assigned to each side of an IP. In Figure 3.4, the BICs related to beam 1 are denoted as B1, while B2 indicates the BICs related to beam 2. R1 refers to the Versa Module Europa (VME) crate to the right of IP1 (containing BICs B1 and B2), L1 to the VME crate to the left of IP1 [1]. The BICs of each beam are individually connected in series ('daisy-chained') by optical fibers around the ring, resulting in two independent signal loops ('beam permit loops', red lines [1, 26]). The loops start (and end) in the BICs at IP6, where the interfaces to the LBDS are located (Fig. 3.3, top-right).

The CIBUs represent the interfaces between the User Systems and the BIS. The CIBUs are grouped into two types of interface units, namely CIBUD and CIBUS. A CIBUD unit contains two sets of interface electronics (i.e. CIBUs), providing a double connection for an independent interlock of the two beams [1]. A CIBUS unit provides a single connection for the simultaneous interlock of both beams (or the interlock of a single beam). In Figure 3.4, the boxes with a branched connection line to both B1 and B2 relate to CIBUS. The black boxes represent the CIBUS interfacing the BLMS branches

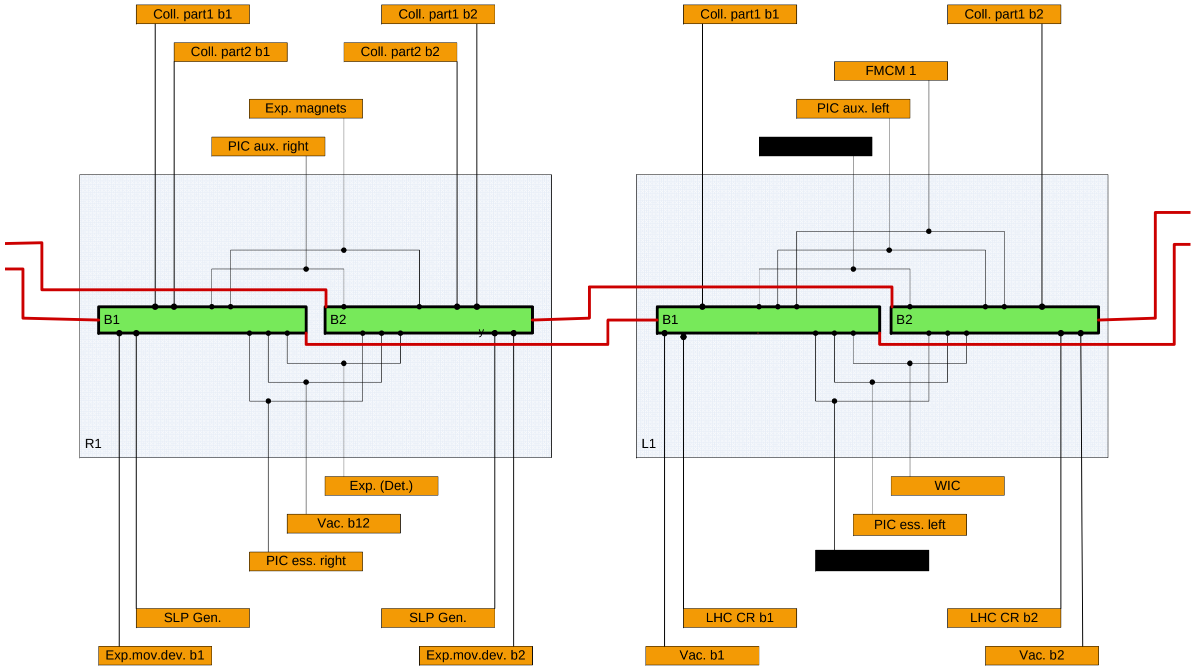


Figure 3.4: *Detail of the BIS structure around IP1. Green: BIC, orange/black: CIBU*

(Section 3.3) .

The illustrations in Figures 3.3 and 3.4 include a series of simplifications. They imply a total number of 32 BICs (16 per beam), which corresponds to the design projected at an early stage of this work. Meanwhile, two additional BICs (one per beam) have been installed. They interface the CERN control room and are used for the end-of-mission beam dumps. Additionally, they interface the access system and the Software Interlock System (SIS).

An additional simplification regards to the redundancy of the connection lines. In reality, there are two redundant signal loops (optical fibers) for each of the beams (i.e. the red lines in the illustration should be doubled, resulting in four loops in total). One fiber is used for clockwise dump request transmission (loop B), the other one for counter-clockwise transmission (loop A [1,26]). Each BIC hence transmits incoming beam dump requests to both of its neighbouring BICs. This means that the transmission of a dump request from any BIC to the LBDS interface (in IP6) only spans half of the ring at most, for the benefit of a limited transmission time. The redundancy of the beam permit loops applies throughout to the user permit loops (black connection lines in Figure 3.4 [1]).

3.2.2 Function and Operation

The function of the BIS is illustrated by Figure 3.5. The boxes in the left represent CIBUs, the ones in the middle BICs (for beam 1 and 2) and the ones in the right the interfaces to the LBDS. The BICs convert the USER_PERMIT signals provided by the connected User Systems into local BEAM_PERMIT signals [1]. The underlying logic is such that the local BEAM_PERMIT of a BIC is TRUE only if all of the connected USER_PERMITS are TRUE. Accordingly, the BEAM_PERMIT signal delivered to the LBDS interface (for each of the beams) is TRUE only if the local BEAM_PERMIT signals of all the related BICs is TRUE.

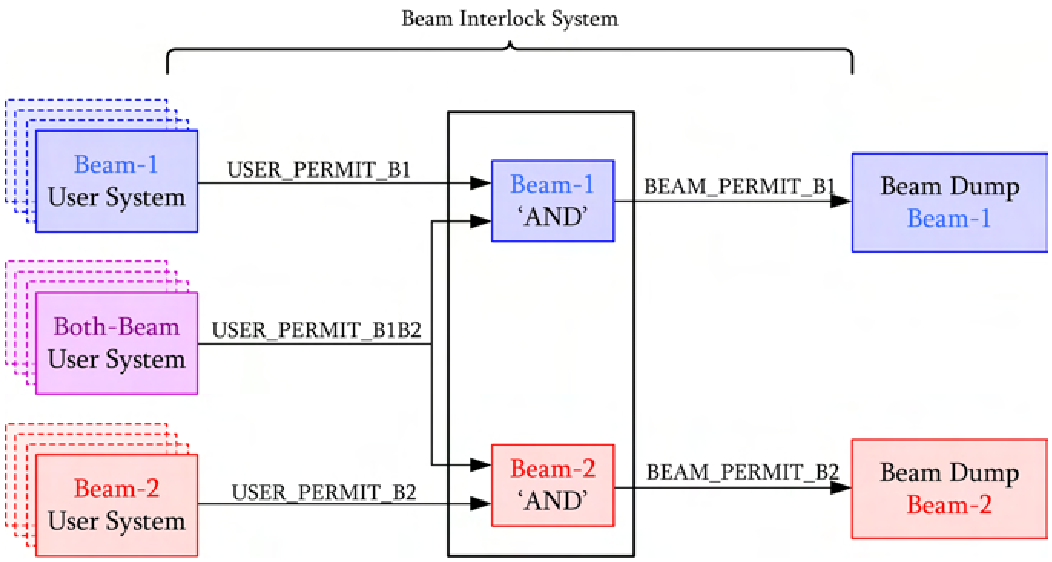


Figure 3.5: *Beam 1, beam 2 and both-beam interface to the BIS [1]*

The beam permit loops are realised by means of a square signal of approximately 10 MHz. It is provided by a generator located at IP6 and passes through the daisy-chained BICs around the ring. The generation of the signal (i.e. frequency) requires the presence of the signal at the end of the loop, a condition which is referred to as *closed-loop operation* [1].

The local BEAM_PERMIT signal of the BIC controls a switch on the beam permit loop. If the local BEAM_PERMIT signal is TRUE, the switch is closed, allowing the frequency to pass the BIC. The switch is opened in case of local BEAM_PERMIT signal FALSE, thus interrupting the beam permit loop. The resulting absence of the BEAM_PERMIT signal corresponds to a beam dump request transmitted to the LBDS interface. An interrupted loop inhibits the further generation of a frequency, due to the closed-loop operation mentioned above.

It is to be noted that the illustration in Figure 3.5 corresponds to Figures 3.3 and 3.4 in terms of the level of abstraction. The redundancy in the USER_PERMIT signal and the beam permit loops (loops A and B) are not included. The mentioned logic however applies to redundant loops accordingly. This in particular means that the BEAM_PERMIT signal in loop A is TRUE only if the signal in loop B is TRUE as well, and vice versa [1]. The withdrawal of the beam permit in one loop results in the permit withdrawal in the other loop. It is important to note that this only applies to the redundant loops A and B. It does not necessarily apply across the loops of the different beams.

A more detailed description of the BIS is provided by Todd [1].

3.3 Beam Loss Monitor System

Figure 3.3 indicates the interfaces between the BLMS and the BIS as black boxes. There are 16 interfaces in total, assigned in eight pairs to the left of the eigth IPs. Figure 3.6 shows the simplified structure of the eight BLMS branches, reflecting the primary signal path from the beam loss monitors to the interface (CIBUS). The illustration allows for a basic understanding of the BLMS and at the same time represents the basis for the MPS model (Section 4.2).

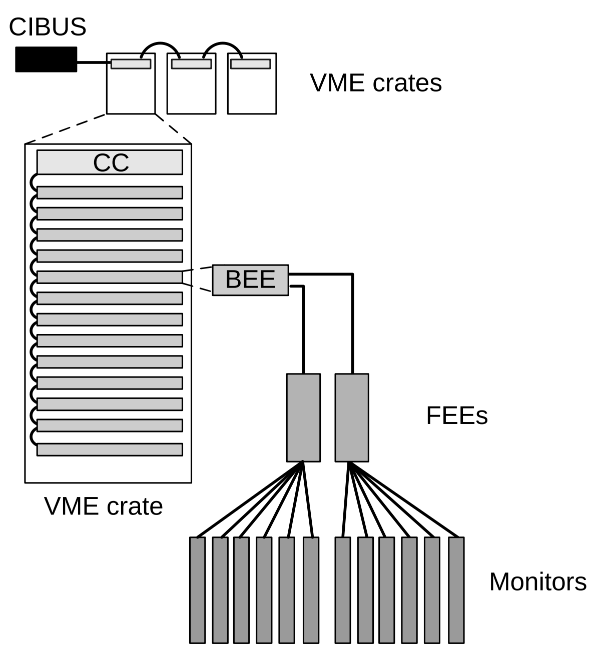


Figure 3.6: *Schematic structure of a BLMS branch*

3.3.1 Structure

The signal path starts in the monitors (Fig. 3.6, bottom-right), which represent the BLMS-related sensors of the MPS. The signals of six monitors are sent to the same Front End Electronics board (FEE). Two FEEs each are connected to a Back End Electronis board (BEE). Thirteen daisy-chained BEEs and a Combiner Card (CC) build up a VME crate. The signals of three daisy-chained VME crates are sent to a CIBUS (interface to the BIS). One BLMS branch according to Figure 3.6 implies a total number of 468 monitors, 78 FEEs, 39 BEEs, 3 CCs and 3 VME crates.

The presented numbers are average numbers. In reality, one FEE may get input signals of up to eight channels. A VME crate can contain up to 16 BEEs and IP7 features four VME crates (instead of three).

The MPS in total consists of about 4000 beam loss monitors (BLMs) of two main types [26]. About 3500 monitors are nitrogen-filled Ionisation Chambers (ICs), while the remaining 500 are Secondary Emission Monitors (SEMs). The monitors can be grouped into families according to their location in the accelerator ring [26]:

BLMS At locations with physical (mechanical) aperture limitation or other critical locations around the machine.

BLMC At the collimators (physical aperture limitation) in the cleaning insertions (IR3 and IR7) [25].

BLMA At the quadrupoles in the arc.

Additionaly, there are monitors that are used for dedicated beam studies and have no relevant role with regard to machine protection [26].

Figure 3.6, besides averaged numbers of components, includes simplifications with regard to the redundancy of connection lines. In particular, the redundacy of the optical fiber between FEE and BEE is not indicated. It connects the FEE located in the underground tunnel to the BEE at the surface. Furthermore, there is redundancy in the loop connecting the CCs to the respective CIBUS.

One may note a discrepancy between Figures 3.6 and 3.3 with regard to the number of CIBUS (indicated as black boxes). The illustration in Figure 3.6 implies a BLMS branch to interface to the BIS through one channel only, i.e. all signals are assumed to be transmitted to the same CIBUS (one CIBUS per branch). In reality, there are two separate channels in a branch, each assigned to a separate CIBUS. They sum up to the 16 BLMS-related CIBUSs depicted in Figure 3.3. This characteristic is further discussed in Section 3.3.3.

3.3.2 Function and Operation

The function of the BLMS is to detect non-nominal beam loss and communicate it to the BIS through the withdrawal of the user permit, i.e. a USER_PERMIT signal with logical value FALSE (as opposed to TRUE).

The monitors at their respective location detect secondary particles which result from the impact of lost protons on the surrounding equipment (e.g. magnet, collimator) [20, 26]. They provide a current signal to the FEE which is proportional to the intensity of the secondary particles. In the FEE, the current signal is digitised by a current frequency converter (CFC). The digital signal is transmitted from the FEE (located in the underground tunnel) to the BEE (at the surface) through an optical fiber.

In the BEE, the signal (indicating the number of lost protons per time and area) is compared to a threshold defined in a table. The BEEs and the CC of a VME crate are connected by a signal loop through VME backplane lines. In case of a signal above the related threshold, the respective BEE switches the loop open. As a result, the loop connecting the CCs and the CIBUS is swiched open by the related CC. This in turn causes the CIBUS to communicate user permit withdrawal to the BIS (through USER_PERMIT signal with value FALSE).

The thresholds used for comparison in the BEE depend on the beam energy. The energy of the beam present in the accelerator is communicated from the CCs to the BEEs by means of a signal that represents a secondary input to the BEE. In order to prevent quenches, the thresholds of BLMs close to sc magnets are defined to values below the magnet quench levels at the different beam energies.

The BLMCs and the BLMSs, due to their location, are considered to be critical for machine protection. They are installed at locations where most fast beam losses are expected. The response time of these monitors and their channels therefore needs to be small. The response time of the monitor channels depends on the time resolution underlying the digitalisation and multiplexing in the FEE. The BLMCs and BLMSs provide a time resolution of about 40μs, corresponding to half a turn [26].

More detailed information on the BLMS is provided by Guaglio [20].

3.3.3 Channel Masking

Unlike implied in the structure shown in Figure 3.6, each of the eight BLMS branches interfaces to the BIS with two channels. They are referred to as *maskable* and *unmaskable* and are linked to the BIS through two separate CIBUSs. Figure 3.7 shows the adapted representation of the structure of a BLMS branch, providing the basis for a possible MPS model extension with regard to channel masking.

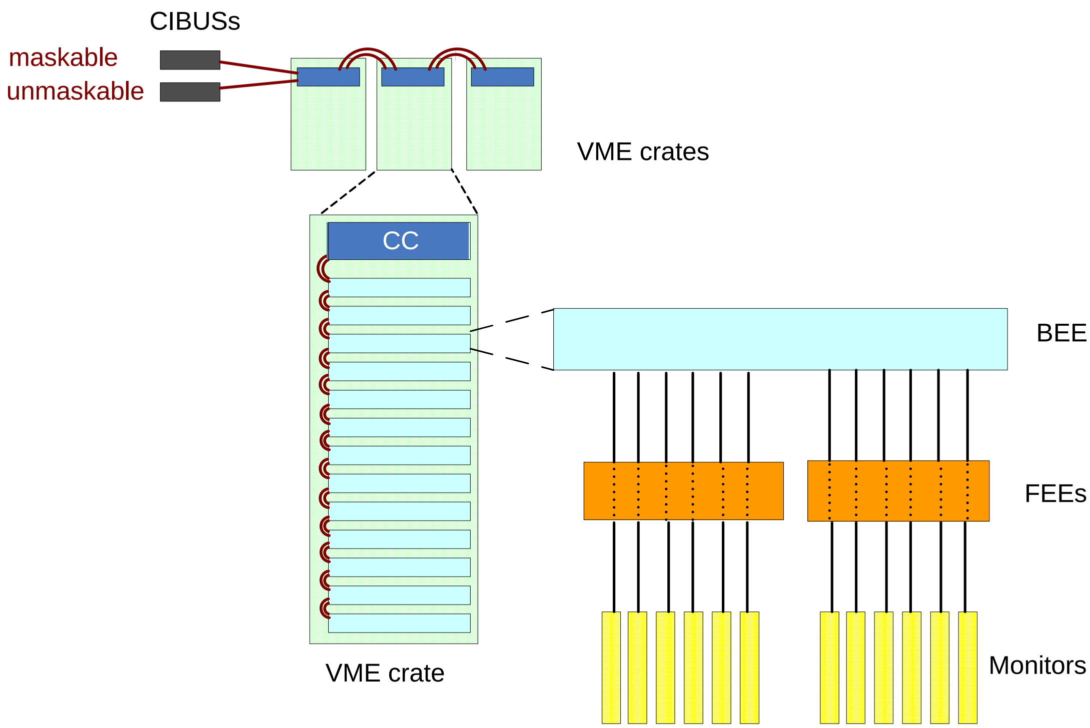


Figure 3.7: *Adapted structure representation of a BLMS branch*

Channel masking refers to the MPS operational option that is related to the separate channel design. In the BEEs, the signals of the different monitor channels are assigned to either the maskable or unmaskable channel. The BICs can optionally be set to ignore the logical value of the USER_PERMIT signals of the maskable channel. This means that the signals of the monitors assigned to the maskable channel are not taken into account in the local BEAM_PERMIT, i.e. their channels are *masked*. The masking may be applied to BLMs that are considered or observed to be not critical with regard to machine protection.

In any case, the masking of channels requires the SAFE BEAM FLAG in the BICs (Fig. 3.1) to be TRUE. The SAFE BEAM FLAG (SBF) being true indicates low beam energy and intensity, i.e. a beam condition below damage thresholds with regard to the energy stored in the beams [1]. The SBF depends on information provided by the Safe Machine Parameters System and sent to the BICs through the CERN General Machine Timing Network [26].

The channel masking allows for more flexible operation, especially during beam commissioning [26]. In particular, it would faciliate operation in the case of monitors significantly contributing to the triggering of false beam dumps. During nominal operation however, the masking will not come into action, since beam conditions with damage potential are reached during injection phase already (due to the high beam intensity).

3.4 Beam Dumping System

The LBDS is briefly introduced by means of the simplified schematic structure shown in Figure 3.8. The MPS model introduced in Section 4 does not cover the LBDS but includes the BLMS and the BIS only. However, the given illustration provides a basis for a possible MPS model extension with regard to the LBDS.

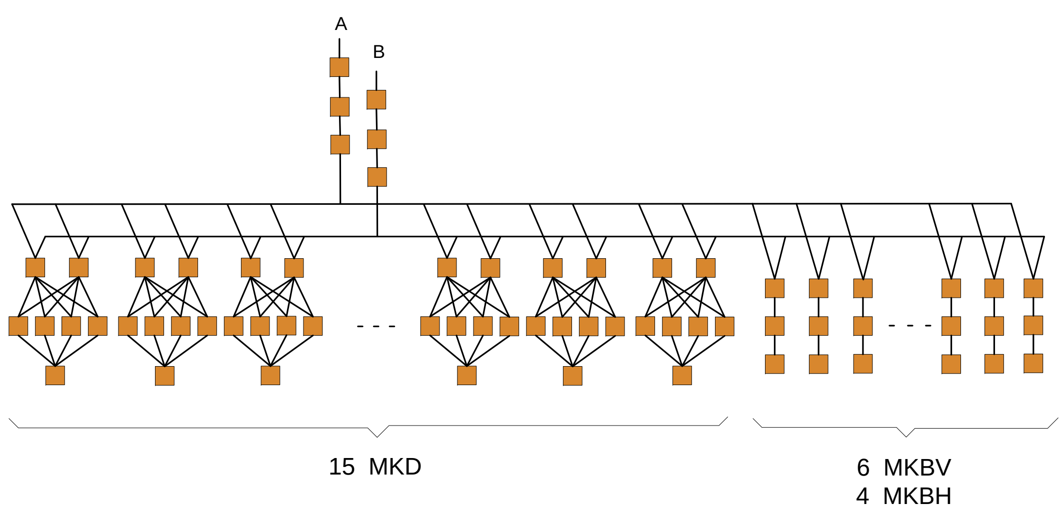


Figure 3.8: *Schematic structure of the signal path in a LBDS branch (including lines A and B)*

The LBDS consists of two independent branches, one for each beam. In IP 6, each of the branches interfaces to the BIS through two Trigger Synchronisation Units (TSUs, one for each line A and B), which monitor the beam permit signal of the related beam permit loop. In case of a BEAM_PERMIT transition from TRUE to FALSE, the TSU creates a signal synchronised with the beam abort gap, which is distributed to the 15

MKD and 10 MKB assemblies (i.e. kicker magnet assemblies, Fig. 3.8). In each magnet assembly, a power switch is activated, connecting the kicker magnet to the power supply. This results in a fast ramping of the magnetic field in the magnets, entailing the extraction of the beam from the ring and its leading into the dump block.

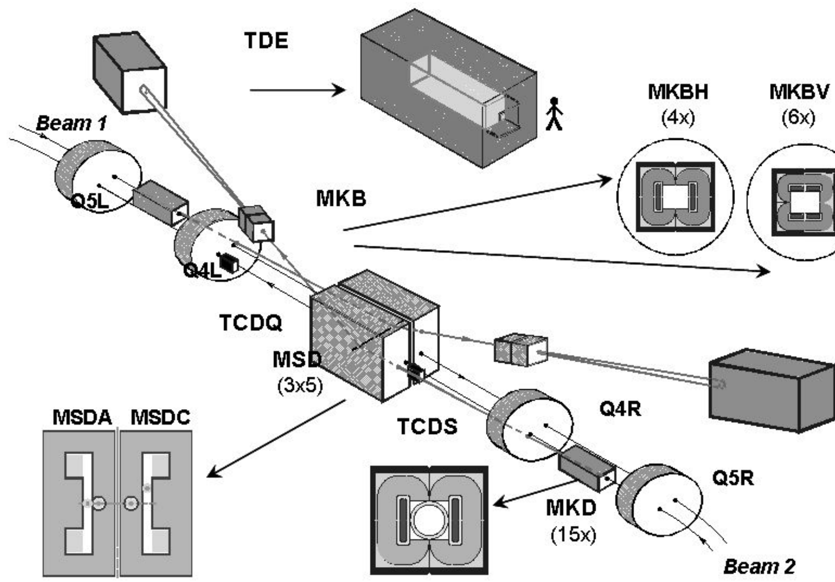


Figure 3.9: *Schematic layout of a LBDS branch ([21])*

Figure 3.9 illustrates the position of the different magnets in the ring and the extraction lines. The 15 fast kicker magnets, the MKDs, are located in the ring. Their magnetic field is built up within three microseconds, which corresponds to the length of the beam abort gap. The MKDs deflect the beam by 280 microrad in the horizontal plane, from the ring orbit onto the extraction trajectory [21, 26]. The deflected beam crosses 15 septum magnets (MSD system), which deviate the beam by 2.4 millirad in the vertical plane. In the extraction line, the beam is diluted by the field of 10 MKB kicker magnets, in order to reduce the energy density on the dump block (TDE). Four of the MKB magnets (MKBHs) deflect the beam horizontally, the remaining six magnets (MKBVs) vertically. This results in the beam impacting the dump block along a spiral transverse trajectory (i.e. in a 'e' shape [21]). The extraction line between the MSDs and the TDE is about 700 meters long. The distance, besides the MKBs, additionally contributes to the spreading of the beam.

The requirements for a clean extraction without non-nominal beam losses can be summarized as follows [26]:

- 14 out of 15 MKD kicker magnets operational (14-out-of-15 redundancy)

- Beam abort gap free of particles
- Ramping of the MKDs synchronised with beam abort gap
- Field of the MKD and MKB kicker magnets corresponding to beam energy
- Offset of the closed orbit in IR6 less than 4 millimeters.

A detailed description of the LBDS is provided by Filippini [21].

3.5 Other Systems and Devices

The following sections briefly introduce some additional monitoring devices and User Systems (Fig. 3.1), divided into beam-related and hardware-related (equipment-related) systems.

3.5.1 Beam-Related Systems

In terms of beam-related systems, the following devices for beam monitoring and diagnostics are mentioned by Gomez Alonso [26]:

Beam Current Monitors (BCMs) There will be two types of BCMs installed in the LHC, namely DC and Fast Beam Current Transformers [44]. They monitor the beam current in a range from 10 μA (single bunch) to over 0.5 mA (nominal beam). The Fast Beam Current Transformers are additionally used for lifetime and bunch to bunch intensity measurements. The beam lifetime measurements serve machine protection purposes [26].

Beam Position Monitors (BPMs) There are 1032 BPMs installed in the LHC, measuring the beams in both the horizontal and vertical plane [44]. They are essential for the adjustment and control of the closed beam orbit and can detect fast changes of the orbit [26]. Currently, most of the BPMs are not implemented for machine protection use. However, some dedicated BPMs located at IP6 are used to monitor the orbit offset, which is relevant for a clean beam extraction (as mentioned above). These BPMs trigger a beam dump request if the closed orbit offset exceeds three millimeters (an offset of less than four millimeters is required for clean extraction).

3.5.2 Hardware-Related Systems

The hardware-related systems particularly refer to the protection of the magnets. They can be divided into the systems protecting the sc magnets and the systems for the protection of the nc magnets.

There are two independent powering interlock systems that collect the relevant information regarding the powering of the magnets and transmit beam dump requests to the BIS if needed. The following sections briefly introduce the two powering interlock systems, as well as the Quench Protection System (QPS, monitoring sc magnets) and the Fast Magnet Current Change Monitors (monitoring nc magnets).

A more detailed description on the powering scheme of the LHC magnets is provided by Gomez Alonso [26].

3.5.2.1 Powering Interlock System for Superconducting Magnets

The Powering Interlock System for sc magnets consists of 36 Power Interlock Controllers (PICs) in total. The PICs receive input from the QPS and the power converters, as well as of the AUG (Arret d'Urgence Generaux, Emergency Stop System [45]), UPS (Uninterruptible Power Supply) and cryogenics (Fig. 3.1).

The PICs interface the QPS, which monitors specific magnets, with the related power converters. They grant powering permit in case of nominal conditions of the magnets and the components involved in the electrical circuits (e.g. power converters). In case of a non-nominal condition, the PIC initiates a power off. The extent of the power off depends on the criticality of the affected circuit with regard to the stored energy and the impact on the beam. For main dipole and quadrupole magnet circuits for example, a failure in a magnet implies the powering off of the complete powering subsector [26], i.e. of all its circuits. Besides the initiation of a power off, a beam dump request is transmitted from the PIC to the BIS.

In case of a failure in a circuit storing less energy and with less impact on the beam (e.g. circuits of auxiliary magnets like orbit correctors or multipolar compensators), the power off is limited to the affected circuit and no beam dump request is generated [26].

Detailed information on the Powering Interlock System is provided by Zerlauth [45].

3.5.2.2 Quench Protection System

In principle, the MPS is designed to detect non-nominal conditions (and perform a beam dump) before they results in a magnet quench. However, experience in existing accelerators with sc magnets shows that quenches cannot be completely avoided [26].

The main sc magnets are protected by the QPS, which detects a quench and initiates the safe discharge of the energy stored in the magnet (i.e. the magnetic field) and its electrical circuit. At the same time, it triggers a beam dump request, which is transmitted to the PIC that covers the affected magnet and its circuit respectively.

The mechanism is summarised by Gomez Alonso [26] as follows:

Detection of the quench The resistive voltage across the coils of a magnet is measured in real time. If it exceeds a predefined threshold for a given time interval [25, 37], the switching off of the power converter is initiated and a beam dump request is triggered. In case of sc magnets (i.e. main dipoles (MBs), main quadrupoles (MQs) and other high current magnets), resistive heaters are fired additionally.

Distribution of the energy over the coil The resistive heaters installed along the sc magnet coils increase the temperature of the coil, hence forcing the entire coil to quench. Thus, the release of energy is distributed over the coil instead of being concentrated in the original quenching point (hot spot).

Deviation of the current out of the quenched coil In the sc main circuits with several magnets in series (i.e. MB and MQ circuits), a sc bypass diode is installed in parallel with each magnet. The rising resistance in the quenched coil entails a voltage across the coils. Upon the voltage exceeding the conduction threshold of the diode, the current bypasses the quenched magnet through the diode thus avoiding further heating of the quenched coil.

Extraction of the energy from the circuit In the sc main circuits, the current decay is slow. In a MB circuit (including 154 magnets) for example, the natural time constant for the discharge is several hours due to the large inductance and marginal resistance of the sc circuits [26]. Therefore, in order to reduce the time constant, discharge resistors are switched in series with the magnets of the quenched MB circuit. Thereby, the time constant for the discharge is reduced to about 100 s [25]. Recently, it has been further reduced to 50 s for operation with a beam energy up to 3.5 TeV. The energy is safely released in the resistors.

For the protection of the LHC main dipole and quadrupole circuits, both resistive heaters and bypass diodes are used [25]. For many stand-alone sc magnets and their circuits, the heaters are sufficient and diodes are not required.

Due to the time needed for the resistive heaters and the extraction switches (in case of MB circuits) to become effective, the beam dump is usually performed before the magnetic fields decay [25]. A failure within the QPS, for example the firing of the heaters without prior quench, leads to a beam dump before the beam is affected.

A more detailed description of the QPS is provided by Vergara Fernandez [22]

3.5.2.3 Powering Interlock System for Normal Conducting Magnets

The Powering Interlock System for nc magnets consists of Warm Interlock Controllers (WICs). The WICs receive input from temperature sensors installed in the nc magnets and from the power converters [26] (Fig. 3.1). The connection between a WIC and the thermoswitches of a magnet is realised by a current loop. The same applies to the connection between the WIC and the power converter.

The thermoswitch opens the loop if the temperature of the magnet coil exceeds a threshold. As a result, the WIC transmits a beam dump request to the BIS and switches off the power converter of the magnet. Likewise, the WIC triggers a beam dump request in case of a failure of a power converter.

3.5.2.4 Fast Magnet Current Change Monitors

The detection of failures in the powering of magnets through power converters and WICs (as explained above) takes a comparatively long time. Failures that result in a fast change of the magnet current (causing beam losses within a very short time) ask for additional monitor devices. The most critical nc magnets are therefore additionally controlled by Fast Magnet Current Change Monitors (FMCMs) that monitor the electrical circuits powering the magnets [25]. The FMCMs are able to detect fast current changes within a very short time and transmit dump requests to the BIS directly (Fig.3.1). They are based on the measurement and processing of the voltage signal across the magnets [26].

3.5.2.5 Other Systems and Devices for Hardware Monitoring and Diagnostics

The systems introduced above provide specific protection against magnet-related failures. Failures of other devices however may also lead to unsafe operation. Therefore, most of the hardware required for LHC operation integrates self-monitoring, generating a beam dump request in case of incorrect settings or parameters.

The following hardware-related systems and devices are able to trigger a beam dump request [26] (cp. Fig. 3.1, Tab. 3.2):

Collimation system Position and temperature are controlled.

Experimental devices Position of the movable detectors and powering of the experimental magnets are checked.

Other movable devices Position is checked (OUT position required).

RF System Radio frequency and voltage are monitored.

Access system Access to the LHC tunnel is controlled (during operation the opening of any access door triggers a beam dump request).

3.6 Other Aspects of MPS Operation

The above system description focuses on the MPS reacting to non-nominal conditions of the beam or of machine equipment. This section adds the operational aspects with regard to nominal machine conditions and internal failing.

3.6.1 MPS and Nominal LHC Cycle

Prior to any beam injection into the LHC, clearing (i.e. beam permit) by the BIS is required. The absence of the beam permit inhibits injection. During a nominal LHC cycle, beam permit holds till the moment when an operator in the CERN control room triggers a request for an end-of-mission beam dump. The beam dump request is transmitted to the two related BICs (the '17th' BICs per beam), which withdraw the beam permit, resulting in a beam dump performed by the LBDS.

After any beam dump, the MPS remains operational. The magnet circuits at that point still store energy with damage potential. The machine equipment therefore during ramp down still requires protection by the hardware monitoring subsystems (e.g. QPS). Once all the equipment is in a safe state, the BIS may release the permit for post-mortem testing and diagnostics.

3.6.2 MPS and Internal Failures

A failure by definition is an event leaving a component in a (faulty) state that prevents it from fulfilling its function (Chapter 2). Failures occurring in MPS components, i.e. within the MPS, can be classified into two main categories, depending on their effect regarding system operation (adapted from [1]):

Blind failure Undetected failure. Incoming dump request signals are not treated and transmitted accordingly. A blind failure of a BIC for example could appear in an incoming USER_PERMIT signal with logical FALSE being unnoticeably decoded as TRUE [1].

False failure Failure that is covered by inherent active or passive fail-safe measures, triggering a beam dump request upon occurrence. The resulting beam dumps are referred to as *false beam dumps*. The term *false failure* in itself might be misleading. It it important to understand that it is derived from the denotation of its consequence (i.e. false beam dump). The interruption of an optical fiber of the beam permit loops for example would be referred to as false failure (including passive fail-safe measure), since the interruption automatically results in an absent signal which relates to a beam dump request.

A third failure category is introduced by Todd [1] and Guaglio [20]. It refers to failures that result in the generation of a warning, providing hints for maintenance actions after the mission. The interference of these failures (and the resulting faulty state of the component respectively) with the global performance of the MPS is considered to be acceptable for the remaining mission time. This consideration applies to the failure of a redundant component, for example [1]. The immediate impact of these failures with regard to the remaining mission time however is the same as for the *blind failures*, i.e. leaving the component in a faulty state.

The *blind failures* are potentially machine safety-critical (through missed emergency beam dump scenario), while the *false failures* interfere with beam availability (through false beam dumps). However, it is to be noted that each beam dump (false beam dump and also end-of-mission beam dump) carries a certain risk for machine safety (in case of unclean extraction).

More specific information on component failures can be found in the previous reliability studies on MPS subsystems [1, 20–22, 45].

Chapter 4

Model

In search of a modeling approach appropriate for the required reliability analysis of the MPS, two key aspects have to be taken into account:

- The MPS is hardly comprehensible as a whole due to its size and complexity
- Easy traceability of the model is crucial for a sustainable analysis

Consequentially, the model is required to describe the MPS on an adequate level of detail while allowing 1) the analysis to be broken down into segments that are easily and individually manageable and 2) the reliability and system specialists involved to keep track of the model coverage and the underlying assumptions and simplifications respectively.

The developed modeling approach is straightforward and considered to be appropriate with regard to the these requirements. While established approaches like fault trees and Markov models depend on abstract system description reflecting global functionality and state space, the developed approach is based on the component-by-component representation of the physical signal path. The approach uses component models as the basic modules for the system model, with the component models involving FTA and Markov models.

4.1 Concept

The model concept is explained by means of the system illustrated in Figure 4.1. Six (identical) components are linked representing a signal path from virtual component S

(start) to virtual component E (end). At the start and end of the signal path the system features 1-out-of-2 redundancy .

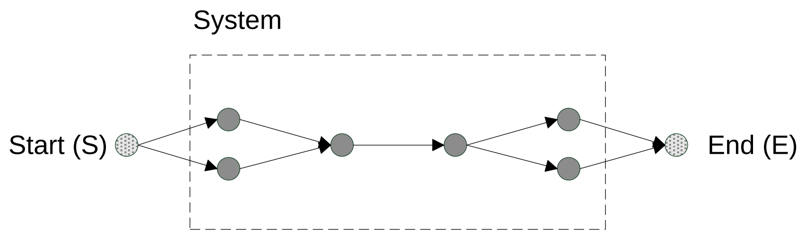


Figure 4.1: *Six-component system representing signal path from virtual component S to E*

The components and their behavior are assumed to be described by the models presented in Figure 4.2. The system components (middle) can be in three states *ready*, *blind* or *false*, while component S (left) is either *silent* or *system demanding*. Component E (right) is static, i.e. does not change state.

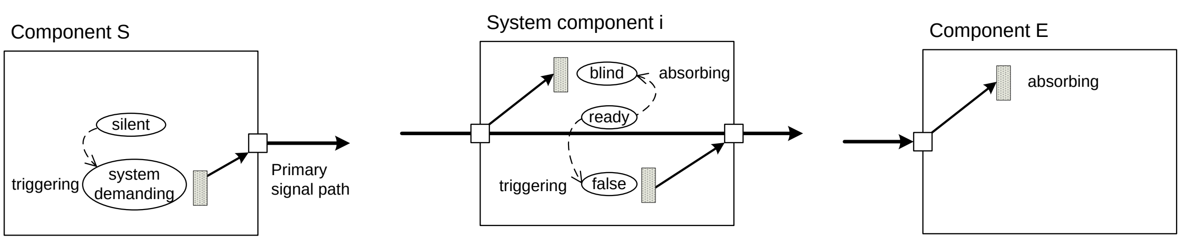


Figure 4.2: *Basic component models. Component S (left), system component i (middle) and component E (right)*

A signal is triggered upon

- component S going *system demanding* or
- system component i going *false*.

A signal is absorbed by

- system component i being *blind* or
- component E.

A signal is transmitted by a system component being *ready*.

In terms of the objectives of this thesis, the conceptual system in Figure 4.1 refers to the MPS. A component going *false* corresponds to the triggering of a *false beam dump* request. Component S represents the machine (Chapter 2). The component going *system demanding* corresponds to the triggering of an *emergency beam dump* request. With the initial states being *blind* and *silent*, over time, the components change their states independently. The resulting state configurations can be assigned to the global scenarios of interest, including *missed emergency beam dump* and *false beam dump* (Chapter 2). The observation of the system state over a certain time period, starting with the initial state, yields a measure for the probability of the different scenarios to occur.

The modelling concept is further explained in the following sections.

4.1.1 Basic Component Models

The core of the component models (Fig. 4.2) is given by the state diagrams illustrated in Figure 4.3. They reflect a continuous-time, discrete-state Markov model describing the behaviour of a component with one (left) and two (right) failure modes [46].

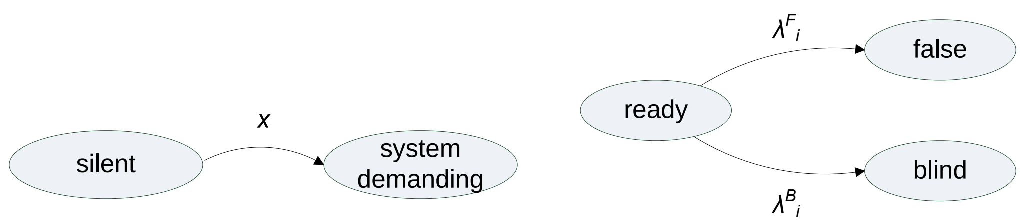


Figure 4.3: *State diagram of virtual component S (left) and system component i (right)*

A system component i (i.e. MPS component) can take three states (Fig. 4.3, right). State *ready* refers to the component being able to fulfill its function, i.e. incoming beam dump request signals are treated and transmitted accordingly (Tab. 4.1). State *blind* reflects one out of two component failure modes. It refers to undetected failures prohibiting the component function. Incoming signals are ignored, i.e. not transmitted (corresponding to an interruption of the signal path). State *false* refers to failures that are covered by fail-safe measures and thus upon occurrence trigger a (false) beam dump request.

Starting in the initial state *ready*, within a given time period, the component can go *false* or *blind* (or stay *ready*) . The two failure states are absorbing, i.e. the component remains in the state once taken. This means that, within a given time period, the

Table 4.1: *States of a system component*

State	Description	Consequence
ready	Full functionality	Incoming signal transmitted
blind	Undetected failure	Incoming signal absorbed
false	Failure covered by fail-safe measures	Signal triggered

component can only fail in one way, either *blind* or *false*.

The time to transition to either the *blind* or the *false* state, i.e. the time to failure, is assumed to be exponentially distributed with constant failure rates λ_i^B and λ_i^F, where λ_i^B, $\lambda_i^F > 0$ (Fig. 4.3). The exponential distribution function is commonly used in the field of reliability and risk analysis [43, 46, 47]. The assumption of constant rates is further discussed in Section 4.3.1.2.

Exponentially distributed state transition is likewise assumed for component S (i.e. Machine), defined by system demand rate x, where $x > 0$ (Fig. 4.2, left). The states are described in Table 4.2. State *silent* refers to the machine running under nominal conditions. State *system demanding* relates to machine failures leading to non-nominal machine or beam conditions which require the MPS to trigger a beam dump request and perform an *emergency beam dump*. Accordingly, component S going *system demanding* entails the triggering of a signal.

Table 4.2: *States of virtual component S (Start)*

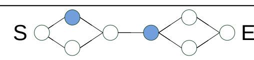
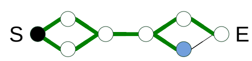
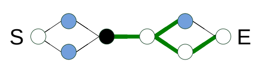
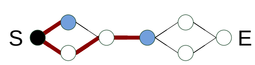
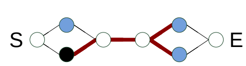
State	Description	Consequence
silent	Nominal machine and beam condition	No signal triggered
system demanding	Non-nominal machine or beam condition	Signal triggered

The determination of the component failure rates and the system demand rate is discussed in Section 4.1.3.

4.1.2 System Model

The system model is composed of the component models as the basic modules. The resulting state configurations define the global scenarios. The interrelation between state configuration and global scenario is exemplified in Table 4.3. Scenario I denotes nominal operation. It differs from the remaining scenarios through the absence of a signal triggering event. Scenarios II and IV are characterised by the triggering of a signal by component S, which, depending on the condition of the components along the (downstream) signal path, either results in Scenario II (*emergency beam dump*) or IV (*missed emergency beam dump*). Scenarios III and V are characterised by the triggering of a signal by a system component. Likewise, the triggering results in Scenario III (*false beam dump*) in case of open downstream signal path and Scenario V (*missed false beam dump*) in case of interrupted signal path.

Table 4.3: *Interrelation between state configurations and global scenarios. Black: trigger, blue: absorber*

Scen.	**Description**	**State configuration**	**Signal trigger**	**Signal path condition (downstream)**
I	Nominal operation (→ completed mission)		None	Component S *silent* and system components either *blind* or *ready*
II	Emergency beam dump		Component S going *system demanding*	Open
III	False beam dump		System component going *false*	Open
IV	Missed emergency beam dump		Component S going *system demanding*	Interrupted due to *blind* component
V	Missed false beam dump		System component going *false*	Interrupted due to *blind* components

The scenario probabilities are defined by the probability of the related state configurations to occur during a given time period (observation time). The probabilities correspond to the limit of the relative occurrence frequencies over an infinite number of independent observations.

Figure 4.4 further illustrates the concept by means of a single fictitious observation resulting in a *missed emergency beam dump* scenario.

The model observation (Fig. 4.4) includes the following events:

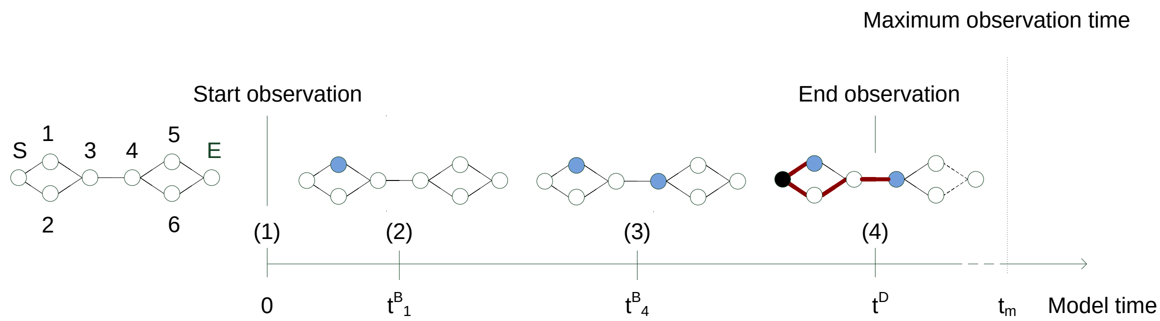


Figure 4.4: *Model observation in a time scale*

1. $t = 0$: All components in their initial state (*silent* or *ready*)

2. $t = t_1^B$: Component 1 going *blind*

3. $t = t_4^B$: Component 4 going *blind*

4. $t = t^D$: Component S going *system demanding*. The resulting scenario is *missed emergency beam dump* caused by *blind* component 4. Model observation is stopped.

The components being in their initial state at $t = 0$ applies to all observations. It is a basic assumption relating to the system initially being as good as new. Observation stop upon signal triggering likewise is a main model characteristic. The triggering of a signal instantaneously results in a successful (II, III) or missed (IV, V) beam dump scenario, depending on the downstream signal path condition (Tab. 4.3). Possible subsequent changes in the state configuration are irrelevant and not considered. The consequential observation stop at the time of the triggering implies no time delay for signal transmission. The model assumptions are further discussed in Section 4.3.

Time t_m (Fig. 4.4) is a model parameter defining the maximum observation time. If until time $t = t_m$ no signal is triggered, corresponding to state configuration I (*nominal operation*, Tab. 4.3), the resulting scenario is referred to as *completed mission*. Model parameter t_m is further discussed in Sections 4.1.3.3 and 4.2.4.

The observation of the components causing *false beam dump* scenario (III, through signal triggering) and missed beam dump scenarios (IV and V, through signal absorbing) allows for the determination of the relative importance of the different components. Importance refers to the relative contribution of the component to the related scenario. In case of the importance regarding signal absorbing (missed beam dump scenarios), it is to be noted that redundancies are taken into account. Using the example of Figure 4.4, this means that component 1 being *blind* in this case does not contribute to the importance with regard to signal absorbing, while *blind* component 4 does.

4.1.3 Input Parameters

The model input parameters comprise

- component failure rates λ, reflecting component (system) reliability,
- system demand rate x of component S, reflecting machine reliability and
- observation time t_m, reflecting the nominal operational cycle.

Their determination is discussed in the following sections.

4.1.3.1 Component Failure Rates

The failure rates λ_i^F and λ_i^B are determined individually for each component i. The ideal data source are statistics based on operational experience. It requires extensive component operating time under comparable operating conditions (e.g. in terms of temperature, radiation, current, voltage etc.). Furthermore, accordance with regard to inherent fail-safe measures or software is required. These limitations are to be considered when using operational data, be it based on the system under consideration or other facilities.

Most often, operational data is not available, especially if the system under consideration and its components are prototypes. In that case, the failure rates have to be estimated, e.g. based on data of the devices the component is composed of. The data for electronic devices may be drawn from reliability handbooks or manufacturer data sheets. Such data allows for a comprehensive reliability analysis using established methods such as FTA. In accordance with the model, the FTA related top events are *component fails in blind mode* and *component fails in false mode*.

In case of missing data, the failure rates rely on expert judgement by designers and users of the components. This is mostly based on a mixture of personal experience and common sense.

Key aspects to be considered in the deduction of the failure rate are discussed in Section 4.3.1.

4.1.3.2 System Demand Rate

The above considerations on the deduction of the component failure rates in principle applies for the system demand rate x likewise. If operational data of the machine or a comparable facility is available, it is the obvious source to deduce a figure for the rate. It may be supported by a comprehensive reliability analysis of the machine, although depending on the size and complexity of the machine this is a costly task. Besides, it is questionable whether a single parameter is worth this effort. An estimate based on operational experience and expert judgments is considered to be adequate to begin with. The uncertainty related to the single system demand rate x is manageable and easily addressed by a sensitivity analysis based on parameter variation.

Additionally to the rate, system demand involves the pattern specifying which parts of the system are affected in case of non-nominal conditions in the machine. In terms of the system model, the system demand pattern is reflected by the interconnections between virtual component S and the system components. In case of the six-component system presented in Figure 4.1 for example, component S could alternatively be connected to one component only instead of two, defeating the redundancy at the beginning of the signal path. On that account, the system demand pattern includes a weighting of different system parts.

The specification of the system demand pattern requires an analysis of the machine in terms of the location and impact of its failing. Again, it may be based on operational experience. In case of missing operational experience, it is supported by calculations and simulations of failures and the related physical phenomena.

4.1.3.3 Observation Time

Observation time t_m reflects an operational parameter (i.e. length of operational cycle) and thus is straightforward to be determined. As pointed out above for the system demand rate, it is easily varied in terms a sensitivity analysis.

4.1.4 Suppression Effect

The model features a characteristic which subsequently is referred to as *suppression*. It is briefly discussed here for later reference.

The suppression effect bases upon interdependencies of the failures within a single component and across components. It is classified into two types based on the coverage of the interdependency and the underlying cause (Tab. 4.4).

Table 4.4: *Types of suppression*

Type	Interdependency	Cause
I	Component-internal	Absorbing failure states
II	Across components	Observation stop upon signal triggering

The suppression entails that the state transitions actually coming into effect are not exponentially distributed with the related rates λ or x. In fact, the individual state transitions occur less frequently than they would if considered independently. In case of the component-internal suppression (Type I), the reason for the suppression lies in the two absorbing failure states. Within a single observation, a component i going *false* at time t_i^F cannot go *blind* at $t_i^B > t_i^F$ (or vice versa). On that account, the transition to a failure state is *suppressed* by a previous transition to the other failure state. This characteristic reflects the real system. If a component fails in the *false* (*blind*) mode, a (potential) later failure in the *blind* (*false*) mode is irrelevant.

The suppression across components (Type II) is caused by the observation stop upon a signal triggering. A triggered signal immediately results in a beam dump scenario (successful or missed). Potential later state transitions are irrelevant and not considered. Figure 4.5 illustrates the described interdependencies for suppression Type I (left) and II (right). It is to be noted that the interdependency between signal triggering failures (states *system demanding* and *false*) and *blind* failures is directed. Since a *blind* failure does not cause an observation stop, it does not influence the failures of other components. *Blind* failures therefore contribute to suppression of Type I only.

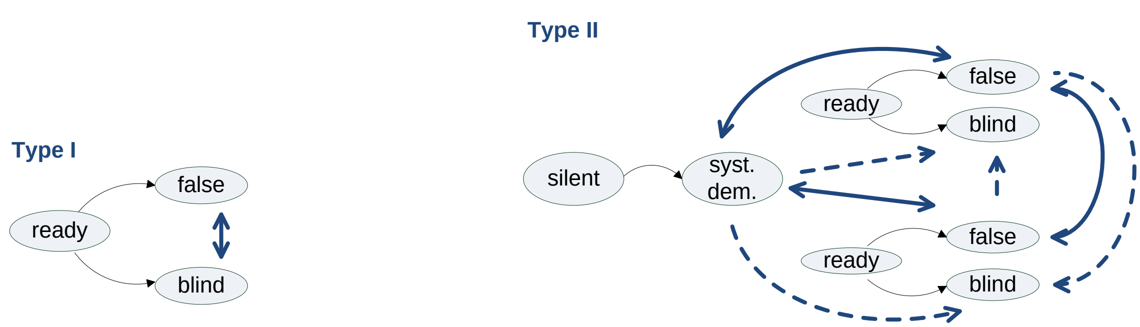


Figure 4.5: *Suppression effect: Interdependencies between failures within a component (left) and across components (right)*

4.2 MPS Model

The MPS model is illustrated in Figure 4.6. It represents one BLMS branch (out of eight) and the related BIS components (e.g. CIBUS and BICs) according to the structure introduced in Chapter 3 (Fig. 3.3 and 3.6). The signal path is exemplified using six out of the 468 monitors of the branch, starting in component S (bottom right) and ending in component E (top left).

Figure 4.7 shows the entire MPS model, consisting of eight branches (as illustrated in Figure 4.6). The branches and their signal path are self-contained, i.e. they are linked through component E only. Component E represents the interface to the LBDS (Chapter 3). Note that component S is connected to six monitors of one of the branches only, reflecting the underlying system demand pattern (Section 4.2.3).

The model includes eight types (classes) of system components. Table 4.5 lists the classes and the total number of components. They correspond to the number of system components included in a branch (Fig. 4.6) times eight, summing up to a total of 4768 components. It is to be noted that Figure 4.6 only includes the BICs (and CIBUSs) directly linked to the BLMS branch. The model includes the eight additional BICs per beam with no input from the BLMS (Fig.3.3), which makes up for the discrepancy between Figure 4.6 and Table 4.5.

Table 4.5: *Number of components included in the MPS model*

i	Component class *i*	Number n_i
1	Monitor	3744
2	FEE	624
3	BEE	312
4	CC	24
5	VME	24
6	CIBUS	8
7	BIC1	16
8	BIC2	16
	Total	4768

The models of the different component classes are presented in the following section.

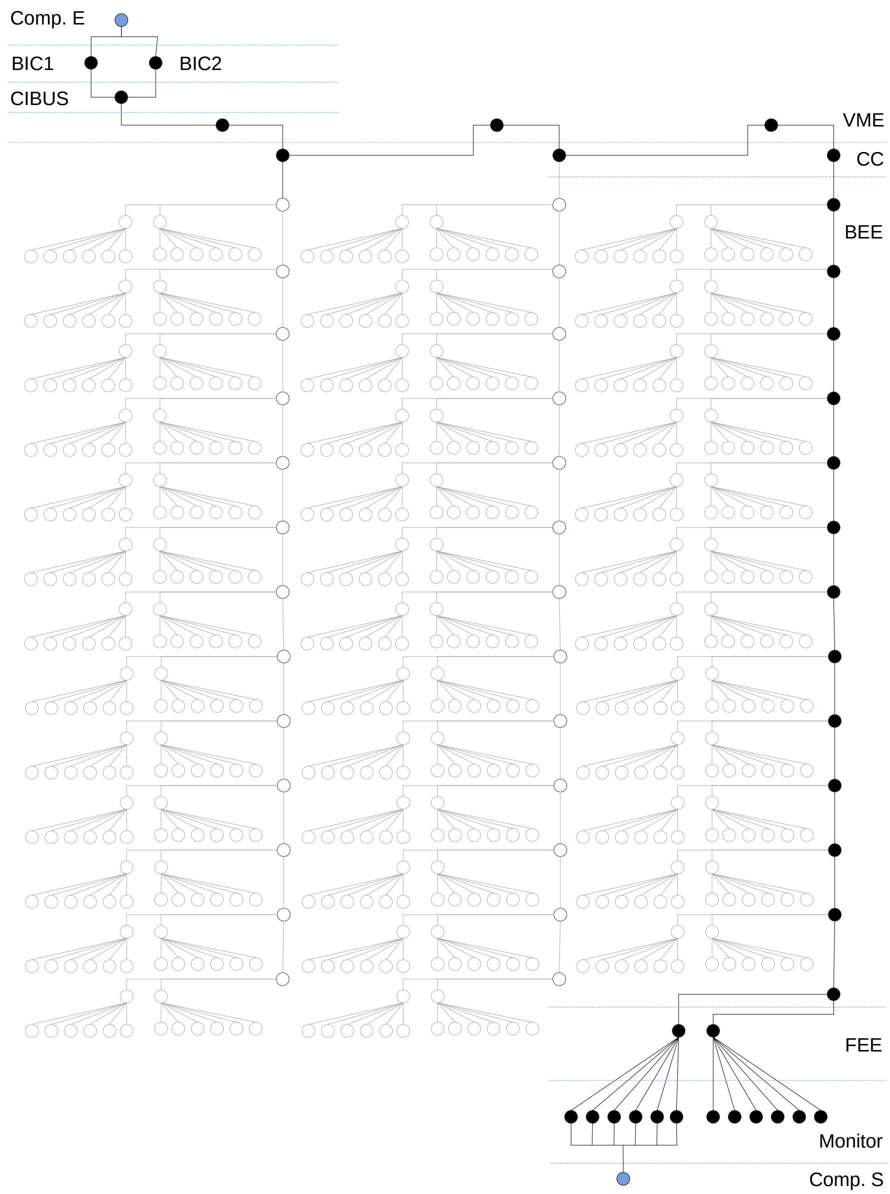


Figure 4.6: *Model structure of one BLMS branch including the related BIS components (CIBUS and BICs)*

4.2.1 Component Models

Table 4.6 presents the models of the different component classes. The models correspond to the basic system component model described above (Fig.4.2, middle), except for an individual number of in- and outputs according to the MPS model structure (Fig. 4.6). Next to the in- and outputs, the related neighbouring components are pointed out. Numbers in brackets (in models FEE and BEE) denote multiple inputs from com-

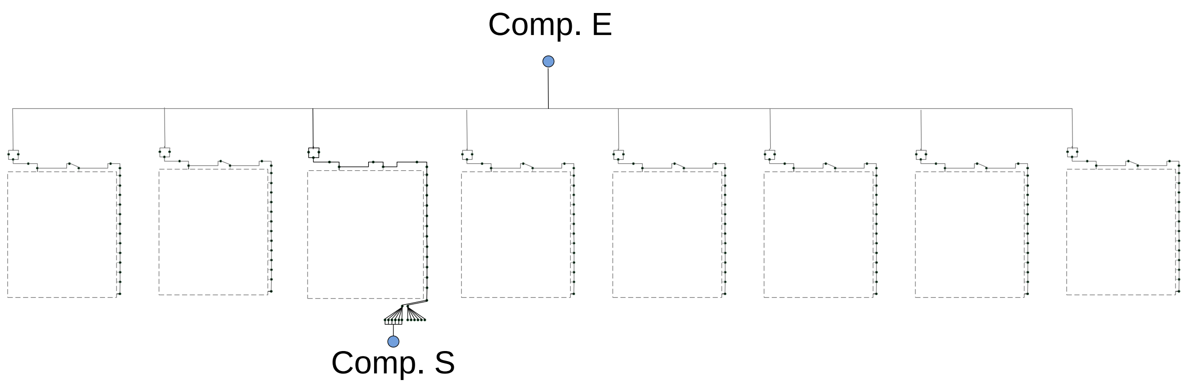
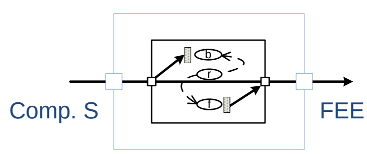
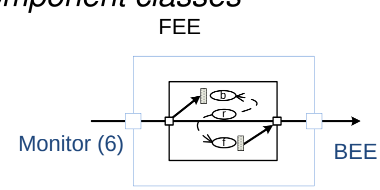
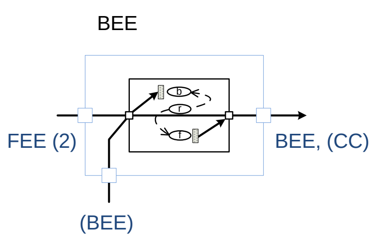
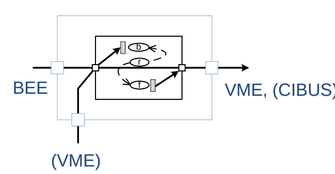
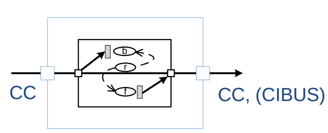
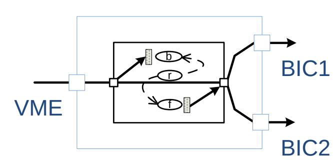
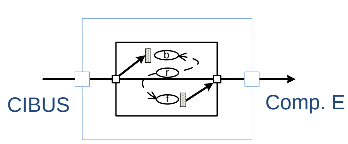
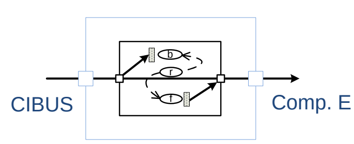


Figure 4.7: *Overall MPS model structure*

ponents of the same class. Names in brackets (in models BEE, CC, VME) indicate the special conditions for the first and last component in a daisy-chain. The first (downstream) BEE does not get input from another BEE, while the last one is connected to the CC instead of a subsequent BEE (Fig. 4.6). The same consideration applies to the CC and VME models, according to their alternating daisy-chain.

Table 4.6: *Models of the different component classes*

Monitor		FEE	
BEE		CC	
VME		CIBUS	
BIC1		BIC2	

The models BIC1 and BIC2 seem considerably simplified in comparison to the real

BICs, which are connected with neighbouring BICs in two loops A and B (Chapter 3). The simplification relies on a specific characteristic of the daisy-chain between the BICs. It is based on the assumption that the failures of the BICs (defined by the basic component model) only affect beam dump request signals coming in from the CIBUS, i.e. do not interfere with the transmission of beam dump request signals across the BICs. The assumption is discussed in more detail in Section 4.3.1.2.

4.2.2 Component Failure Rates

A comprehensive reliability analysis of the components (as suggested in Section 4.1.3.1) is beyond the scope of this work. Instead, a set of component failure rates has been defined based on previous studies on MPS reliability (Tab. 4.7). This set is subsequently referred to as set *STANDARD*.

Table 4.7: *Component failure rates (STANDARD)*

i	Component *i*	Rate λ_i^F [h^{-1}]	Rate λ_i^B [h^{-1}]
1	Monitor	1E-7	1E-7
2	FEE	1E-6	1E-8
3	BEE	1E-8	1E-9
4	CC	1E-8	1E-9
5	VME	1E-5	1E-8
6	CIBUS	1E-6	1E-13
7	BIC1	1E-5	1E-13
8	BIC2	1E-5	1E-13

The rates of the BLMS components represent orders of magnitudes based on the study performed by Guaglio [20]. His work includes fault trees of a BLMS branch with top events '*Damage Risk*' (relating to missed emergency beam dump) and '*False Alarm*' (relating to false beam dump). As a first approximation and following the rare event approach described by Guaglio, the (constant) failure rates of the basic events in the fault trees were summed up to the level corresponding to the model components. The resulting orders of magnitude provide the presented rates for the BLMS components. The rates for the BIS components base upon expert judgement taking into account the results of a FMECA analysis performed by Todd [1].

4.2.3 System Demand

In the MPS model, the system demand rate reflects the occurrence of non-nominal beam losses (Chapter 2). The system demand rate, i.e. the rate of beam loss events, is set to $x = 1E-2\ h^{-1}$. The figure is based on expert judgment assuming a mean time to beam loss of 100 hours. It is a rough estimate representing the scenario of about 10% of the missions encountering a (non-nominal) beam loss event. The figure for the system demand rate and the component failure rates presented above constitute the parameter set *STANDARD A* (Chapter 7).

The system demand pattern reflects a simplified beam loss pattern with the following characteristics:

- Limited to one (out of eight) sectors (Chapter 2)
- Limited to one magnet at a time
- Simultaneously triggering all monitors of the respective magnet

The system demand pattern is illustrated in Figure 4.8. The beam losses are assumed to be uniformly distributed in the sector. Component S going *system demanding* (i.e. a beam loss event) randomly affects the six monitors assigned to one of the 78 FEEs of the related BLMS branch (i.e. the monitors of one of the 78 magnets covered by beam loss monitors). Given this pattern, the monitors thus include a 1-out-of-6 redundancy.

4.2.4 Observation Time

Corresponding to the nominal LHC operational cycle (Chapter 2), observation time is set to 12 hours ($t_m = 12$). It includes the ramping-down and pre-injection phases after a beam dump, where no beam is present in the machine. However, during ramping-down (and pre-injection), the magnets still store remarkable energy. The MPS (i.e. the QPS in particular) therefore has to be operational during this phase.

Besides component failure rates, system demand rate and observation time, the model structure (e.g. in terms of the amount of components or signal path) could be considered as an additional model parameter, allowing for the comparison of different system designs. Within the scope of this work, the structure is assumed static according to the model presented above. Fundamental changes in the model structure require time-consuming adaptations in the model implementation (Chapters 5 and 6). However, a

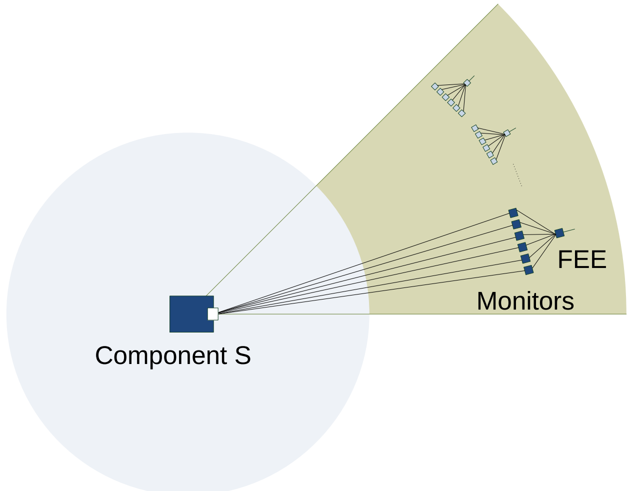


Figure 4.8: *System demand pattern in the MPS model*

first attempt towards a simplified handling of the modelling approach in terms of structure variation has been taken. It is presented in Chapter 11.

4.3 Discussion

The presented modeling approach for the reliability analysis of a protection system distinguishes itself by a range of characteristics. In particular, it includes

- two component failure modes with different consequences (described by a simple Markov model),
- system demand and
- the quantification of five system scenarios within the same model.

The system model structure, representing the signal path from sensors (monitors) to actuators (in the case of this model the interface to the subsequent subsystem), covers redundancy on system level, i.e. across components. Redundancy within components is included by the component failure rates. Fail-safe measures are taken into account by one of the component failure modes (*false* mode).

The component models (Markov models) describe the system component reliability. The system demand reflects the reliability of the machine. System scenario *completed mission* relates to LHC reliability. Scenario *missed emergency beam dump* provides indications in terms of machine safety, while scenario *false beam dump* is relevant regarding beam availability (Chapter 2).

The modeling approach hence meets the requirements defined above (Chapter 1). For the tracing of the related analysis and in particular the assessment of its results and uncertainties, a thorough understanding of the underlying assumptions and simplifications is essential. The following sections address the basic model assumptions and the main simplifications with regard to the real MPS.

4.3.1 General Model Assumptions

The modeling approach includes a range of assumptions regarding the component boundaries, the component model and the triggering and transmission of signals.

4.3.1.1 Component Boundaries

The modeling approach relies on the component models as the basic modules. For the purpose of a consistent determination of the component models, a guideline regarding the component boundaries is required. The definition of the (virtual) boundaries and the related assumptions are discussed by means of the illustration given in Figure 4.9.

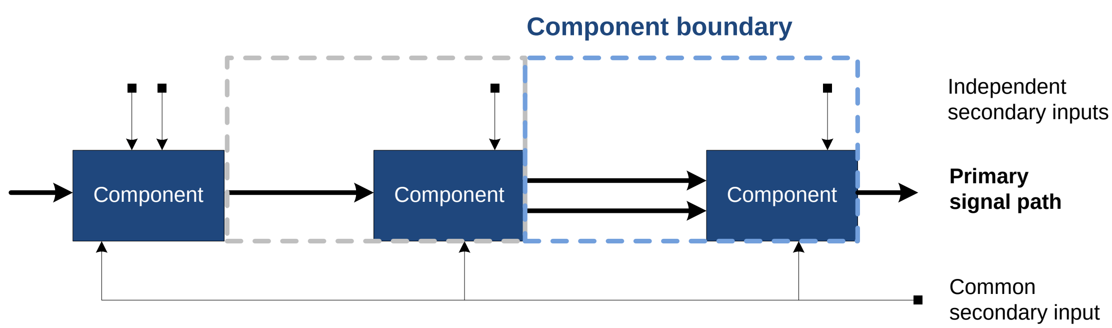


Figure 4.9: *Definition of component boundaries*

The system model (representing the primary signal path of a system) is built up by the component models which are connected by fault-free lines. Line characteristics that are relevant in terms of system reliability therefore have to be included in the component model. For this purpose, the component boundary is drawn at the output of the neighbouring upstream component (Fig. 4.9). This means that the interconnection line between two components is assigned to the downstream component by definition. The characteristics related to the line, e.g. redundancy or continuous testing in the optical connection between FEEs and BEE (Section 3.3), are to be taken into account in the analysis of the downstream component.

The analysis of a component in terms of reliability bases upon the definition of the component primary function. The fulfilling of the function implies a fault-free output. In order to simplify matters and in terms of a conservative approach, in case of redundant outputs each is postulated to be fault-free. According to the definition of the system boundaries, a component's output represents the neighbouring component's input. The component reliability analysis therefore bases upon fault-free inputs.

Besides the inputs related to the primary signal path, a component may depend on secondary inputs such as power or status signals. The definition of the boundaries with regard to secondary inputs arises from one of the most basic model assumptions, i.e. the independency of the components in terms of their failures. The analysis of a component therefore takes into account secondary inputs if they are individually assignable to the component, i.e. secondary inputs whose failure affects one component only. As an example, individual component power supply, if applicable, is to be included in the component analysis.

Common secondary inputs affecting more than one component are not included (Fig. 4.9). In the component analysis, they are postulated to be fault-free. However, the inclusion of common secondary inputs, representing a potential source of common cause failures, is introduced as a model extension in Chapter 9.

The considerations on the component boundaries with regard to secondary inputs apply to fail-safe measures accordingly. The fail-safe measures taken into account in the component analysis are to be inherent to the component and independent respectively. Extra monitoring systems which cover several components, triggering false beam dump requests in case of component failures, are excluded. They have to be treated in the context of common cause failures. It is to be noted that *false* failures are assigned to the component that triggers the false beam dump request. It does not necessarily correspond to the location of the failure occurrence.

4.3.1.2 Component Model

The component boundaries (Section 4.3.1.1) provide the frame for individual analyses of the components. The analysis results in a component model specified by

- two constant failure rates related to absorbing failure states,

- an individual number of in- and outputs and
- a distinct internal wiring.

Constant failure rates and absorbing states

The component model bases upon constant failure rates defining exponentially distributed times to failure. Exponentially distributed failures are a common assumption in reliability analysis [43]. Constant failure rates relate to the intermediate phase in the component life cycle (reflected by the well-known bathtub curve), which is characterized by the memoryless property [43]. In the present case, constant failure rates are considered to be a reasonable approach given the limited time of model observation and the general uncertainty inherent in the failure rate figures. A possible degradation of the components over a larger time period (i.e. over a multitude of missions) is easily addressed by sensitivity analysis based on increased failure rates. As for the distribution of the times to failure, the model would allow for the use of alternative distribution functions (e.g. Weibull distribution).

The Markov model (state diagram) describing the component behaviour is based on absorbing failure states. It implies that a component during a certain time period can fail in one way only (either *blind* or *false*), since the state of failure is permanent (up to repair or replacement). The model excludes the possible scenario that a (partially) *blind* component additionally encounters a *false* failure. Since the latter would trigger an early beam dump, the exclusion of the scenario represents a conservative assumption. A component going *false* in the first place immediately results in a successful or missed false beam dump scenario. The possible scenario of a component failing in the *blind* mode after having triggered a false beam dump request is irrelevant. The same consideration applies to the exclusion of transient failures, i.e. possible non-permanent failures (e.g. neutron-induced single events). With regard to transient *blind* failures, the model is conservative in assuming them to be permanent. The impact of transient *false* failures corresponds to that of permanent *false* failures, triggering a false beam dump request (if coming into effect at all). They are thus covered by the *false* failure mode of the model and may be included in the related failure rate if necessary.

In- and outputs and wiring

The deduction of the component models includes the definition of the component-internal signal path between in-and output (subsequently referred to as internal *wiring*). It is important to realize that different wiring implies different component behaviour. In particular, the wiring defines whether or not a particular input is affected by component failures. The relevance of the wiring is exemplified on the basis of the BIC component model (Fig. 4.10).

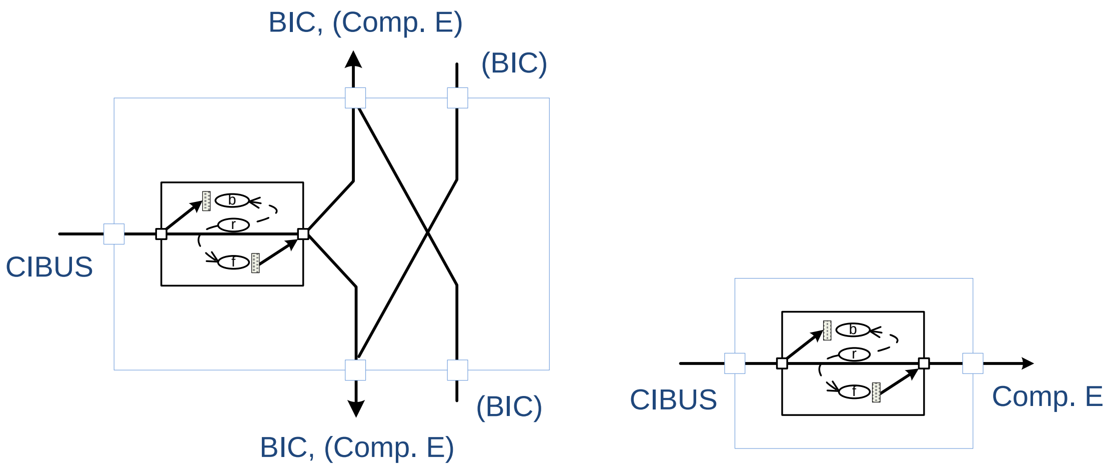


Figure 4.10: *Component wiring: 'realistic' (left) and abstract (right) BIC component model*

The 'realistic' wiring (Fig. 4.10, left) is based on the assumption that a *blind* failure of the BIC affects the signals coming from the CIBUS only. Input signals from neighbouring BICs are not affected by failures. The daisy-chain of the BICs thus is assumed to be fault-free. Accordingly, the wiring of the component model on the left is corresponding to the simplified wiring depicted in the model on the right. It directly reflects the assumption that a signal once reaching a BIC output cannot be interfered with, thus reaching the end of the signal path (i.e. component E) by definition.

In view of the BIC design and based on expert judgement [48], the described assumption regarding the BIC component wiring is considered to be a reasonable first approach. Profound knowledge on the component model wiring requires a comprehensive analysis. Depending on the complexity of the real component function and wiring, its modelling might ask for the use of several Markov models (state diagrams) instead of just one.

4.3.1.3 Signal Triggering and Transmission

With regard to signal triggering and transmission, the following assumptions are included:

Single signal triggering events The model bases upon the assumption of only one signal triggering event at a time, i.e. each state configuration (Tab. 4.3) includes at most one signal triggering event (either component S going *system demanding* or a system component going *false*). The simultaneous occurrence of several triggering events, e.g. two components going *false* at the same instant of time, is excluded. This means that a part of the theoretically possible system configurations and the related possibilities is not taken into account. Considering the order of magnitude of the component failure rates however, those probabilities are considered to be negligible. It is to be noted that this assumption relates to simultaneous *independent* failures, which are not to be confused with common cause failures. The latter base upon a *single* event affecting several components (Chapter 9) and as such do not relate to the presented assumption.

Signal related to failing The signals in the model relate to failures, i.e. they represent active transmitters of beam dump requests. This is contradictory to the real MPS design, where beam dump requests are represented by *missing* signals (Chapter 3). The model reflects the complementary behaviour of the MPS and its components. The complementary character is exemplified by means of the *blind* failures. In the model, blind failures are defined to result in signal absorption. In the real system by contrast, *blind* failures result in an upkeep of user permit signals even in case of a dropp out of incoming signals. The described model assumption implies a significant deviation from the MPS in terms of emergency beam dump requests, namely the anticipation of the threshold comparison from the BEEs to component S. Component S going *system demanding* strictly relates to non-nominal beam losses, i.e. to beam losses above threshold. This restriction has to be taken into account when addressing system demand rate and pattern.

Instantaneous signal transmission The signal transmission is assumed to be instantaneous, i.e. the model does not include a time delay for the transmission. The MPS is designed to perform an emergency beam dump within a few hundred microseconds (Section 3.1.3). In conjunction with the order of magnitude of rates λ^B, the exclusion of time delay in the signal transmission seems justifiable. The model does not include the theoretical scenario of a presumable successful beam dump turning into a missed beam dump due to *blind* failures occurring (downstream) during dump request signal transmission. The probabilities of such a scenario are considered to be negligible.

4.3.2 Specific Simplifications with regard to MPS

The following sections address the most significant simplifications and limitations of the MPS model with regard to the real system.

4.3.2.1 Model Coverage

The current model includes the BLMS and the BIS, i.e. the largest monitor system detecting fast beam losses and the core subsystem linking all monitor systems to the LBDS (Chapter 3). The model covers the signal path from the monitors to the interface of the LBDS. This coverage is reasonable in view of the system demand being a key feature of the model, directly linked to the monitors. The LBDS itself is not included (or assumed to be fault-free respectively). Since it is involved in each beam dump, the LBDS is crucial for MPS reliability and machine safety. Its inclusion therefore is an obvious starting point with regard to a model extension.

The model structure of both the BLMS and BIS represents a simplification based on average numbers and numbers from earlier design phases. The BIS model includes 16 BICs per beam instead of the 17 included in the current MPS. With regard to the CIBUS, it only covers the eight CIBUS that are related to the BLMS, while the actual MPS incorporates about 150 CIBUs. The limitation of the CIBUS included in the model supposedly results in a slight underestimation of the *false beam dump* scenario but does not affect the safety relevant missed beam dump scenarios.

The BLMS model assumes an average number of 3 VME crates per branch, 13 BEEs per crate and 6 monitors per FEE. The actual BLMS includes 4 VME crates at IP7 (Section 3.3), while a crate contains up to 16 BEEs each with up to 8 input channels (Section 3.3).

4.3.2.2 Component Classes

The model bases upon 8 component classes, each with a dedicated *false* and *blind* failure rate. Accordingly, the components of a class are assumed to be identical, particularly in terms of reliability. The assumption in particular implies that the components meet comparable environmental and operating conditions, e.g. in terms of temperature or radiation. It represents a noticeable simplification especially regarding the *monitor* class. The model assumes 3744 identical monitors, while the real MPS involves two different types of beam loss monitors (ICs and SEMs, Section 3.3.1) in different environmental conditions (especially in terms of radiation).

4.3.2.3 Component Failure Modes

The model includes two failure modes *false* and *blind*. Previous reliability studies on the MPS include failure mode *warning* as a third mode [1, 20]. It relates to component failures that are considered not to compromise MPS functioning but that are classified significant enough to entail the generation of a warning upon failure as a hint for maintenance action after the mission (Section 3.6.2).

These failures are relevant in terms of the maintenance in between missions, which is not included in the model. They are not directly safety relevant and do not contribute to any of the five system scenarios included in the model. They therefore are not taken into account explicitly. However, if these failures concern components in the primary signal path, they should be considered and included in the existing failure mode *blind* through adjusted failure rates.

4.3.2.4 Signal Path

The model bases upon single fault-free interconnection of two components. As mentioned above, this implies that possible characteristics of the lines which are relevant in terms of reliability (such as redundancy in case of the interconnection between the FEEs and the BEEs) are included in the reliability analysis of the (downstream) components.

The model signal path from the monitors to the BEEs represents a simplification regarding the real system. The threshold comparison in the BEEs is based on individual channels from the monitors to the BEEs. In the model, the lines (channels) of six monitors are merged into one by the FEE. With regard to the signal path, they therefore conjointly depend on the state of the FEE, while in the real MPS a failure in the FEE may affect only parts of the related monitor channels.

The model assumes one and the same signal path for all signals. It thereby does not include the channel masking feature of the MPS (Section 3.3.3).

4.3.2.5 System Demand

The system demand model assumes non-nominal beam losses that are exponentially distributed in time. Furthermore, it reflects the imaginary scenario that each beam loss event is locally limited to the area of a single magnet and that it is simultaneously detected by the related six monitors. The model in this way particularly excludes the scenario of beam losses occurring in parts of the machine that are not covered by monitors.

4.3.2.6 Model Observation Conditions

Model observation starts with all components in their initial state, i.e. *silent* or *ready*. This condition relates to a MPS as-good-as-new at the beginning of the mission, i.e. to the previous detection and repair or replacement of failed components.

The observation is stopped upon a trigger event or after 12 hours of (model) observation time. The model does not explicitly take into account end-of-mission beam dumps, i.e. the (planned) beam dumps performed after twelve hours of a nominal mission (Chapter 2). The end-of-mission dumps base involve the seventeenth BICs, which are not included in the model (Section 4.3.2.1). As already pointed out for the LBDS, the model in this way assumes the triggering of the end-of-mission dumps and the related BICs to be fault-free (at time $t = 12h$).

Chapter 5

Monte Carlo Simulation

The MPS model introduced in the previous chapter is implemented twice, first based on Monte Carlo simulation and then based on an analytical description. This chapter presents the Monte Carlo approach and the related MPS model implementation (Fig. 5.1). The Monte Carlo approach is based on state transitions being defined by random numbers.

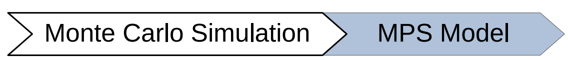


Figure 5.1: *Monte Carlo approach: First implementation of MPS model*

5.1 Concept

The component model represents the basic module of the MPS model and the implementation of the Monte Carlo approach likewise. The concept of the approach is introduced by means of the component model and the six-component system presented above (Section 4.1, Fig. 4.1).

5.1.1 Component Model

The Monte Carlo approach is based on state transitions defined by random numbers. The time to transition T is a variate (random variable) following an exponential distribution, which is defined by the related component failure rate and system demand rate respectively (Fig. 5.2).

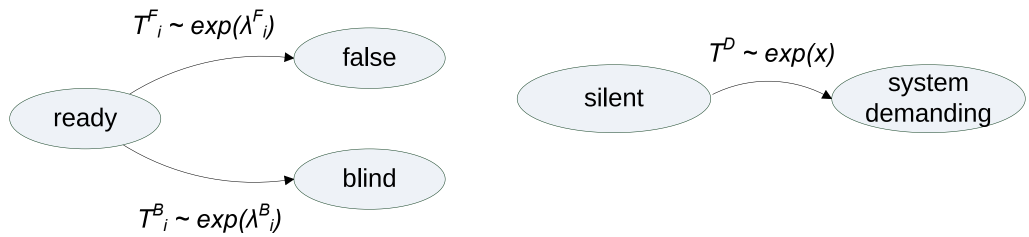


Figure 5.2: *Monte Carlo approach: State diagrams for system component i (left) and component S (right). T: Time to transition (variate)*

There are various methods for generating variates with nonuniform continuous distribution such as exponential distribution. The most common method applies inverse transform [47]. It is based on the inverse cumulative distribution function using uniform random numbers in interval [0,1].

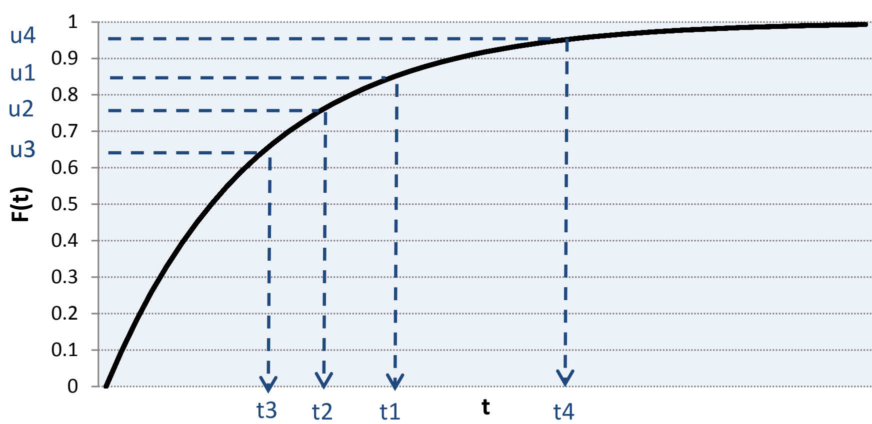


Figure 5.3: *Monte Carlo approach: Inverse Transform. Generating exponentially distributed variate T using uniform random numbers u*

.

The concept is illustrated in Figure 5.3. Given variate U with uniform distribution in interval [0,1], exponentially distributed variate T is defined by the inverse of the cumulative exponential distribution function $F(t) = 1 - e^{-\lambda t}$ with $\lambda > 0$, $t \geq 0$ as

$$T = -\frac{1}{\lambda} ln(1 - U). \tag{5.1}$$

Since 1-U and U are equally distributed, Equation 5.1 is equivalent to $T = -\frac{1}{\lambda} ln(U)$ [47]. The generation of uniform random numbers (u_i) nowadays relies on pseudo-random number generators based on recursion formulas [47].

In Excel for instance, the uniform random number generator is provided by function *RAND()*. According to Equation 5.1, an exponentially distributed variate T_i with rate λ is produced using formula $-\frac{1}{\lambda}ln(RAND())$.

5.1.2 Six-Component System

Figure 5.4 illustrates the simulation concept based on the Monte Carlo approach. It shows a fictitious example of a simulated mission resulting in Scenario *missed emergency dump* at $t = 8$.

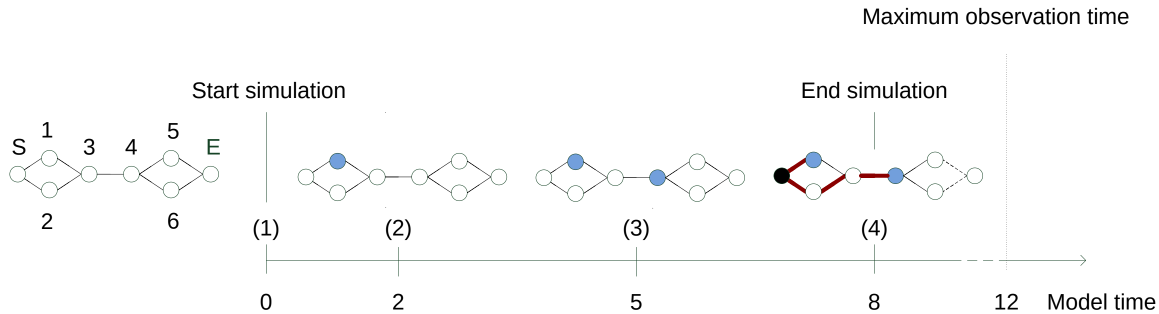


Figure 5.4: *Monte Carlo approach: Concept of simulation. Black: trigger, blue: absorber*

The simulation includes the following steps:

1. Random times to transition t assigned to each component (Tab. 5.1)
2. Transition with smallest t executed: Component 1 going *blind* (at $t = 2$)
3. Next transition executed: Component 4 going *blind* (at $t = 5$)
4. Next transition executed: Component S going *system demanding* (at $t = 8$). The resulting scenario is a missed emergency beam dump caused by *blind* component 4.

For simple systems, i.e. a small system with components connected in series, the Monte Carlo approach can easily be implemented using Excel. It requires the generation of a reasonable number of random number sets, each containing random times to transmission for each component. The sets are assigned to the different global scenarios by means of logical expressions including the times to transition of all components.

Table 5.1: *Monte Carlo approach: Fictitious random number set*

Component i (**S**)	t_i^F (**t^D**)	t_i^B
1	9	2
2	12	13
3	21	16
4	11	5
5	15	10
6	18	23
S	8	-

5.2 MPS Model Implementation

The Monte Carlo approach to the MPS model is implemented using the Java-based AnyLogic 5.5 software [49, 50]. The implementation involves the procedure described in Table 5.2. The listed steps represent the key tasks of the implementation. They are discussed in the following sections for the purpose of facilitating the tracing of the implementation and the underlying files.

Table 5.2: *Implementation: Procedure*

Task	**Step**	**File**
Create model	Create objects *BehaviourBox*	LHC_MPS_v2.alp
	Create component objects	ditto
	Connect component objects	ditto
	Set model input parameters	ditto
	Implement system demand	ditto
	Implement data logging	ditto
	Define stop conditions	ditto
	Build model	ditto
Run simulations	Create simulation environment	-
	Set simulation parameters	-
	Run simulations	LHC_MPS_v2.jar
Analyse data	Analyse data	statslog-<>-<>-<>.txt

5.2.1 Create Model

Building the MPS model in AnyLogic follows its modular conception. The custom object *BehaviourBox* reflects the basic component model (Section 4.1, Fig. 4.2). It defines the behaviour of the component objects, which in turn represent the modules that the system model is composed of. The entire model building is done within the same AnyLogic project and stored in a single file (in the present implementation file *LHC_MPS_v2.alp*).

5.2.1.1 Create Objects *BehaviourBox*

The implementation of the basic component model is introduced on the basis of the monitors. Figure 5.5 (left) shows the graphic representation of object *BehaviourBox-Monitor* reflecting the basic component model (of a monitor). It is derived from the generic AnyLogic class *ActiveObject* and composed of a custom state diagram *statechartStateObject* (right) and three object classes defined in AnyLogic Enterprise Library [51]. The used object classes are *Enter* (named *falseGenerator*, Fig. 5.5 left), *SelectOutput* (named *selectOutput*) and *Sink* (named *missedDump*).

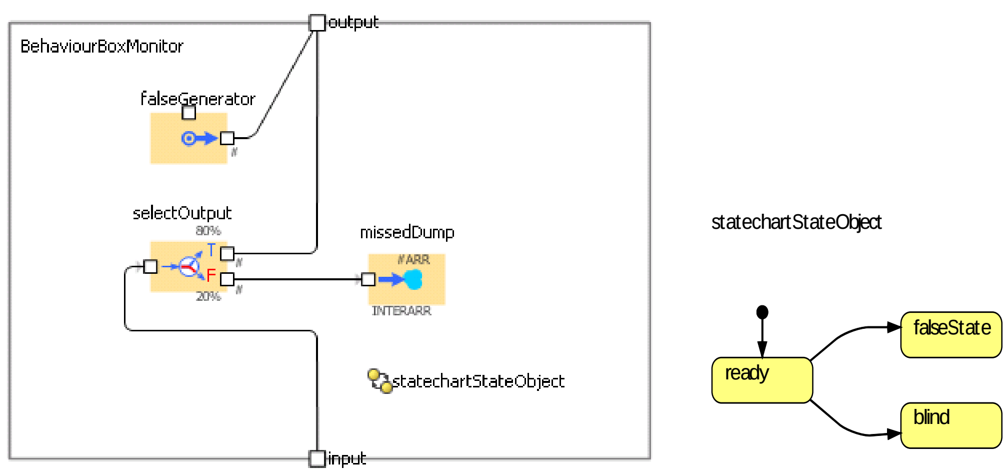


Figure 5.5: *Implementation: Object BehaviourBoxMonitor reflecting basic component model*

The functions of the objects and their interrelation with the state diagram *statechart-StateObject* is straight-forward according to the component model. They are shown in Table 5.3. For detailed information on the AnyLogic internal semantics the reader is referred to the AnyLogic Enterprise Library Reference Guide [51].

As long as object *BehaviourBoxMonitor* is in state *ready*, an incoming signal goes from

Table 5.3: *Implementation: Function of objects defining BehaviourBoxMonitor*

Name	**Type**	**Function**
falseGenerator	Enter	Releases signal upon state transition to *falseState*
selectOutput	SelectOutput	Switches output from *T* to *F* upon transition to *blind*
missedDump	Sink	Absorbs incoming signal

input via *selectOutput*(T) through to *output* (Fig. 5.5). In case of state *blind*, an incoming signal goes from *input* via *selectOutput*(F) to dead-end *missedDump*. In case of transition to *falseState*, a newly generated signal is released by *falseGenerator* and goes through to *output*. Signals are modelled using a message class of type *Entity* (named *Signal*) [51]. The state transitions are defined by parameters *lambdaFalseMonitor* and *lambdaBlindMonitor*. They correspond to the failure rates $\lambda^F_{Monitor}$ and $\lambda^B_{Monitor}$ (Section 5.2.1.4) and are set in the model development environment *Main* (Section 5.2.1.4).

The implementation of the basic component model, as described for the monitors, is generic and applies to the remaining components (FEE, BEE, CC, VME, CIBUS, BIC1, BIC2) likewise. The resulting objects BehaviourBox (Appendix B) only differ in the parameters defining the state transitions.

5.2.1.2 Create Component Objects

Figure 5.6 shows the graphic representation of object *Monitor*, reflecting the component model *Monitor* (Chapter 4, Tab. 4.6). Object *Monitor* is composed of *BehaviourBoxMonitor* and object *Enter* of the AnyLogic Enterprise Library. Object *Enter* represents the starting point of the system signal path (i.e. component S).

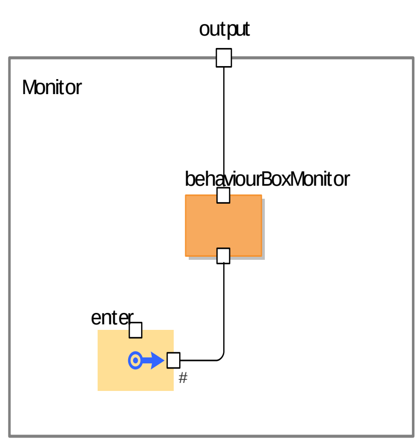


Figure 5.6: *Implementation: Object Monitor reflecting component model of monitors*

The remaining component models are implemented likewise, including the related *BehaviourBox* objects and the appropriate number of in- and outputs (Appendix B). The implementation of the component models *CIBUS*, *BIC1* and *BIC2* involves additional AnyLogic Enterprise Library objects *Split* and *Queue*. The former realises the splitting of the signal path within the CIBUS, the latter is needed to control the signal flow in the daisy chain across the BICs.

In addition to the common component objects, an extra component object *Rack* is defined. It does not include a *BehaviourBox* and as such is an object without any influence on the system behaviour. The object reflects the racks in the real MPS (containing several VME crates), and is introduced for reasons of a clear object hierarchy in the model implementation.

The naming scheme of the component objects used in the implementation deviates from the component names introduced in Chapter 4. In order to avoid confusion, Table 5.4 points out the deviations.

Table 5.4: *Implementation: Names of component objects compared to MPS model components*

Object	Component
BICbeam1	BIC1
BICbeam2	BIC2
CIBUS	CIBUS
Rack	-
VMEcrate	VME
VMEcrateCombinerCard	CC
BEE	BEE
FEE	FEE
Monitor	Monitor

5.2.1.3 Connect Component Objects

In order to build the MPS model structure, the component objects are connected in the AnyLogic graphical user interface either graphically (using the AnyLogic *connector* widget) or by explicitly written Java code. The latter particularly applies to the implementation of the daisy chains involved in the model. The required number of component objects of a certain type (i.e. instances of the same component class) is achieved by object replication.

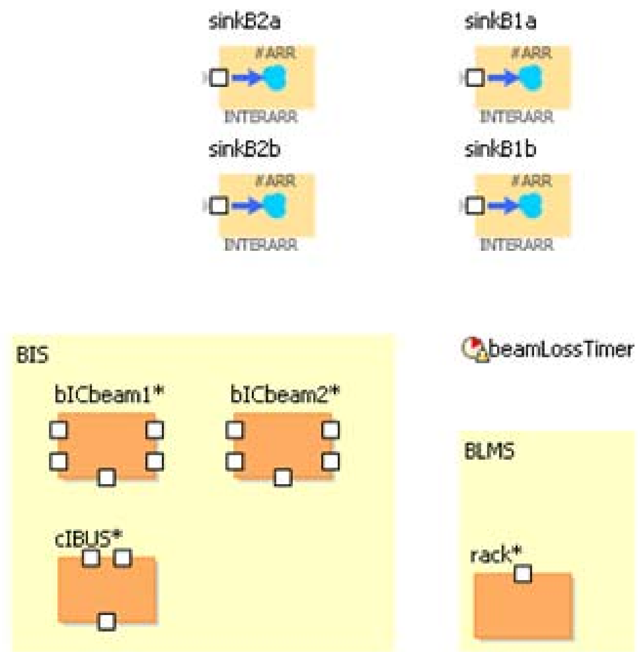


Figure 5.7: *Implementation: Model development environment Main*

Each AnyLogic project (model) by default includes an object class *Main* which provides the basic environment for model development. Figure 5.7 shows the graphic representation of the *Main* class structure for the MPS model. It contains the *CIBUS*, *BICbeam1* and *BICbeam2* objects representing the BIS. The asterisk after the names indicates object replications. The objects are connected according to the MPS model structure by explicit program code. The four *Sink* objects represent the end of the system signal path (corresponding to a redundant component E, Section 4.1).

The structure of the BLMS with its branches and their self-contained signal path allows for the encapsulation of the related component objects. It means that the (replicated) *Rack* object represents an entire BLMS branch with its components. The encapsulation is described by Table 5.5. The numbers in brackets indicate the number of object replications, i.e. a *Rack* contains three *VMEcrates*, which in turn includes one *VMEcombinerCard* and thirteen *BEE*, and so on. Objects *VMEcrateCombinerCards* and *Monitor* do not contain other component objects.

Timer object *beamLossTimer* in the *Main* environment (Fig. 5.7) relates to the implementation of the system demand. It is further explained in Section 5.2.1.5.

Table 5.5: *Implementation: BLMS branch encapsulation. In brackets: number of object replications*

Name	**Encapsulating**
Rack	VMEcrate (3)
VMEcrate	VMEcombinerCard, BEE (13)
VMEcrateCombinerCard	-
BEE	FEE (2)
FEE	Monitor (6)
Monitor	-

5.2.1.4 Set Model Input Parameters

The system demand rate x and component failure rates λ are defined as parameters in *Main*. Table 5.6 lists their names and the default values according to Chapter 4.

Table 5.6: *Implementation: Model input parameters*

Name	**Default value**	**Description**
lambdaBeamLoss	1E-2	x
lambdaFalseMonitor	1E-7	$\lambda^F_{Monitor}$
lambdaBlindMonitor	1E-7	$\lambda^B_{Monitor}$
lambdaFalseFEE	1E-6	λ^F_{FEE}
lambdaBlindFEE	1E-8	λ^B_{FEE}
lambdaFalseBEE	1E-8	λ^F_{BEE}
lambdaBlindBEE	1E-9	λ^B_{BEE}
lambdaFalseCC	1E-8	λ^F_{CC}
lambdaBlindCC	1E-9	λ^B_{CC}
lambdaFalseVME	1E-5	λ^F_{VME}
lambdaBlindVME	1E-8	λ^B_{VME}
lambdaFalseCIBUS	1E-6	λ^F_{CIBUS}
lambdaBlindCIBUS	1E-13	λ^B_{CIBUS}
lambdaFalseBIC1	1E-5	λ^F_{BIC1}
lambdaBlindBIC1	1E-13	λ^B_{BIC1}
lambdaFalseBIC2	1E-5	λ^F_{BIC2}
lambdaBlindBIC2	1E-13	λ^B_{BIC2}

Additionally, a number of auxiliary parameters are defined (Appendix B). They are used for

- object replication and daisy chain loops (parameters $numberOf <>$)
- data logging (parameters *mission*, *triggerType*, Section 5.2.1.6) and
- stop conditioning (parameters *dump*, *missedDump*, *trigger*, Section 5.2.1.7).

Observation time t_m is not defined explicitly as a parameter but inherently in the stop condition expression (Section 5.2.1.7).

5.2.1.5 Implement System Demand

The implementation of the system demand bases upon a timer (*beamLossTimer*, Fig. 5.7) that expires after exponentially distributed time intervals (defined by parameter *lambdaBeamLoss*, Tab. 5.6). The expiry of the timer results in the (simultaneous) triggering of a signal in six monitor objects of BLMS branch 1.

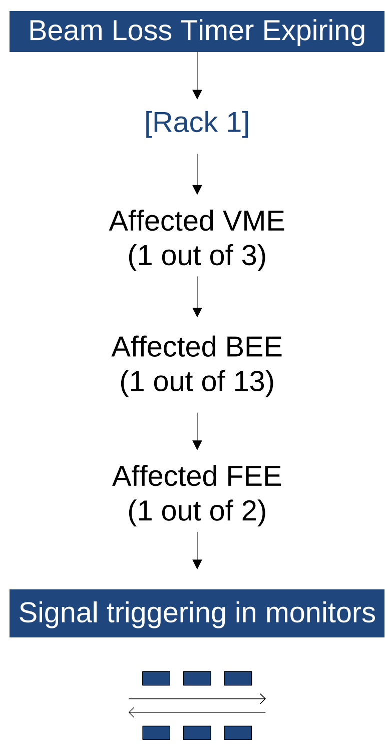


Figure 5.8: *Implementation: Schematic representation of the system demand algorithm*

Figure 5.8 shows a schematic representation of the algorithm. The monitors to be triggered are determined step-wise based on discrete uniform distribution. The algorithm starts with choosing one of the three *VMEcrate* of *Rack* 1 (representing BLMS branch 1). Next, one of the thirteen *BEE* of the chosen *VMEcrate* is determined, in turn followed by the choice of one of the connected two *FEE*. A signal is then triggered in the *Enter* object of each of the six *Monitors* related to the *FEE*. There is no model time delay between the expiring of the timer and the triggering of the signals in the monitors, i.e. they occur at the same model time. The timer is set to expire (at most) once in a single simulation run, in accordance with the model assumption of observation stop upon (first) trigger event (Section 4.1.2). Possible further timer expiries in a simulation run thus are irrelevant. Besides, they would interfere with the implementation of the stop conditions.

The program code segment defining the system demand is given in Appendix B.

5.2.1.6 Implement Data Logging

Table 5.7 shows the header of the simulation log file, specifying the logged data. The file includes 11 columns.

Table 5.7: *Implementation: Header of log file (log file columns)*

1	2	3	4	5-11
Mission	**Event type**	**Trigger type**	**Model time**	**Related component**

Data is logged upon the occurrence of one of four predefined events. According to Table 5.7, the log of the event occurrence includes the following data:

- **Mission** Number of the current mission
- **Event type** Type of the occurred event
- **Trigger type** Type of the underlying signal triggering
- **Model time** Time of occurrence
- **Related component** Underlying component (with regard to triggering)

Table 5.8 specifies the different event types and the condition for their logging. The implementation of the logging for the first three events is straight-forward, based on the leaving or entering of a signal in an object. The logging of event *missedDump* however is more sophisticated since it bases upon the non-occurrence of a (dump) event. The problem is solved using boolean auxiliary parameter *trigger* of *Main* (Section 5.2.1.4). It is set *true* upon *beamLossTimer* expiry and upon component object transition to *falseState*. The related *dump* event, i.e. the triggered signal arriving at the *Sink*, sets the parameter back to *false* within the same model time. Expression $trigger = true$ at a later point of model time therefore indicates a *missedDump* event. This condition is checked previous to any execution of action related to *beamLossTimer* expiry or state transition. In case of $trigger = true$, a *missedDump* event is logged. Since the logging of such events are delayed, the related model time cannot be used for statistics on event occurrence. It is important to note that in spite of the delay, there is no spurious extra logging, because the delay by definition does not include any further events (which would be logged). A simulation run without a trigger event occurrence is logged at model time $t = 12$ by code -1.

Table 5.8: *Implementation: Event types (log file rows)*

Event type	**Code**	**Log condition**
beamLoss	0	Signal leaving *Enter* object of *Monitor* (Fig. 5.6)
falseGenerator	1	Signal leaving *FalseGenerator* object of BehaviourBox (Fig. 5.5)
dump	4	Signal entering a *Sink* object of *Main* (Fig. 5.7)
missedDump	5	$trigger = true$ at *beamLossTimer* expiry
		$trigger = true$ previous to any state transition
		$trigger = true$ at model time $t = 12$

The log of an event includes the specification of the underlying trigger event. There are two types, *system demand* or *false failure* (Table 5.9). Their logging is based on auxiliary parameter *triggerType* (Section 5.2.1.4) and specifies the *dump* and *missedDump* events in terms of the global scenarios (Section 5.2.3).

Table 5.9: *Implementation: Trigger types*

Trigger type	**Code**
system demand	0
false failure	1

The detailed key for the log files is given in Appendix B. It includes the key for the identification of the component object underlying the events. The identification is based on the different component objects and the indexes of the related replications. The key file also covers additional events useful for a more sophisticated data analysis (Section 5.2.3) or needed for the data analysis of the extended models (Chapter 9).

5.2.1.7 Define Stop Conditions

According to the modelling approach, a simulation run is stopped upon the occurrence of any of the global scenarios. The scenarios are defined by the events and the underlying trigger (Section 5.2.1.6). The stop of a simulation run relies on three stop conditions that base on boolean auxiliary parameters *dump* and *missedDump* (Tab. 5.10). If a stop condition is fulfilled, simulation is stopped and the related scenario is logged.

Table 5.10: *Implementation: Simulation stop conditions*

Scen.	Stop condition	Fulfilled
I	$t = 12$	upon model time reaching 12 hours
II, III	$dump = true$	upon signal entering *Sink* object in *Main*
IV, V	$missedDump = true$	if $trigger = true$ at *beamLossTimer* expiry
		if $trigger = true$ previous to any statechart transition
		if $trigger = true$ at model time $t = 12$

Simulation is stopped upon model time reaching 12 hours, which is the case in absence of a triggering event only. Otherwise, a signal entering one of the *Sink* objects in *Main* (Fig. 5.7) sets parameter *dump* to *true*, which represents the stop condition for scenarios *emergency beam dump* (II) and *false beam dump* (III). The fulfilling of the stop condition for the missed beam dump scenarios (IV, V) is linked to the detection and logging of event *missedDump* (Section 5.2.1.6, Tab. 5.8).

5.2.1.8 Build Model

The model is built using the related command in AnyLogic. It is then ready to be used for simulations which are run either directly in AnyLogic or independently by invoking the Java application in command line mode. The latter is based on an executable jar-file, in this case *LHC_MPS_v2.jar*, that is generated upon building the model.

5.2.2 Run Simulations

Within AnyLogic, simulations are performed simply by using the related command to run the model (*LHC_MPS_v2.alp*). The simulation data is logged in a text file called *statslog-na.txt* in folder *data*. The simulation stops after a certain number of missions defined by parameter *maxMissions* (included in the model).

For the benefit of a reduced simulation time, the case studies and sensitivity analysis presented in this work (Chapters 7 and 8) base upon simulations performed on a multi-processor server running Linux, using the independent Java application (i.e. file *LHC_MPS_v2.jar*). The framework for running the simulations was developed and documented in detail by Nibali [52]. The following sections provide a brief overview thereof. The specifications of the simulation server are given in Appendix B.

5.2.2.1 Create Simulation Environment

The simulation environment is provided by a folder with the content shown in Table 5.11.

Table 5.11: *Implementation: Simulation environment*

File/Folder	Description
LHC_MPS_v2.jar	Executable jar-file of model
StartModell.sh	Script to start simulation
lib	Folder containing runtime libraries *xjanylogic5engine.jar* and *enterprise_library.jar*
PrepareModel.sh	Script to generate simulation environment automatically
CheckStatus.sh	Script to observe simulation during runtime
CreateFilelist.sh	Script for data analysis
CleanEnv.sh	Script for environment cleaning

The first three entires of the list are mandatory to run a simulation. The remaining scripts are optional. They support specific tasks of the simulation procedure.

5.2.2.2 Set Simulation Parameters and Run Simulations

The simulation is started in a Unix shell using the following command (in the directory related to the environment folder introduced above):

```
nohup ./StartModell.sh
<# missions> <# threads> <"name extension"> <target folder> &
```

The brackets <> indicate the simulation parameters. The desired number of simulated missions and number of parallel simulation threads have to be specified, as well as a name extension and target folder for the log files. The related files are named as follows:

```
statslog-<"name extension">-<file number>-<# missions>.txt
```

The files are numbered by *file number* for the case that a simulation results in several files (depending on the number of missions and the number of parallel threads). If desired, the simulation progress can be observed using script *CheckStatus.sh*.

5.2.3 Analyse Data

The analysis of the log files is done using Open Source statistics software R [53]. The extraction of the relative frequencies of the global scenarios uses columns 1 (*Mission*), 2 (*Event Type*) and 3 (*Trigger type*) of the log files (Tab. 5.7) and simply bases upon counting the number of occurrences of the related code combinations.

The number of simulated missions is extracted in column 1. It represents the denominator of the relative frequencies. The code combinations for the different scenarios, providing the numerators, are shown in Table 5.12.

Table 5.12: *Implementation: Log code combinations of scenarios*

Scenario	**Event type** (Column 2)	**Trigger type** (Column 3)
Emergency beam dump	4	0
False beam dump	4	1
Missed emergency beam dump	5	0
Missed false beam dump	5	1

The consideration of columns 5-11 (specifying the underlying components) additionally allows for a straightforward extraction of the relative importances with regard to false beam dump triggering. Similarly, the system demand pattern can be extracted. The inclusion of column 4 (*Model time*) allows for an analysis in terms of time of event occurrence. The extraction of importances with regard to signal absorbing requires a more sophisticated data analysis (not addressed within this study).

In case of several log files, the data analysis is supported by script CreateFilelist.sh. Additional information on the handling are provided by Nibali [52].

Chapter 6

Analytical Description

Besides the Monte Carlo approach presented in the previous chapter, an analytical approach was followed. This chapter introduces the resulting analytical description and the related implementation of the MPS model (Fig. 6.1). It is based on basic probability calculation and numerical integration.

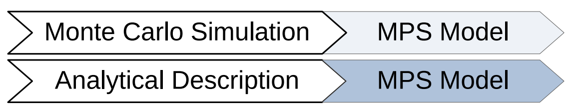


Figure 6.1: *Analytical description: Second implementation of MPS model*

6.1 Concept

The component model represents the basic module of the MPS model and the implementation of the analytical description likewise. The concept of the analytical description is introduced by means of the component model and the six-component system presented above (Section 4.1, Fig. 4.1). The described events form the basis of the scenarios *missed emergency beam dump* and *false beam dump*.

6.1.1 Component Model

The analytical description is based on the exponential probability density function $f(t) = \lambda \cdot e^{-\lambda \cdot t}$ defining the state transitions. The density function $f^F(t)$ ($f^B(t)$) reflects the distribution of the time to transition T^F (T^B) to state *false* (*blind*), if the transition is considered independently (Fig. 6.2).

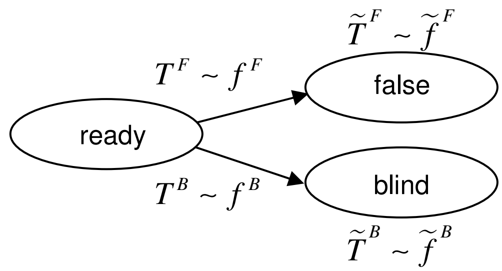


Figure 6.2: *Analytical description: State diagram for system component. T: Time to transition, $f(t)$: Probability density function.*

The transitions in the component model however are not independent. The dependency is described on the basis of the *false* failure mode. It applies to the *blind* mode accordingly. The component going *false* at time t conditions that it has not gone *blind* at time $t' < t$, i.e. it survived with regard to the *blind* mode up to time t. The adjusted function, defining the transitions that actually come into effect, is thus given by

$$\widetilde{f}^F(t) = f^F(t)(1 - F^B(t)) \tag{6.1}$$

where $F^B(t)$ is the failure probability with regard to the *blind* mode defined by

$$F^B(t) = \int_0^t f^B(t')dt' \tag{6.2}$$

The probability of the component going *false* during time period t with Eq. 6.1 becomes

$$\widetilde{F}^F(t) = \int_0^t \widetilde{f}^F(t')dt'. \tag{6.3}$$

The probability of the component surviving with regard to the *false* mode, i.e. being either *ready* or *blind*, is given by

$$\widetilde{R}^F(t) = 1 - \widetilde{F}^F(t). \tag{6.4}$$

The state diagram of component S includes one transition only. It is defined by exponential probability density function $f^D(t) = x \cdot e^{-x \cdot t}$ based on system demand rate x (Fig. 6.3). Accordingly, the probability of component S going *system demanding* during time period t is given by $F^D(t) = \int_0^t f^D(t')dt'$.

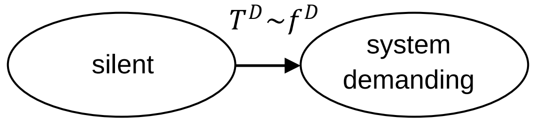


Figure 6.3: *Analytical description: State diagram of component S. T: Time to transition, $f(t)$: Probability density function.*

As mentioned above, the component model represents a simple Markov model. It can be shown that the presented analytical description of the component model corresponds to the related Markov equations.

6.1.2 Six-Component System

Fig. 6.4 illustrates a system of six identical components according to the basic system component model (Fig. 6.2), representing a signal path from the virtual component S (start) to E (end). The behaviour of component S follows the state diagram illustrated in Fig. 6.3.

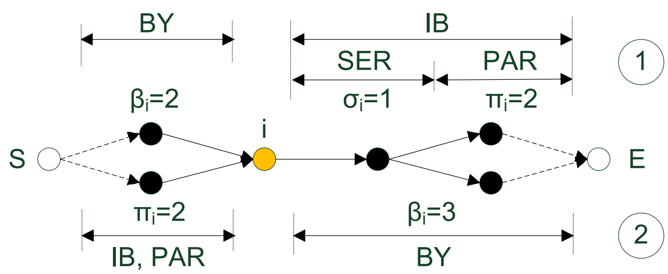


Figure 6.4: *Six-component system representing a signal path from S to E.*

A signal is triggered upon component S going *system demanding* or upon a component going *false*, and it is absorbed by *blind* components. The system features 1-out-of-2 redundancy at the start and end of the signal path.

The analytical description is introduced on the basis of basic events:

- **Event 1** A signal is triggered in component i within time period t (due to i going *false*) and goes through to the end of the signal path E.

- **Event 2** A signal is triggered in component S within time period t (due to S going *system demanding*) and is absorbed by component i (due to i having gone *blind*).

In accordance with the model assumptions (Section 4.3.1.2), both events exclude signal triggering by other components during time period t.

The deduction of the equations relies on the following steps:

- Subdivision of the signal path into specified sections
- Definition of component states related to the sections
- Composition of the equations including the states of all components

The subdivision of the signal path depends on the system configuration (i.e. event) under consideration. It is based on the signal triggering component and the absorbing component and the section in-between (section *IB*). In order to allow the triggered signal to go through to the absorbing component, the states of the components in section *IB* have to reflect an open signal path. The components in the section beyond (*BY*) the relevant section IB may be *ready* or *blind* at the time of the signal triggering. They are not allowed to be in the *false* state though, since that would imply a previous signal triggering. Component S is treated individually. It either acts as the triggering component or as a component beyond the relevant section. In the latter case, it has to be *ready* (since its second state, *system demanding*, includes signal triggering). The composition of the equation bases on the multiplication (and integration) of the transition probability of the triggering component and the component state probabilities of the remaining components. The transition probability refers to the (adjusted) probability density function introduced above. The multiplication depends on the statistical independence of the state transitions across the components.

The subdivision of the signal path with regard to Event 1 is illustrated in Fig. 6.4 (top). IB marks the relevant section in-between the signal triggering component i and the end of the signal path E. BY marks the section beyond the relevant section IB. Section IB is further divided into subsections SER (including components connected in series) and PAR (including components connected in parallel). β, σ, π refer to the number of components within the related section.

The probability of a signal being generated in component i during time period t and going through to the end of the signal path E (Event 1) derives from Equations 6.1, 6.3 and 6.4 to

$$P^F(false_{i \to E}, t) = \int_0^t \widetilde{f}_i^F I_S I_{BY} I_{SER} I_{PAR} dt'. \tag{6.5}$$

Table 6.1: *Analytical description: Equation terms for Event 1*

Term	Formula	Description
$\widetilde{f}_i^F$	$\widetilde{f}_i^F$	Component i going *false*
I_S	$1 - F^D$	Survival of component S
I_{BY}	$(1 - \widetilde{F}^F)^{\beta_i}$	Survival of components in BY with regard to *false* mode
I_{SER}	$[(1 - F^F)(1 - F^B)]^{\sigma_i}$	Survival of component in SER
I_{PAR}	$(1 - \widetilde{F}^F)^{\pi_i} - (\widetilde{F}^B)^{\pi_i}$	Survival of redundant components in PAR with regard to *false* mode, excluding case with both components *blind*

The terms of Equation 6.5 are explained in Table 6.1.

The related global probability (i.e. any component going *false*) results to

$$P^F(false_{\to E}, t) = \sum_{i=1}^{6} P^F(false_{i\to E}, t) \tag{6.6}$$

based on the mutually exclusive events described by $P^F(false_{i\to E}, t)$ for different components i.

With path subdivision according to Event 2 (Fig. 6.4, bottom), the probability of a signal being generated in S during time period t and being absorbed by component i analogously derives to

$$P^D(S_{\to blind\ i}, t) = \int_0^t f^D I_{IB} \widetilde{F}_i^B I_{BY} dt'. \tag{6.7}$$

The terms of equation 6.7 are explained in Table 6.2.

The related global probability $P^D(S_{\to blind}, t)$ results from summing up all signal absorbing events, taking into account the redundancies.

Given the signal generated in S representing an emergency beam dump request, probability $P^D(S_{\to blind}, t)$ corresponds to the probability of scenario *missed emergency beam dump* (i.e. $P_{IV}(t)$, affecting machine safety). With the signal generated upon a component going *false* representing a false beam dump request, $P^F(false_{\to E}, t)$ (Eq. 6.6) corresponds to the probability of scenario *false beam dump* (i.e. $P_{III}(t)$, affecting beam availability).

Table 6.2: *Analytical description: Equation terms for Event 2*

Term	Formula	Description
f^D	f^D	Component S going *system demanding*
I_{IB}	$(1-\widetilde{F}^F)^{\pi_i} - (\widetilde{F}^B)^{\pi_i}$	Survival of redundant components in IB with regard to *false* mode, excluding case with both components *blind*
$\widetilde{F}_i^B$	$\widetilde{F}_i^B$	Component i being *blind*
I_{BY}	$(1-\widetilde{F}^F)^{\beta_i}$	Survival of components in BY with regard to *false* mode

6.2 MPS Model Implementation

The analytical description of the MPS model derives from the above description of the six-component system. It was implemented in close cooperation with Nibali [54] using software Maple 12. The implementation involves the procedure described in Table 6.3. The listed steps represent the key tasks of the implementation. They are discussed in short in the following sections for the purpose of facilitating the tracing of the implementation and the underlying files.

Table 6.3: *Implementation: Procedure*

Task	Step	File
Build equations	Set model input parameters	Constants.txt
	Define terms	TermsAndFunctions.txt
	Compose equations	B5A_demand_end.txt
		B5B_false_end.txt
		B6A_demand_absorbed.txt
		B6B_false_absorbed.txt
		B6X_false_absorbed.txt
		B7_not_demand_false.txt
	Build equations	Generate_Library.mw
	Define cross-checks	cc.txt
Run calculation	Set calculation parameters	ParameterVariation.mw
	Run calculations	ditto
Analyse results	Analyse results	result_D<>_P<>-<>.txt

The listed work files underlying the implementation are given in Appendix C.

6.2.1 Build Equations

The building of the equations for the MPS model follows the modular conception of the modelling approach. The model input parameters, in particular the component failure rates, define the component state transitions and the state probabilities. The related (adjusted) probability density and distribution functions on component level represent the modules (*terms*) that the system equations are composed of.

6.2.1.1 Set Model Input Parameters

The model input parameters are set in file *Constants.txt*. Table 6.4 lists them and their default values according to the model description in Section 4.2.

Table 6.4: *Implementation: Model input parameters*

	Default value	Description
x	0.01	System demand rate x
lf	[1e-7, 1e-6, 1e-8, 1e-8, 1e-5, 1e-6, 1e-5, 1e-5]	Failure rates λ_i^F, i=1...8
lb	[1e-7, 1e-8, 1e-9, 1e-9, 1e-8, 1e-13, 1e-13, 1e-13]	Failure rates λ_i^B, i=1...8
t0	0	Start of observation period
tf	12	End of observation period, corresponding to observation time in case of $t0 = 0$

Additionally, a number of auxiliary model parameters are to be set (Appendix C). They define factors and exponents in the equations and the loops of the building algorithm (Section 6.2.1.3).

6.2.1.2 Define Terms

The terms are functions representing the modules of the equations. They are defined in file *TermsAndFunctions.txt*. Table 6.5 lists them and links them to the component's (adjusted) density and state probability functions introduced above.

Factor $a = 1/(VMEcrate \cdot BEE \cdot FEE)$ in termD() is made up by auxiliary quantities. It represents the reciprocal of the number of FEEs in one BLMS branch. The factor derives from the characteristics of the system demand pattern and the equation definition (Section 6.2.1.3). It is needed in the equations of the scenarios involving system

Table 6.5: *Implementation: Equation terms description*

Name	Function	Description
termD()	$f^D \cdot a$	Component S going *system demanding*
termND()	R^D	Component S being *silent*
termF(i)	$\widetilde{f}_i^F$	Component of class i going *false*
termB(i)	$\widetilde{F}_i^B$	Component of class i being *blind*
termIB(i,p)	$(R_i^R)^p$	p components of class i being *ready*
termEX(i,p)	$(R_i^R + \widetilde{F}_i^B)^p$	p components of class i being *ready* or *blind*

demand (Scenarios II and IV) and is assigned to termD() instead of being added individually to each of the affected equations.

It is to be noted that i here refers to component class i, according to the MPS model description in Chapter 4.2. termIB(i,p) covers a number of p components of the same component class i being in state *ready*. termEX(i,p) includes p components of class i being *ready* or *blind*.

Table 6.6 provides the default definitions of the terms based on the input parameters introduced in Table 6.4. Factor $lf[i]$ ($lb[i]$) refers to element i of vector lf (lb, Tab. 6.4)

Table 6.6: *Implementation: Equation terms definition*

Name	Definition
termD()	$x \cdot e^{-x \cdot t} \cdot a$
termND()	$e^{-x \cdot t}$
termF(i)	$lf[i] \cdot e^{-(lf[i]+lb[i]) \cdot t}$
termB(i)	$lb[i]/(lb[i] + lf[i]) \cdot (1 - e^{-(lb[i]+lf[i]) \cdot t})$
termIB(i,p)	$e^{-(lb[i]+lf[i]) \cdot t \cdot p}$
termEX(i,p)	$((lb[i] + lf[i] \cdot e^{-(lb[i]+lf[i]) \cdot t})/(lb[i] + lf[i]))^p$

6.2.1.3 Compose Equations

The composition of the equations relies on the concept described in Section 6.1.2. For each of the global scenarios, all possible combinatorial pairs of triggering and absorbing components have to be identified. For each pair, the equation (i.e. equation integrand) is composed based on the underlying subdivision of the signal path. The integration over the sum of all integrands of a scenario yields the equation of the scenario.

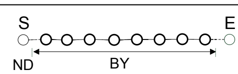
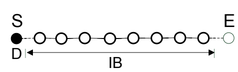
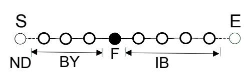
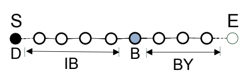
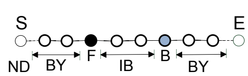
The resulting equations for the different scenarios according to Table 6.7 are given in Appendix C. Scenario V, *missed false beam dump*, requires two independent implementations in order to extract the importances with regard to triggering (B6B) and absorbing (B6X).

Table 6.7: *Implementation: Equation files*

Scenario	Equation	File
I	B7	B7_not_demand_false.txt
II	B5A	B5A_demand_end.txt
III	B5B	B5B_false_end.txt
IV	B6A	B6A_demand_absorbed.txt
V	B6B, B6X	B6B_false_absorbed.txt, B6X_false_absorbed.txt

The composition of the integrands based on the terms defined above (Tab. 6.5 and 6.6) is exemplified by means of a system with components connected in series (Tab. 6.8).

Table 6.8: *Implementation: Integrand structure of scenarios*

Scen.	Path configuration	Integrand
I		$termND \cdot (\prod termEX)$, Equation
II		**termD** $\cdot (\prod termIB)$
III		$termND \cdot (\prod termEX) \cdot$ **termF** $\cdot (\prod termIB)$
IV		**termD** $\cdot (\prod termIB) \cdot termB \cdot (\prod termEX)$
V		$termND \cdot (\prod termEX) \cdot$ **termF** $\cdot (\prod termIB) \cdot termB \cdot (\prod termEX)$

The term given for Scenario I (*completed mission*) is a product of probabilities. It represents the scenario probability and thus relates to the complete equation rather than a (single) integrand. Unlike the remaining terms, it does not contain an (adjusted) density function (i.e. termD or termF). No additional integration is required.

The integrands of the remaining scenarios (II to V) are composed in a combinatorial matter according to the following steps:

- Define triggering component: S in case of emergency beam dump triggering (*termD*), system component in case of false beam dump triggering (*termF*)

- Define absorbing component: E in case of successful triggering, system component in case of missed triggering (*termB*)
- Identify the components of the path section in-between (IB) the triggering and absorbing component
- Identify the components beyond (BY) the in-between section

In case of components connected in series, the in-between section is described simply by the product of *termIB* of the related components. Accordingly, the section beyond is described by the product of *termND* (component S) and *termEX* of the related system components. In case of parallel (redundant) path structure in the in-between section, the related factor of the integrand is a composition (including sums) of different terms of the involved components (cp. six-component system, Section 6.1.2).

The number of integrands in a scenario is defined by the number of possible pairs of triggering and absorbing components (e.g. for eight-component serial system: II: 1, III: 8, IV: 8, V: 28). If the pair is adjacent, the IB part is omitted. It is to be noted that each integrand includes the state of each and every component of the system. All system components beyond the in-between section contribute with factor *termEX*.

Besides the global scenarios (I-V, Tab. 6.8), Scenarios *emergency beam dump triggered* (covering II, IV) and *false beam dump triggered* (covering III, V) are implemented independently (B1, B2plus, Appendix C). They entail additional options for verification (Section 8.1).

6.2.1.4 Build Equations

The equations are built by executing Maple worksheet *Generate_Library.mw*. It reads the previously introduced files with the parameters, the terms and equations and combines them in the library *LHC_MPS.mla*. This library contains the entire analytical description of the MPS model and is ready for use. Calculations can be performed in any Maple worksheet by loading the library and executing the worksheet (Section 6.2.2).

6.2.1.5 Define Cross-Checks

The characteristics of the scenarios and the implementation of the equations allows for a set of cross-checks to be defined, which facilitate the debugging of the implementation and serve the verification of the results (Section 8.1). These cross-checks are defined in file *cc.txt*.

6.2.2 Run Calculation

File *ParameterVariation.mw* is a ready-for-use Maple worksheet to perform calculations. It loads the library (*LHC_MPS.mla*) and the cross-checks (*cc.txt*). Two calculation parameters are to be set, *Precision* and *Digits*. They are described in Table 6.9.

Table 6.9: *Implementation: Calculation parameters*

Name	Default value	Description
Precision	6	Decimal place to which accuracy of results of numerical integration is guaranteed
Digits	10	Number of digits of the floating point numbers used in calculation

The two parameters are discussed in more detail by Nibali [55]. More extensive specifications on the parameters and their interaction require an in-depth analysis of Maple software. With regard to actual application however it is sufficient to note that the presented default values have proven to be appropriate for reasonably fast and accurate calculations. For more accurate results, particularly in view of rare scenarios and related small probabilities, higher precision (and more digits) are required at the expense of calculation time.

The calculation runs can optionally be assigned a specific name (parameter *SetName*), e.g. relating to the underlying model parameter set. Upon execution of the worksheet, a file $result\_D < Digits > \_P < Precision > - < SetName > .txt$ is generated, containing the scenario probabilities and some additional calculation specifications (Section 6.2.3). The calculation bases upon Maple's default integration method and with the default calculation parameters takes about a second on a multi-processor server running Windows. Detailed specifications of the calculation server are given in Appendix C.

For parameter variation studies, there are basically two options. Model input parameters, e.g. component failure rates λ, can be changed in either

- file *Contants.txt*, requiring the subsequent generation of the related library, or

- file *ParameterVariation.mw* through reinitialisation.

For simple parameter variation the latter is recommended. This is possible since the content of the library, once loaded in a worksheet, is global.

6.2.3 Analyse Results

File $result\_D < Digits > \_P < Precision > - < SetName > .txt$ contains the calculation results:

- Scenario probabilities ($B1$, $B2plus$, $B5A$, $B5B$, $B6A$, $B6B$, $B6X$ and $B7$) with importances
- Results of the cross-checks ($cc1$, $cc2$, $cc3$, $cc3split$, $cc4$) for verification

Additionally, it provides the most relevant specifications of the calculation:

- Model input parameters (x, lf, lb, $t0$, tf)
- Calculation parameters ($Precision, Digits$, integration method $intMethod$)
- Calculation time ($ttime$, [s])
- Specifications on the individual numerical integration processes ($NumEvent$)

It is to be noted that the results are displayed according to parameter *Digits*. They are accurate to the decimal place according to parameter *Precision* only.

Additional information on the files and the handling of the implementation are provided by Nibali [54].

Chapter 7

Results

This chapter presents the results of six case studies on the MPS model, based on the implementation of the analytical description (Fig. 7.1).

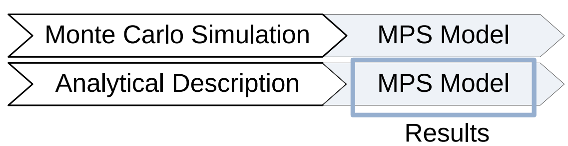


Figure 7.1: *Results: Case studies based on analytical description*

The case studies vary in the input parameters. Three sets of component failure rates are used with two different system demand rates (Tab. 7.1).

Table 7.1: *Parameter sets for case studies*

		Demand rate (x)	
		STANDARD	UPDATE
Component Failure Rates (λ)	STANDARD	A	B
	UPDATE	C	D
	CIBUS	E	F

The basic parameter set, *STANDARD A*, was defined in an early stage of the project when parameters were needed to feed the first model implementation. Mid-2009, these parameters were reconsidered and adjusted, taking into account interim operational experience. The adjusted parameters define the parameter sets *B* through *F*. The related rates λ and x are introduced in the following section. The results of the case studies are presented in separate sections, the discussion of the results is handled comprehensively in Section 7.6

7.1 Input Parameters

The comprehensive analysis of the components with regard to their failure rates (as suggested in Chapter 4.1.3.1) and of the system demand is beyond the scope of this thesis. The figures presented in this sections are rough estimates based on limited data. Accordingly, the results are to be considered in a relative rather than absolute manner. The purpose of the case studies is to demonstrate the use of the approach.

7.1.1 Demand Rates

Table 7.2 shows the two system demand rates used in the case studies. The *STANDARD* value is based on expert judgment assuming a mean time to beam loss of 100 hours. It is a rough estimate representing the scenario of about 10% of the missions encountering a (non-nominal) beam loss event (based on a mission time of ten hours). The *UPDATE* value bases upon operational experience with the Hadron-Electron Ring Accelerator (HERA) at DESY (Deutsches Elektronen-Synchrotron) in Hamburg, Germany. HERA as a collider, even though much smaller than the LHC, provides benchmarks for certain characteristics of the operational performance. HERA operational data is analysed with regard to beam loss by Wittenburg [56]. He states that between 1994 and 2004, about 1000 beam dumps triggered by beam loss monitors occurred (without magnet quench). Additionally, about 200 beam loss induced quenches appeared. This relates to a total of 1200 beam loss events in 10 years. Assuming 200 days of operation per year (2 missions per day), an estimate of the system demand rate (based on beam loss) can be calculated as follows (maximum likelihood estimator of a poisson parameter [57]).

$$x = \frac{1200}{10y \cdot 200d/y \cdot 24h/d} = 2.5 \cdot 10^{-2} h^{-1} \tag{7.1}$$

Following a conservative approach, 3E-2/h is used as *UPDATE* demand rate (Tab. 7.2).

Table 7.2: *System demand rates*

	STANDARD	UPDATE
x [h^{-1}]	1E-2	3E-2

7.1.2 Component Failure Rates

Table 7.3 shows the sets of failure rates used in the case studies.

Table 7.3: *Component failure rates*

		STANDARD		UPDATE		CIBUS	
		λ_i^F	λ_i^B	λ_i^F	λ_i^B	λ_i^F	λ_i^B
i	Component *i*	[h^{-1}]	[h^{-1}]	[h^{-1}]	[h^{-1}]	[h^{-1}]	[h^{-1}]
1	Monitor	1E-7	1E-7	1E-7	1E-7	1E-7	1E-7
2	FEE	1E-6	1E-8	1E-6	1E-8	1E-6	1E-8
3	BEE	1E-8	1E-9	**1E-6**	1E-9	1E-8	1E-9
4	CC	1E-8	1E-9	1E-8	1E-9	1E-8	1E-9
5	VME	1E-5	1E-8	1E-5	1E-8	1E-5	1E-8
6	CIBUS	1E-6	1E-13	**1E-4**	**1E-11**	1E-6	**1E-6**
7	BIC1	1E-5	1E-13	**1E-4**	**1E-12**	1E-5	1E-13
8	BIC2	1E-5	1E-13	**1E-4**	**1E-12**	1E-5	1E-13

The *STANDARD* set, with regard to the BLMS components (1-5), is based on the study performed by Guaglio [20]. His work includes fault trees of a BLMS branch with top events corresponding to *blind* failure and *false* failure of the BLMS. As a first approximation and following the rare event approach described by Guaglio [20], the failure rates of the basic events in the fault trees are summed up to the system level corresponding to the components in the model. The rates for the BIS components (6-8) are derived from expert judgement [58].

The *UPDATE* set differs from the *STANDARD* set in rate λ_{BEE}^F and the rates of the BIS components. Rate λ_{BEE}^F is based on experience gained during LHC commissioning. During the time period of a year, one of about 350 installed DAB cards (used in BEE) failed in the *false* mode [59]. Assuming 200 days of operation per year, a rough estimate for rate λ_{BEE}^F is calculated according to Equation 7.2:

$$\lambda_{BEE}^F = \frac{1}{350 \cdot 1y \cdot 200d/y \cdot 24h/d} = 6 \cdot 10^{-7} h^{-1} \tag{7.2}$$

Following a conservative approach, 1E-6/h is used as rate λ_{BEE}^F (Tab. 7.3).

In addition to the mentioned failure of the DAB card, two failures of the *false* mode occurred in the VME crates. Given 25 installed crates, the related estimate following Equation 7.2 yields a figure in the order of magnitude of 1E-5/h for rate λ_{VME}^F, corresponding to the *STANDARD* value. Therefore, no adaptation is required.

The *UPDATE* rates of the BIS components are based on a FMECA performed by Todd ([1], Table H.1), roughly resulting in an increase by one (λ_{BIC1} and λ_{BIC2}) and two orders of magnitude (λ_{CIBUS}).

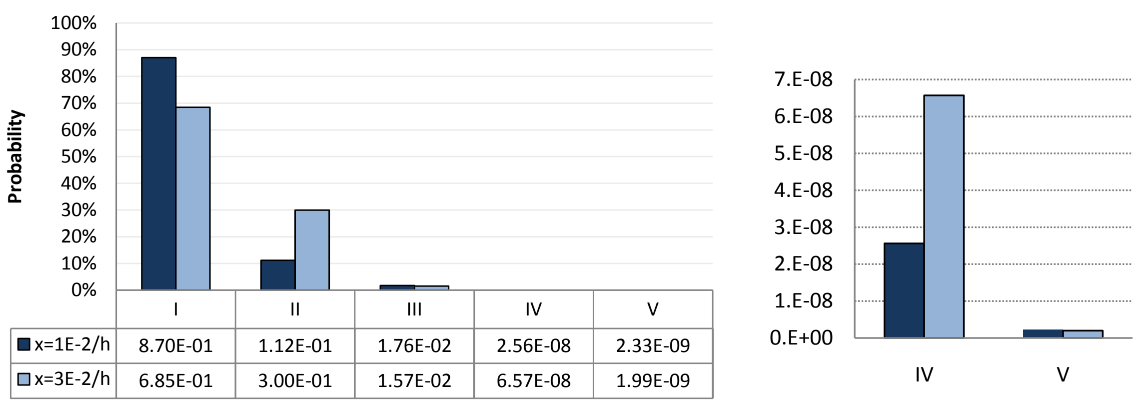
The *CIBUS* set differs from the *STANDARD* set in rate λ^B_{CIBUS}. It is based on experience gained during LHC commissioning between 2006 and 2009. During that time period, one CIBUS failure of the *blind* mode (internally referred to as double-blind failure) occurred [60]. It was identified to eventually base upon human failure and meanwhile the cause has been eliminated. However, for the purpose of a what-if study, the failure occurrence is taken as the basis for an adapted rate λ^B_{CIBUS}. A rough estimate is calculated based on 300 installed CIBUS and 200 days of operation per year (Eq. 7.3).

$$\lambda^B_{CIBUS} = \frac{1}{300 \cdot 3y \cdot 200d/y \cdot 24h/d} = 2.3 \cdot 10^{-7} h^{-1} \tag{7.3}$$

Following a conservative approach, 1E-6/h is used as rate λ^B_{CIBUS} (Tab. 7.3).

7.2 Case Study: STANDARD

The results presented in this and the following sections base upon the analytical description (Chapter 6). The related simulation results are used for verification purposes (Chapter 8). The calculations were performed using Maple's default integration method with a precision of 15 decimal places for the numerical integration (Section 6.2.2). With the given precision, the total calculation time amounted to a few minutes on a multi-processor server (Appendix C).



	I	II	III	IV	V
x=1E-2/h	8.70E-01	1.12E-01	1.76E-02	2.56E-08	2.33E-09
x=3E-2/h	6.85E-01	3.00E-01	1.57E-02	6.57E-08	1.99E-09

Figure 7.2: *STANDARD A, B: Scenario probabilities*

Fig. 7.2 shows the probabilities of the global scenarios for the parameter sets *STANDARD A* (1E-2/h) and *B* (3E-2/h). *STANDARD A* yields 87.0% of the missions completed (I), while the remaining missions are ended early by an emergency beam dump (II, 11.2%) or false beam dump (III, 1.8%). The safety relevant probability of a missed emergency beam dump (IV) is in the range of 1E-8. Given 400 missions per year, the probabilities correspond to

45 emergency beam dumps per year,

7 false beam dumps per year and

1 missed emergency beam dump in 100,000 years.

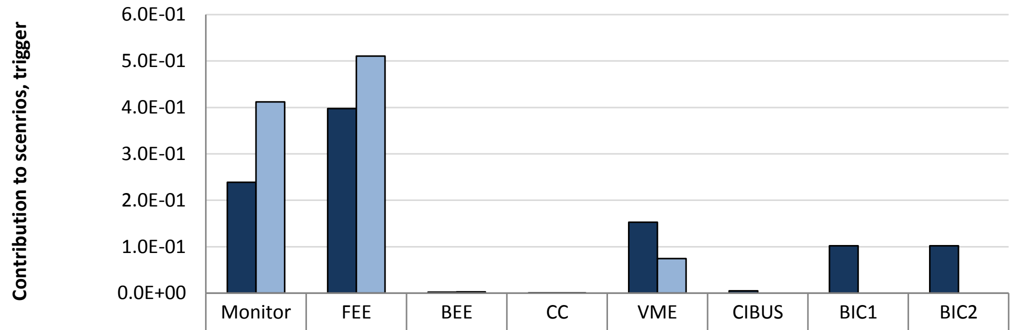
The increase of the system demand rate by factor 3 (*STANDARD B*) leads to a decrease of the completed missions by about 20%, leaving one-third of the missions early ended. Both the probabilities of an emergency beam dump (II) and missed emergency beam dump(IV) are increased by roughly factor 2.5, reflecting the increase of the system demand rate. The scenarios *false beam dump* (III) and *missed false beam dump* (V) encounter a decrease of 10 and 15%.

In terms of an importance analysis, the probabilities of scenarios III (*false beam dump*), IV (*missed emergency beam dump*) and V (*missed false beam dump*) are broken down with regard to the component classes and their contribution to the scenarios. Fig. 7.3 shows the relative contribution of the components to the false dump request triggers broken down to successful (resulting in false beam dumps) and missed triggers (resulting in missed false beam dumps).

Nearly 40% of the false beam dumps are triggered by FEEs. The Monitors and BICs contribute with about 25% and 20% respectively. In the case of missed false beam dumps, the relative importance of the FEEs and Monitors is even more distinct with about 50% and 40% respectively.

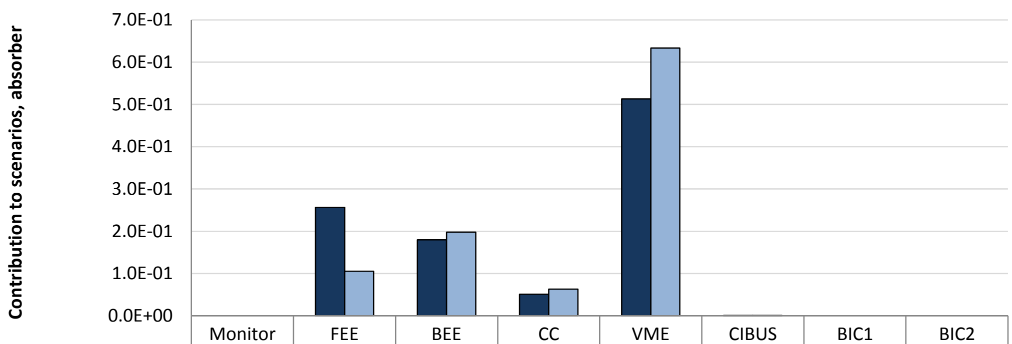
Fig. 7.4 shows the relative contribution of the components to the absorbing of dump requests, broken down to emergency (resulting in missed emergency beam dumps) and false beam dump requests (resulting in missed false beam dumps).

More than 50% of missed emergency dumps are caused by VMEs, the FEEs contribute with about 25%. In the case of missed false beam dumps, the importance of the VMEs is even more distinct with almost 65%. The absence of monitors contributing to missed beam dumps is reasonable in view of their 1oo6-redundancy 4.2.3.



	Monitor	FEE	BEE	CC	VME	CIBUS	BIC1	BIC2
False dump	2.39E-01	3.98E-01	1.99E-03	1.53E-04	1.53E-01	5.10E-03	1.02E-01	1.02E-01
Missed false dump	4.12E-01	5.11E-01	2.47E-03	1.42E-04	7.45E-02	0.00E+00	0.00E+00	0.00E+00

Figure 7.3: *STANDARD A: Relative importance with regard to false dump request triggering*



	Monitor	FEE	BEE	CC	VME	CIBUS	BIC1	BIC2
Missed e. dump	0.00E+00	2.56E-01	1.79E-01	5.13E-02	5.13E-01	2.54E-06	0.00E+00	0.00E+00
Missed false dump	0.00E+00	1.06E-01	1.98E-01	6.33E-02	6.33E-01	3.44E-06	0.00E+00	0.00E+00

Figure 7.4: *STANDARD A: Relative importance with regard to dump request absorbing*

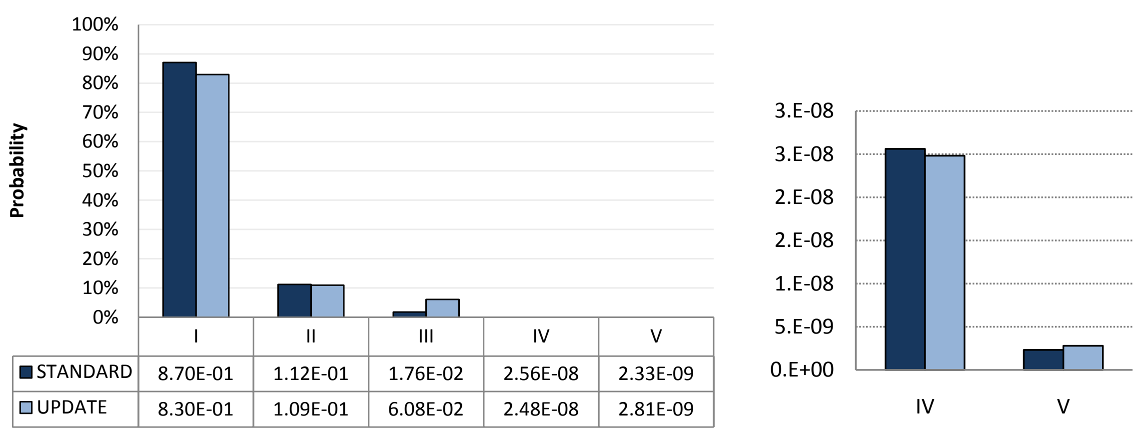
Both with regard to triggering (Fig. 7.3) and absorbing (Fig. 7.4), the importance of the components in *STANDARD B* correspond to *STANDARD A* (Appendix D).

The presented results imply that in order to increase machine safety, the improvement of the VME with regard to the *blind* failure mode is the measure to start with. In terms of beam availability (at the given system demand rate), the starting point is the improvement of the FEEs with regard to the *false* mode. The related sensitivity analysis with decreased failure rates for VMEs and FEEs are presented in Chapter 8.2.

7.3 Case Study: UPDATE

Fig. 7.5 shows the probabilities of the global scenarios for the parameter set *UPDATE C* in comparison to *STANDARD A*. *UPDATE C* yields 83.0% of the missions completed (I), 10.9% ended by emergency beam dumps (II) and 6.1% by false beam dumps (III). The probability of a missed emergency beam dump (IV) is in the range of 1E-8. Compared to *STANDARD A*, this corresponds to an increase of the false beam dumps by about factor 3.5 (missed false beam dumps: 20%) and a slight decrease of the mission completed (5%) and the (missed) emergency beam dumps (2-3%).

The comparison of *UPDATE D* to *C* (increase of system demand rate by factor 3) is consistent with the relations between *STANDARD B* and *A* pointed out in Section 7.2 (Appendix D).



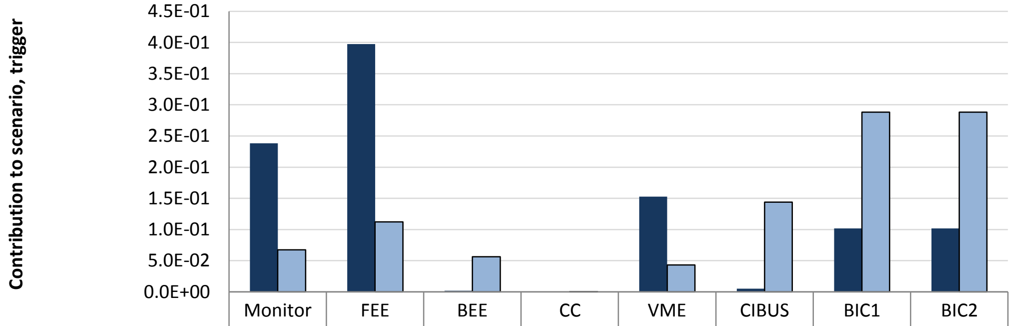
	I	II	III	IV	V
STANDARD	8.70E-01	1.12E-01	1.76E-02	2.56E-08	2.33E-09
UPDATE	8.30E-01	1.09E-01	6.08E-02	2.48E-08	2.81E-09

Figure 7.5: *UPDATE C, STANDARD A: Scenario probabilities*

Given the *UPDATE* failure rates for the *false* mode (Tab. 7.3), the importance of BEEs, CIBUSs and BICs with regard to false request triggers is expected to increase compared to *STANDARD*. This hypothesis is confirmed by the results shown in Figure 7.6.

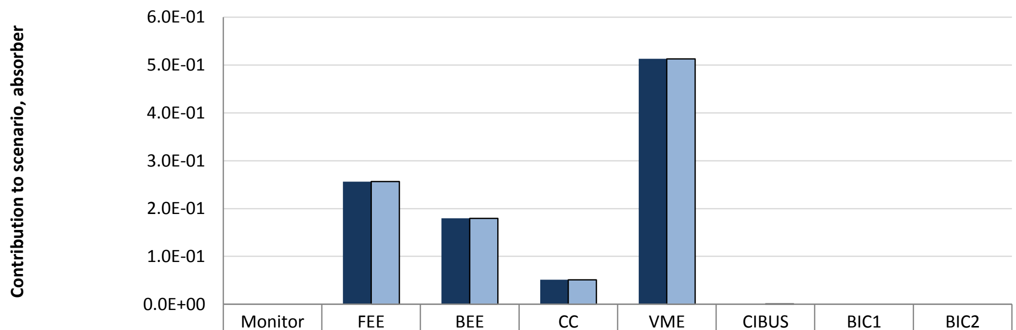
The importance with regard to false beam dumps increases by almost factor 3 for the BICs and 30 for the BEEs and CIBUS, while for the other component it decreases. Nearly 75% of the false beam dumps are triggered by the BIS (CIBUSs and BICs), while in *STANDARD* the Monitors and FEEs contribute about 65%.

Given the *UPDATE* failure rates for the *blind* mode (Tab. 7.3), the importance of the CIBUSs and BICs with regard to dump request absorbing is expected to increase compared to *STANDARD*. This hypothesis is confirmed by the numbers presented in Figure 7.7.



	Monitor	FEE	BEE	CC	VME	CIBUS	BIC1	BIC2
STANDARD false dump	2.39E-01	3.98E-01	1.99E-03	1.53E-04	1.53E-01	5.10E-03	1.02E-01	1.02E-01
UPDATE false dump	6.75E-02	1.12E-01	5.62E-02	4.32E-05	4.32E-02	1.44E-01	2.88E-01	2.88E-01

Figure 7.6: *UPDATE C, STANDARD A: Relative importance with regard to false beam dump triggering*



	Monitor	FEE	BEE	CC	VME	CIBUS	BIC1	BIC2
STANDARD missed e. d.	0.00E+00	2.56E-01	1.79E-01	5.13E-02	5.13E-01	2.54E-06	0.00E+00	0.00E+00
UPDATE missed e. d.	0.00E+00	2.56E-01	1.79E-01	5.13E-02	5.13E-01	2.56E-04	0.00E+00	0.00E+00

Figure 7.7: *UPDATE C, STANDARD A: Relative importance with regard to emergency dump request absorbing*

The importance of the CIBUS with regard to missed emergency dumps increases by two orders of magnitude. Nevertheless, compared to the contribution of the BLMS components, the CIBUSs (and BICs) are negliglible.

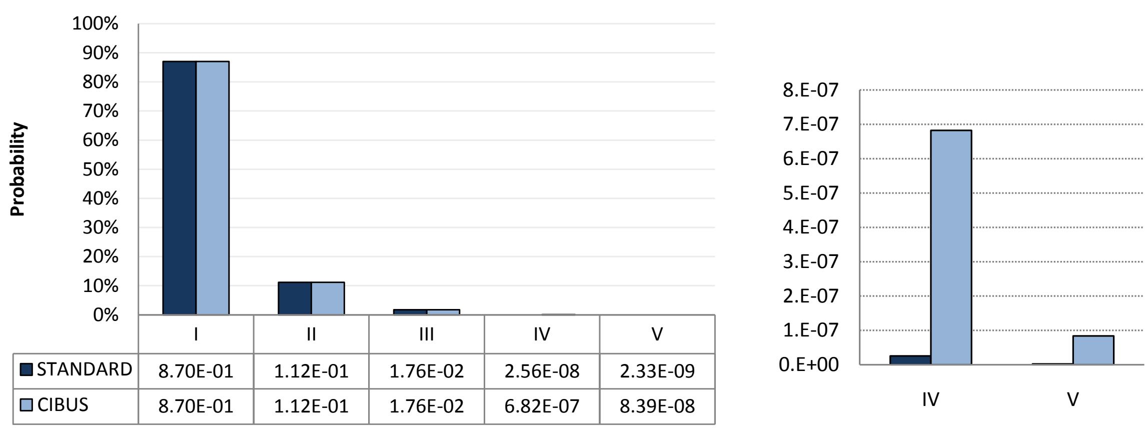
The scenario *missed false beam dump* with its small probability is crucial neither with regard to machine safety nor beam availability. For reason of clarity, it is omitted in the importance considerations. The related results are given in Appendix D.

The importances of the components in *UPDATE D* correspond to *UPDATE C* (Appendix D).

7.4 Case Study: CIBUS

Fig. 7.8 shows the probabilities of the global scenarios for the parameter set *CIBUS E* in comparison to *STANDARD A*. The increase of rate λ^B_{CIBUS} by 7 orders of magnitude (Tab. 7.3) results in an increase of missed emergency dumps (IV) by about factor 25 (missed false beam dumps: 35), with no change in the probability of completed missions (I) and a slight decrease of successful dumps (Scenarios II and III, Appendix D).

The comparison of *CIBUS F* to *E* is consistent with the relations between *STANDARD B* and *A* as pointed out in Section 7.2 (Appendix D).



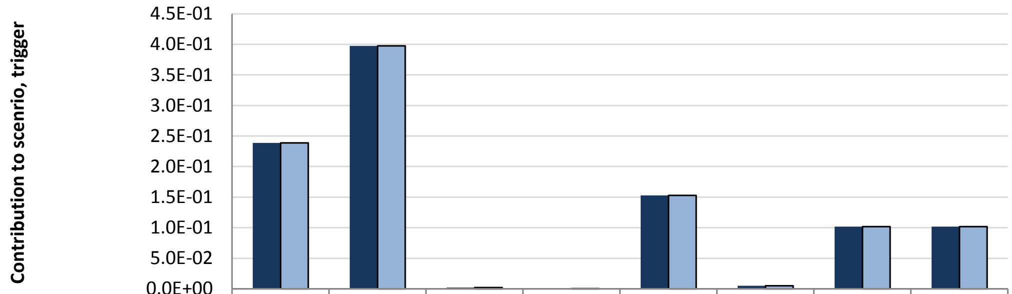
	I	II	III	IV	V
STANDARD	8.70E-01	1.12E-01	1.76E-02	2.56E-08	2.33E-09
CIBUS	8.70E-01	1.12E-01	1.76E-02	6.82E-07	8.39E-08

Figure 7.8: *CIBUS E, STANDARD A: Scenario probabilities*

Given the *CIBUS* failure rates for the false mode corresponding to the *STANDARD* rates, no change of the component importance with regard to false request triggers is to be expected, which is confirmed by the results shown in Figure 7.9 (Appendix D).

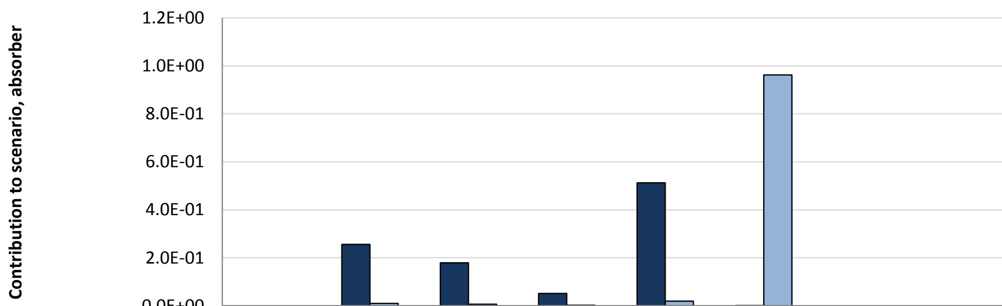
Given the increase of the λ^B_{CIBUS}, the importance of CIBUS with regard to dump request absorbing is expected to increase compared to *STANDARD*. The speculation is confirmed by the results shown in Figure 7.10. More than 95% of the missed emergency dumps are caused by CIBUSs.

The importances of the components in *CIBUS F* correspond to *CIBUS E*, with a deviation in the range of 1E-7 (Appendix D).



	Monitor	FEE	BEE	CC	VME	CIBUS	BIC1	BIC2
STANDARD false dump	2.39E-01	3.98E-01	1.99E-03	1.53E-04	1.53E-01	5.10E-03	1.02E-01	1.02E-01
CIBUS false dump	2.39E-01	3.98E-01	1.99E-03	1.53E-04	1.53E-01	5.10E-03	1.02E-01	1.02E-01

Figure 7.9: *CIBUS E, STANDARD A: Relative importance with regard to false beam dump triggering*



	Monitor	FEE	BEE	CC	VME	CIBUS	BIC1	BIC2
STANDARD missed e. d.	0.00E+00	2.56E-01	1.79E-01	5.13E-02	5.13E-01	2.54E-06	0.00E+00	0.00E+00
CIBUS missed e. d.	0.00E+00	9.62E-03	6.74E-03	1.92E-03	1.93E-02	9.62E-01	0.00E+00	0.00E+00

Figure 7.10: *CIBUS E, STANDARD A: Relative importance with regard to emergency dump request absorbing*

7.5 Case Studies in Comparison

Table 7.4 summarises the results of the case studies. It includes the probabilities of the safety-relevant scenario *missed emergency beam dump* (IV) and the scenario *false beam dump* (III), which is of importance with regard to beam availability.

The different case studies are indicated according to Table 7.1. The underlying input parameters, i.e. system demand rates and component failure rates are given in Tables 7.2 and 7.3.

For the sake of completeness, the table includes the results of case studies *UPDATE D* and *CIBUS F*, which are not explicitly presented above. For a complete listing of all results, the reader is referred to Appendix D.

Table 7.4: *Result comparison across the case studies*

Scenario	Case study					
	A	B	C	D	E	F
III	1.76E-2	1.57E-2	6.08E-2	5.42E-2	1.76E-2	1.57E-2
IV	2.56E-8	6.57E-8	2.48E-8	6.37E-8	6.82E-7	1.75E-6

7.6 Discussion

The case studies confirm the basic feasibility and usability of the approach. Parameter variations are performed within a few minutes and their impact on the system reliability behaviour can be studied comprehensively on the basis of the scenario probabilities and the importance of the component classes. Since the model and thus the calculation incorporates safety and availability aspects both at the same time (scenarios III and IV), their interrelation is addressed easily based on the global scenario probabilities, while the component importance reveals hints with regard to weak points.

On a general basis, the results match the output of the FTA previously performed on the BLMS, even though the underlying models and parameters are not completely corresponding. Guaglio [20] determined a false beam dump trigger to occur during a mission with a probability in the range of 1E-2, which is confirmed by the related case study results (scenarios III and V). The FTA yielded a probability of a beam loss not to be detected by the closest monitor in the range of 1E-6. The figures for missed emergency beams dumps (scenario IV) resulting from the case studies are one to two orders of magnitude lower. It is expected that the inclusion of the 1-out-of-6 redundancy of the monitors (Section 4.2.3) is one of the factors causing this deviation. Direct comparison of importance results in particular is impeded by model deviations with regard to modelling of redundancy and system subdivision.

In consideration of the case study results, one has to bear in mind that they are based on a model covering the BLMS and the BIS only (Chapter 4). As described in Chapter 3, the MPS includes a range of other subsystems which are not taken into account. They are expected the have a lowering effect on the probability of missed emergency beam dumps (due to redundancy) and an increasing effect on the probability of false beam dumps, compared to the results obtained by the case studies.

7.6.1 Principal Interrelations

The results of the case studies reveal the following principal interrelations with regard to parameter variation:

Change of demand rate has an impact on the scenario probabilities but does not affect the relative importance of the components. The increase of the demand rate has a related effect on the probability of (missed) emergency beam dumps (II and IV), while the remaining scenarios, in particular the missions completed (I), encounter an inverse impact due to suppression. The extent of impact seems reasonable in view of the emergency beam dump triggers (defined by x) predominating the false beam dump triggers (λ_{tot}^F) by about one order of magnitude (Fig. 7.2). λ_{tot}^F refers to the product of the failure rates and the number of components, summed up over all component classes.

Change of false rates directly impacts the probabilities of (missed) false beam dumps (III and V), while the remaining scenarios encounter an inverse impact. With the *false* rates defining the second kind of trigger events besides the beam loss specified by the demand rate, this interrelation corresponds to the above mentioned characteristics for the latter. Unlike the case for the demand rate though, the impact regarding successful or missed dump (III and V, Fig. 7.5) differs. The increase of rates λ^F for the BIS components due to their location at the end of the signal path and the redundancy of the BICs (Section 4.2, Fig. 4.6) contributes to successful false beam dumps rather than missed ones. With the CIBUS being only followed by redundant BICs and the BICs being last in the signal path, their (false) beam dump triggering is unlikely (in case of the CIBUS) or impossible (BIC) to be absorbed by following components.

Change of blind rate directly impacts the probabilities of the missed beam dumps (IV and V), while the successful beam dump scenarios (II and III) encounter an inverse impact. The increase of the rates for the *blind* mode causes the result of a beam dump request trigger to shift from successful to missed beam dump. The probability of completed missions in the present case is not affected. In case of rates for the blind mode to the range of those for the false mode or beyond, the completed missions are expected to increase by trend, due to oppressed false beam dump triggers.

Since the case studies are limited with regard to the model structure, the system demand pattern and the range and interrelation of the failure rates, the above observations are to be further addressed and confirmed by extended studies. Some aspects with regard to the failure rate interrelation are investigated by means of sensitivity analysis (Section 8.2). Changes of the system demand pattern are addressed in Section 9.1.

The exponential distribution underlying the model and the characteristics of the scenarios allow for approximations of parts of the results using basic probability calculus.The probability of a beam dump trigger to occur (covering scenarios II, III, IV and V) is approximated by the cumulative exponential distribution function

$$\widehat{P}_{II,III,IV,V}(t) = 1 - e^{-\lambda_{tot} \cdot t} \tag{7.4}$$

where

$$\lambda_{tot} = x + \sum_{i=1}^{8} \left(n_i \cdot \lambda_i^F \right) \tag{7.5}$$

and t = 12 h; n_i = number of components i (Table 4.5); λ_i^F = *false* failure rate of component i (Table 4.7); x = system demand rate [61].

Table 7.5 shows the relative error of the approximation for *STANDARD* and *UPDATE*. The error arises from the exclusion of the second failure mode (*blind*). Eq. 7.4 does not include the component-internal suppression of *false* failures by the *blind* failure mode (Type I, Section 4.1.4). The error is therefore expected to increase if the *blind* failure rates are increased to the order of magnitude of the related *false* failure rate or beyond.

Table 7.5: *Estimate of probability of early-ended mission*

	STANDARD A	STANDARD B	UPDATE C	UPDATE D
P_{II}	1.12E-01	3.00E-01	1.09E-01	2.93E-01
P_{III}	1.76E-02	1.57E-02	6.08E-02	5.42E-02
P_{IV}	2.56E-08	6.57E-08	2.48E-08	6.37E-08
P_V	2.33E-09	1.99E-09	2.81E-09	2.40E-09
$P_{II,III,IV,V}$	1.30E-01	3.15E-01	1.70E-01	3.47E-01
$\widehat{P}_{II,III,IV,V}$	1.30E-01	3.15E-01	1.70E-01	3.47E-01
|Rel. Error|	**2.23E-08**	**7.20E-09**	**1.63E-08**	**6.28E-09**

An estimate of the component importance with regard to false beam dump triggering is obtained by ignoring both *blind* failures and system demand. Using the first-order approximation of the exponential function based on its Taylor series (at position $x_0 = 0$, [62]), the probability of a false beam dump trigger to occur is approximated by

$$\widehat{P}_{III,V}(t) = 1 - e^{-\lambda_{tot}^F \cdot t} \approx \lambda_{tot}^F \cdot t. \tag{7.6}$$

where

$$\lambda_{tot}^F = \sum_{i=1}^{8} \left(n_i \cdot \lambda_i^F \right). \tag{7.7}$$

According to Lagrange [62], the error of the first-order approximation of the exponential function, R_1, can be estimated by

$$R_1 \leq 0.5 \cdot \lambda^2 \cdot t^2. \tag{7.8}$$

For $\lambda_{tot}^F = 1.57E-3$ (STANDARD) and $t = 12$, R_1 is less than 1.8E-4.

The independent consideration of component class *i* yields

$$\widehat{P}_{iIII,V}(t) = 1 - e^{-n_i \cdot \lambda_i^F \cdot t} \approx n_i \cdot \lambda_i^F \cdot t \tag{7.9}$$

as the estimate of the probability of component class i triggering a false dump request. The importance (i.e. relative probability) of component class i with regard to the triggering of false dump requests is thus estimated by

$$\widehat{I}_{iIII,V} = \widehat{P}_{iIII,V} / \widehat{P}_{III,V}. \tag{7.10}$$

The relative error of the estimation compared to the analytical description is shown in Figure 7.11 and Tables 7.6 and 7.7 for *STANDARD A* and *UPDATE C*.

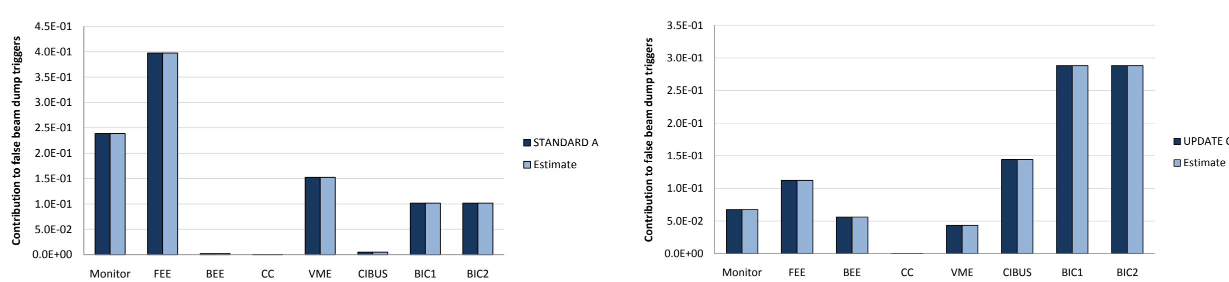


Figure 7.11: *STANDARD A, UPDATE C: Importance estimates ($\widehat{I}_{iIII,V}$) for false beam dump triggering*

The importance estimates show satisfying accuracy. The resulting relative error of $P_{III,V}(t)$ in the range of 1E-2 results from the exclusion of the system demand and

Table 7.6: *STANDARD A: Importance estimates for false beam dump triggering*

	Monitor	**FEE**	**BEE**	**CC**	**VME**	**CIBUS**	**BIC1**	**BIC2**	$P_{III,V}(t)$
$I_{i,III,V}$	2.39E-01	3.98E-01	1.99E-03	1.53E-04	1.53E-01	5.10E-03	1.02E-01	1.02E-01	1.76E-02
$\widehat{I}_{iIII,V}$	2.39E-01	3.98E-01	1.99E-03	1.53E-04	1.53E-01	5.10E-03	1.02E-01	1.02E-01	1.88E-02
|Rel. Error|	**4.14E-07**	**1.13E-07**	**1.66E-07**	**1.66E-07**	**1.13E-07**	**1.72E-07**	**1.72E-07**	**1.72E-07**	**7.10E-02**

Table 7.7: *UPDATE C: Importance estimates for false beam dump triggering*

	Monitor	**FEE**	**BEE**	**CC**	**VME**	**CIBUS**	**BIC1**	**BIC2**	$P_{III,V}(t)$
$I_{i,III,V}$	6.75E-02	1.12E-01	5.62E-02	4.32E-05	4.32E-02	1.44E-01	2.88E-01	2.88E-01	6.08E-02
$\widehat{I}_{iIII,V}$	6.75E-02	1.12E-01	5.62E-02	4.32E-05	4.32E-02	1.44E-01	2.88E-01	2.88E-01	6.66E-02
|Rel. Error|	**5.33E-07**	**9.53E-09**	**4.28E-08**	**4.24E-08**	**9.54E-09**	**4.85E-08**	**4.86E-08**	**4.86E-08**	**9.62E-02**

its suppressing effect on the triggering of false dump requests (Type II, Section 4.1.4). The estimate of $P_{III,V}(t)$ consequentially turns out to be higher than the analytical value. The division included in the calculation of the (relative) importances eliminates this error to a certain extent, which explains the better accuracy of the estimated importances $\widehat{I}_{iIII,V}$. Again, the error of the importance estimates is expected to increase when the *blind* failure rates are of the order of magnitude of the related *false* failure rate or beyond. The opposite, i.e. a decrease, should be observed for the error of the $P_{III,V}(t)$ estimate in case of a total *false* failure rate λ_{tot}^{F} in the order of magnitude of the system demand rate or beyond.

Since the absorbing of beam dump requests is not a single event but based on the conjunction of trigger events and *blind* failures and depends on the system structure, the presented estimation using the number of components and the failure rates is not applicable to the importance with regard to absorbing.

Chapter 8

Verification and Sensitivity Analysis

This chapter introduces the verification options for the use of the modeling approach and applies them to the results of the case studies. A range of sensitivity analyses are presented in the second section (Fig. 8.1).

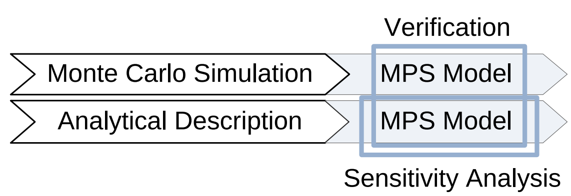


Figure 8.1: *Results: Verification and sensitivity analysis*

8.1 Verification

Given a basic model according to the presented modeling approach, errors leading to faulty results can arise in each step of the intermediate procedure:

- Model description
- Model implementation
- Calculation

To that effect, accurate results on the MPS model using the analytical description (Chapter 6) require:

- Analytical description (Section 6.1) matching the basic model
- Appropriate implementation of the MPS model
- Adequate numerical accuracy of the integration by the software

Accurate results on the MPS model using Monte Carlo simulation (Chapter 5) require:

- Monte Carlo approach (Section 5.1) matching the basic model
- Appropriate implementation of the MPS model
- Adequate accuracy of the simulation by the software and correct analysis of the simulation data

The single steps cannot be verified independently. Verification relies on the resulting probability figures only.

8.1.1 Concept

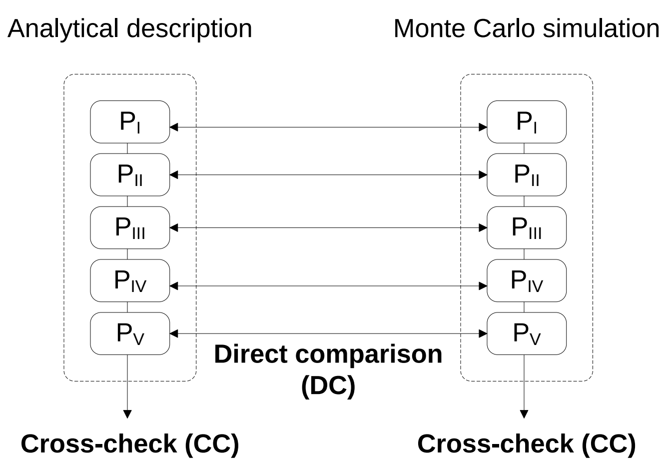


Figure 8.2: *Concept of verification*

The concept of verification is illustrated in Fig. 8.2. The two basic options to verify the results of the case studies are

1. Cross-check (CC) across the different scenarios (I-V) and

2. Direct comparison (DC) across different ways of implementation (i.e. analytical description and Monte Carlo simulation).

The cross-check is based on the complementary character of the global scenarios. The scenarios are mutually exclusive and their probabilities add up to 1:

$$1 - P_I(t) - P_{II}(t) - P_{III}(t) - P_{IV}(t) - P_V(t) \equiv 0 \tag{8.1}$$

Equation 8.1 reflects the fact that any mission results in one of the five scenarios by definition. It is either completed (I) or encountering an emergency or false dump request which in turn is either successful leading to a beam dump (II and III) or missed (IV, V).

The cross-check provides the verification whether the implemention in itself is correct. It does not provide evidence that the implemention represents the model of interest. Furthermore, the cross-check does not necessarily reveal systematic errors with regard to the implementation and calculation.

The mentioned restrictions are overcome by the second verification option. Provided that two different implementations of a model are available, the direct comparison of the results provides a comprehensive verification.

8.1.2 Results

Table 8.1 summarizes the verification of case study *STANDARD A* (Section 7.2). Related simulation results based on 10^5 and 10^7 missions have been published previously [61, 63]. For the benefit of a more advanced verification, 10^8 simulated missions are used here for comparison, requiring a simulation time of several months on a multiprocessor server (Appendix B). The case study results (based on the analytical description) are rounded off accordingly.

Both the results of the case study and the related simulation satisfy Equation 8.1. The error in the order of 1E-20 for the case study is beyond the precision of the calculation (i.e. 15 decimal places, Section 7.2). Tests performed with precision set up to 45 (at the expense of calculation time) yield matching accuracies. The relative freqencies resulting from the simulations satisfy the cross-check strictly and prove that the implementation and the data analysis are correct. Evidence for the accuracy of the simulation results is not provided by the cross-check.

The direct comparison of the analytical and simulation approach shows very good agreement for the frequent scenarios. The relative error of scenario IV is of limited

Table 8.1: *STANDARD A: Verification of results (1E8 missions)*

Scenario	Analytical Description	Simulation	|Rel. Error| (DC)
I	8.7036979E-1	8.7037066E-1	1.00E-6
II	1.1204224E-1	1.1204946E-1	6.4441E-5
III	1.758794E-2	1.757983E-2	4.6132E-4
IV	3E-8	4E-8	2.5000000E-1
V	(1E-9)	1E-8	NA
CC	0 (1E-20)	0	

value and of scenario V is not available because 10^8 missions are below an adequate quantity of simulation data to provide significant results.

Compared to the results based on 10^7 [63], the relative error of scenarios I and II is one order of magnitude smaller, as can be expected. Surprisingly though, for scenario III, the relative error is twice as big. This is caused by a (missing) suppression effect of Type II (Section 4.1.4) in the 10^7 mission simulation due to too few emergency beam dump triggers. The frequency of scenario II is significantly below the calculated value [63]. This involves a decreased suppression effect on false beam dump triggers resulting in an increase of the frequency of scenario III, which therefore misleadingly comes closer to the analytical result. This explanation is confirmed by considering the combined probabilities of a beam dump triggering (scenarios II-V). The related relative errors is two orders of magnitudes smaller compared to the 10^7 mission simulation. This confirms the expected increase of agreement between the results of calculations and simulations with an increased number of simulated missions.

8.1.3 Discussion

The results of the case studies are successfully verified by the cross-check across the different scenarios and by the direct comparison across the two ways of implementation. Besides the verification purpose, the cross-check proved to be a very useful instrument for debugging during the implementation of both the analytical description and the Monte-Carlo approach. The implementation of the analytical description is error-prone particularly in terms of missing subscenarios or terms in the equations. In the Monte Carlo approach one of the main error-prone parts of the implementation is the script for the analysis of the logged data. However, as stated above, certain systematic errors are not detected by the cross-check. For the analytical description in particular it has to be noted that the cross-check is meaningful only if the equations of the different scenarios are implemented individually, i.e. if the scenario probabilities are calculated independently and not deduced from each other. In case of individual equations (or simulation data) for scenarios *emergency beam dump triggered* and *false beam dump triggered*, the cross-check can be further divided into three checks [63]. This faciliates the isolation of error locations in terms of the scenario.

The direct comparison of the results across different implementations provides a more comprehensive verification which includes the model description (i.e. whether the implemented description represents the model of interest) as well as the completion of the calculations (simulation). In the present case however, the use of direct comparison with simulation results is limited due to the number of simulated missions and the simulation time required to cover the rare scenarios (IV, V). Given the limited number of 10^8 missions, the probabilities of the rare scenarios therefore have to be verified indirectly, based on the comparison of the frequent scenarios and the cross-check. For small models and rough verification, a less elaborate implementation of the Monte-Carlo approach can be used. During the development of the analytical description based on a few-component model, the related Monte-Carlo simulation was implemented using Excel for the purpose of a quick verification (Section 5.1).

A more elaborate option for the verification within one implementation is the analysis of the results in terms of convergence. If there is only one implementation of a model available, the proof of convergence (against increased calculation precision and increased amount of simulated mission respectively) is the means to supplement the cross-check.

The current implementation of the Monte-Carlo simulation and the script for its data analysis allows for a direct comparison of the component importances in terms of false beam dump request triggering. However, this option is less relevant since a rough verification is easily performed by means of estimates (Section 7.6.1).

It is to be noted that the described verification refers to the given model. Its validation, i.e. the question whether the model with its simplifications, assumptions and input parameters reflects the real system accurately [64], has to be assessed by continuous comparison with operational experience and, if required, related model adaptations.

8.2 Sensitivity Analysis

The basic input parameters of the MPS model are the system demand rate, the component failure rates and the observation time (Chapter 4). The presented case studies

(Chapter 7) include variations of the former two. This section presents the results of more comprehensive parameter variation studies including observation time, based on the analytical description of the MPS model used for the case studies. The purpose is to investigate the dependence of the outcome on the input parameters in general and to further address some interrelations revealed by the case studies in particular.

8.2.1 Concept

The different variation studies are based on the parameter set of case study *STANDARD A* (Tab. 7.2 and 7.3), with separate parameter variation according to the description given in Table 8.2.

Table 8.2: *Parameter variation studies. O(2): Two orders of magnitude*

Parameter	**Study**	**Description**
System demand rate $[h^{-1}]$	SD	Rate x varying from 1E-4 to 1E-1
Component failure rates	BLIND	Rates λ^B varying as a block from O(2) below to O(2) above rates λ^F of *STANDARD A*
	OPTIM	Rate λ^F_{FEE} (λ^B_{VME}) varying from O(2) below to O(2) above the figure of *STANDARD A*
Observation time $[h]$	OT	Time t_m varying from 0 to 100

Study *SD* allows for further proving the dependence on the system demand rate (Section 7.6.1), as well as for studying the suppression effect across different components (Type II, Section 4.1.4). Study *OT* reveals the dependency of the scenario probabilities on the observation time (i.e. mission length).

The studies regarding the component failure rates serve multiple purposes. While providing evidence for the previously observed correlations regarding the *false* rates (*OPTIM*) and *blind* rates (*BLIND*, *OPTIM*), they allow to address more specific aspects. The component-internal suppression effect between the two failure modes *false* and *blind* (Section 4.1.4) is addressed by study *BLIND*. *OPTIM* yields evidence on how the system performance is expected to improve through optimization of the components with the highest importance regarding false beam dump request triggers (FEE) and request absorbing (VME) (Chapter 7).

8.2.2 Results

For the sake of clarity, the sensitivity analysis is limited to the scenario probabilities. The detailed figures of the parameter variation studies are given in Appendix D.

8.2.2.1 Study: SD

The variation of the system demand rate confirms the basic correlation observed in the case studies (Section 7.6.1). Figure 8.3 shows the scenario probabilities resulting from a demand rate variation from 0.5E-2 and to 3.5E-2h^{-1}, covering the range of the demand rates used in the case studies (1E-2 and 3E-2h^{-1}, Tab. 7.2). An increased demand rate (by factor 7) is directly reflected in an increased probability of scenarios II (factor 6) and IV (factor 5.5), while the remaining scenario probabilities decrease. The increase of emergency beam dump triggers (II, IV) mainly occurs at the expense of completed missions (I). The related suppression of false beam dump triggers (III, IV) is less pronounced but still visible in decreasing probabilities (decrease by 15%).

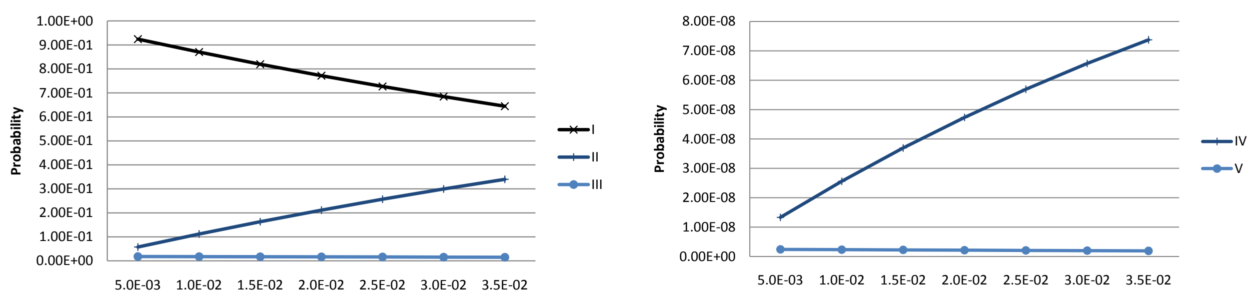


Figure 8.3: *SD: Scenario probabilities against system demand rate x in the range of 1E-2 h^{-1}*

Figure 8.4 shows the scenario probabilities (I, II, III) with the system demand rate varying across three orders of magnitude. The increase of the demand rate from 1E-4 to 1E-3 h^{-1} (left) increases the probability of scenario II by one order of magnitude (from 1.19E-3 to 1.18E-2). However, the absolute impact on the scenario probabilities is marginal and an emergency beam dump (II) remains less likely than a false beam dump (III, 1.85E-2). The increase of the demand rate from 1E-3 to 1E-2h^{-1} (middle) increases the probability of scenario II by factor 9.5 (from 1.18E-2 to 1.12E-1) entailing the decrease of scenario I by 10%. The range between 1E-3 and 4E-3h^{-1} features the intersection point where emergency beam dumps (II) becomes more likely than false beam dumps (III). Given a rate λ_{tot}^{F} (Eq. 7.7) of 1.5E-3, this observation is reasonable.

The increase of the demand rate from 1E-2 to 1E-1h^{-1} (right) results in the most distinct absolute increase of emergency beam dumps (II) from 11.2% to 69.4% at the expense of completed missions (I).

It is to be noted that the system demand rates used for the case studies (1E-2 and 3E-2h^{-1}) lie in the range which the scenario probabilities react most sensitive to (cp. gradient in Fig. 8.4, right).

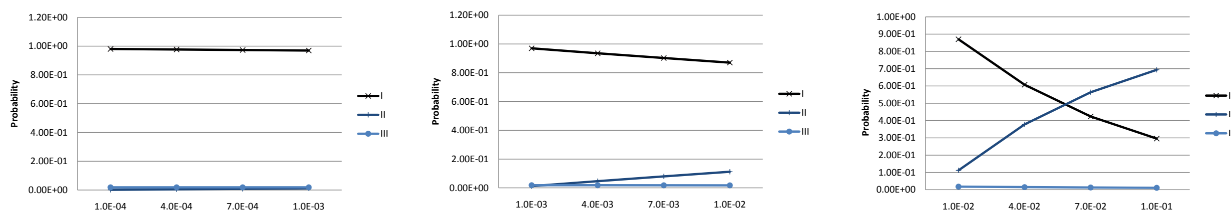


Figure 8.4: *SD: Scenario probabilities I, II, III against system demand rate x across three orders of magnitude*

8.2.2.2 Study: BLIND

This study bases upon the variation of rates λ_i^B of each component class i around the respective rates λ_i^F (Fig. 7.3, STANDARD A):

- **BF_mO2:** Rates λ_i^B two orders of magnitude below rates λ_i^F
- **BF_mO1:** Rates λ_i^B one order of magnitude below rates λ_i^F
- **BF_mO0:** Rates λ_i^B equal to rates λ_i^F
- **BF_pO1:** Rates λ_i^B one order of magnitude above rates λ_i^F
- **BF_pO2:** Rates λ_i^B two orders of magnitude above rates λ_i^F

The results of the study confirm the previously observed correlation regarding the change of the *blind* rates (Chapter 7.6.1) and the supposition regarding the component-internal suppression across the two failure modes. The increase of rates λ_i^B increases the probability of scenarios IV and V (Fig. 8.5, right), with most distinct absolute impact in case of rates λ_i^B above rates λ_i^F. Each increase by one order of magnitude results in probabilities increased by factor 10 (four orders of magnitude in total, up to a probability of missed emergency beam dump of 0.14%).

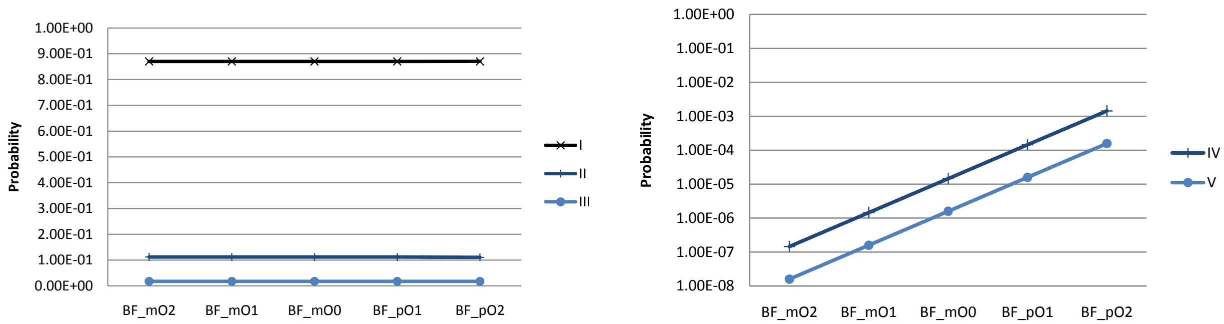


Figure 8.5: *BLIND: Scenario probabilities against rates λ^B in the range of λ^F*

The inverse (decreasing) impact on the successful beam dump scenarios (II,III), due to the shifting from successful to missed beam dump request, is observable at closer inspection (Fig. 8.6, middle and right). The total absolute decrease comes to 1.440E-3 (II) and 1.983E-4 (III). One notes that the latter overcompensates the increase of scenario V (1.575E-4). This is due to the component-internal supression across the two failure modes (Type I). The increase of the *blind* failures not only causes the global shift from successful to missed beam dump requests. It also (within the components) suppresses the occurrence of *false* failures, resulting in a global decrease of false beam dump requests. Fewer false beam dump requests in turn increase the probability of emergency beam dump requests due to an inverse suppression effect across the trigger types (Type II suppression). This effect however is less pronounced, the decrease of emergency beam dumps (II, by 1.440E-3) only slightly undercompensates the increase of missed emergency beam dumps (IV, by 1.441E-3). The suppression of false beam dump requests is predominant, resulting in a related increase of completed missions (I, by 3.915E-5, Fig. 8.6, left).

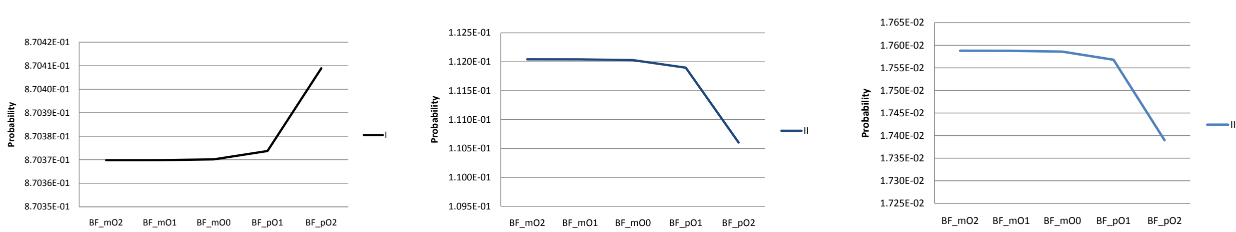


Figure 8.6: *BLIND: Detailed scenario probabilities I, II, III*

It is to be noted that the mentioned correlations result in significant absolute impact on the probabilities of the frequent scenarios(I, II, III) in case of rates λ_i^B above rates λ_i^F only, while for scenarios IV and V the relative impact of increased λ_i^B is distinct.

8.2.2.3 Study: OPTIM

This study covers the variation of rates λ^F_{FEE} and λ^B_{VME} around the respective values of case study *STANDARD A* (Tab. 7.3), individually (Fig. 8.7 and 8.8) and simultaneously (Fig. 8.9).

Figure 8.7 shows the results of the (individual) variation of rate λ^F_{FEE}. It confirms the previously observed basic correlation (Section 7.6.1). The increase of rate λ^F_{FEE} increases the probability of false beam dumps (III) and missed false beam dumps (V), while the probabilities of the remaining scenarios decrease. The decrease of emergency beam dumps (II) (missed emergency beam dumps, IV) results from the suppression of emergency beam dump triggers by false beam dump triggers (inverse effect has been described in study SD, Section 8.2.2.1). The increase of rate λ^F_{FEE} by one order of magnitude brings the probability of false beam dumps (III) (missed false beam dumps, V) close to the emergency beam dumps (II) (missed emergency beam dumps, IV) and beyond with rate further increased. This result is in accordance with expectations, given rate λ^F_{tot} (Eq. 7.7) based on a rate λ^F_{FEE} of 1E-5h^{-1} coming up to 7E-3h^{-1} and as such close to the system demand rate of 1E-2h^{-1}. The corresponding effect in terms of the system demand rate variation, i.e. with a demand rate lowered by one order of magnitude to 1E-2h^{-1} is hidden in Figure 8.4. It would be visible on a logarithmic scale.

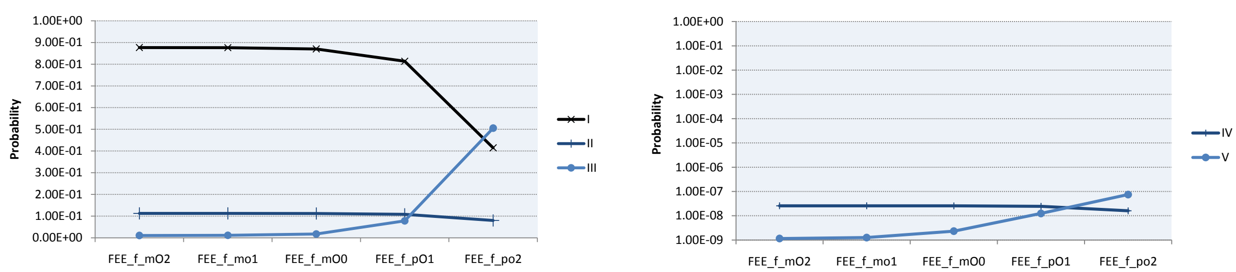


Figure 8.7: *OPTIM: Scenario probabilities against rate λ^F_{FEE} around STANDARD A value (FEE_f_mO0)*

Figure 8.8 shows the results of the (individual) variation of rate λ^F_{FEE}. The basic correlations correspond to the findings of study BLIND (Section 8.2.2.2, note the logarithmic scale in the right plot) and as such confirms the interrelations observed in the case studies.

Figure 8.9 shows the results of a simultaneous variation of both rate λ^F_{FEE} and λ^B_{VME}. It includes two optimisation scenarios with the rates decreased by one (VMEfEE_bf_mO1) and two (VMEfEE_bf_mO2) orders of magnitude:

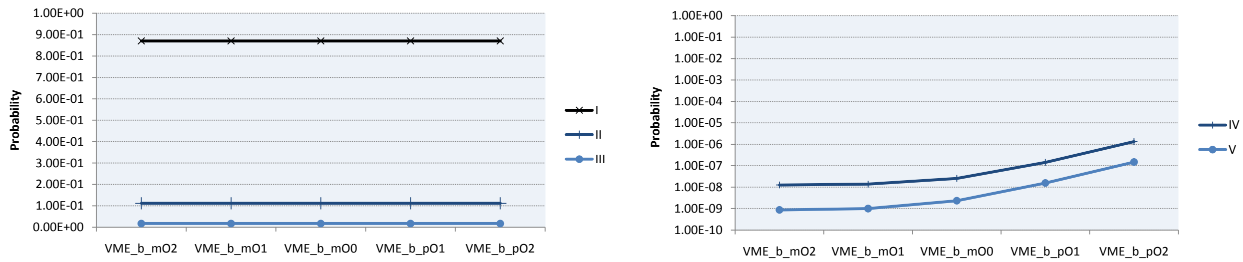


Figure 8.8: *OPTIM: Scenario probabilities against rate λ^B_{VME} around STANDARD A value (VME_b_mO0)*

- **VMEfEE_bf_mO2:** Rate λ^F_{FEE}=1E-8h^{-1} and rate λ^B_{VME}=1E-10h^{-1}
- **VMEfEE_bf_mO1:** Rate λ^F_{FEE}=1E-7h^{-1} and rate λ^B_{VME}=1E-9h^{-1}
- **VMEfEE_bf_mO0:** Rate λ^F_{FEE}=1E-6h^{-1} and rate λ^B_{VME}=1E-8h^{-1} (STANDARD A)
- **VMEfEE_bf_pO1:** Rate λ^F_{FEE}=1E-5h^{-1} and rate λ^B_{VME}=1E-7h^{-1}
- **VMEfEE_bf_pO2:** Rate λ^F_{FEE}=1E-4h^{-1} and rate λ^B_{VME}=1E-6h^{-1}

The basic correlations resemble the results of the λ^F_{FEE} variation (Fig. 8.7). For the purpose of direct comparison and deductions regarding system optimisation, Table 8.3 compiles the results based on rates λ^F_{FEE} and λ^B_{VME} reduced by one order of magnitude.

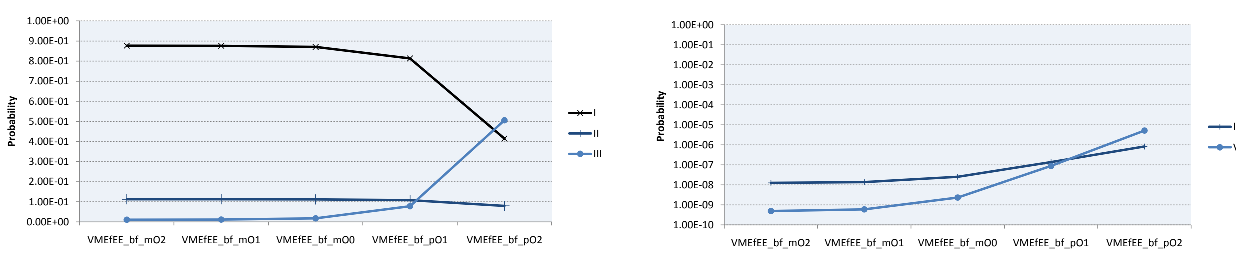


Figure 8.9: *OPTIM: Scenario probabilities against rates λ^F_{FEE} and λ^B_{VME} around the STANDARD A values*

The decrease of rate λ^F_{FEE} by one order of magnitude (FEE_f_mO1) results in a decrease of the probability of false beam dumps (III) by 35%, while the decrease of rate λ^B_{VME} (VME_b_mO1) yields a decrease of missed emergency beam dumps (IV) by 45%. The simultaneous optimisation (VMEfEE_bf_mO1) consequentially includes both benefits.

Table 8.3: *OPTIM: Scenario probabilities with component optimisation. FEE_f_mO1 (VME_b_mO1): rate λ^F_{FEE} (λ^B_{VME}) decreased by O(1), VMEfEE_bf_mO1: λ^F_{FEE} and λ^B_{VME} both decreased by O(1)*

	STANDARD A	FEE_f_mO1	VME_b_mO1	VMEfEE_bf_mO1
I	8.70E-01	8.76E-01 (+0.7%)	8.70E-01 (NA)	8.76E-01 (+0.7%)
II	1.12E-01	1.12E-01 (+0.3%)	1.12E-01 (NA)	1.12E-01 (+0.3%)
III	1.76E-02	1.13E-02 (-35%)	1.76E-02 (NA)	1.13E-02 (-35%)
IV	2.56E-08	2.57E-08 (+0.4%)	1.38E-08 (-45%)	1.39E-08 (-45%)
V	2.33E-09	1.26E-09 (-45%)	1.00E-09 (-55%)	5.98E-10 (-75%)

8.2.2.4 Study: OT

Both Figure 8.10 and 8.11 show the expected interrelation that with increasing observation time t_m the probabilities of the beam dump scenarios (II, III, IV, V) increase, at the expense of completed missions (I). For small t_m in the range of 12 hours, the dependence is linear for scenarios I, II and III (Fig. 8.10), left), which is in accordance with the first-order approximation of the cumulative exponential distribution function based on Taylor series (cp. Eq. 7.6). The linear interrelation does not apply to scenarios IV, V (Fig. 8.10). While scenarios *emergency beam dump* (II) and *false beam dump* (III) depend on a single (trigger) event, scenarios *missed emergency beam dump* (IV) and *missed false beam dump* (V) involve the combination of beam dump triggering failures and *blind* failures, which is expected to be the cause for the non-linear interrelation.

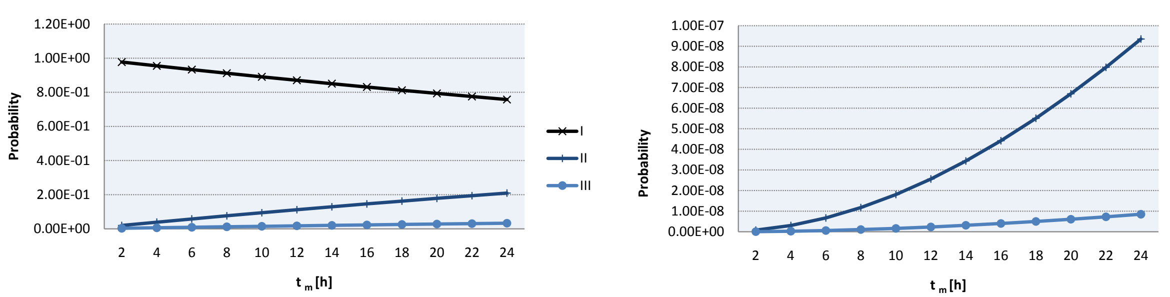


Figure 8.10: *OT: Scenario probabilities against observation time t_m from 0 to 24 h*

Table 8.4 presents some of the results underlying Figure 8.10. A reduction of the observation time from 12 to 6 hours reduces the probability of false beam dumps (III) by 50% and of missed emergency beam dumps (IV) by 75%, increasing the fraction of completed missions (I) from 89% to 93%. A decrease of the observation time by 2 hours entails the decrease of false beam dumps (III) by 15% and of missed emergency beam dumps (IV) by 30%. With 400 missions per year, this results in one false beam dump less per year (6 instead of 7) and a missed emergency beam dump to be expected every 140,000 instead of 100,000 years.

Table 8.4: *OT: Scenario probabilities against t_m from 6 to 14 h*

	6	8	10	12	14
I	9.33E-01(+ 7%)	9.12E-01(+ 5%)	8.91E-01(+ 2%)	**8.70E-01**	8.50E-01(- 2%)
II	5.80E-02(-48%)	7.64E-02(-32%)	9.44E-02(-16%)	**1.12E-01**	1.29E-01(+15%)
III	9.10E-03(-48%)	1.20E-02(-32%)	1.48E-02(-16%)	**1.76E-02**	2.03E-02(+15%)
IV	6.70E-09(-74%)	1.17E-08(-54%)	1.81E-08(-29%)	**2.56E-08**	3.43E-08(+34%)
V	6.09E-10(-74%)	1.07E-09(-54%)	1.64E-09(-29%)	**2.33E-09**	3.12E-09(+34%)

Figure 8.11 shows the scenario probabilities against t_m on a slightly larger scale from 0 to 100 h. The linear dependence observed on the small scale for scenarios I, II and III (Fig. 8.10, left) evolves into a rather exponential dependence. The probability of emergency beam dumps (II) still increases significantly and for t_m beyond 70 h exceeds the probability of completed missions (I).

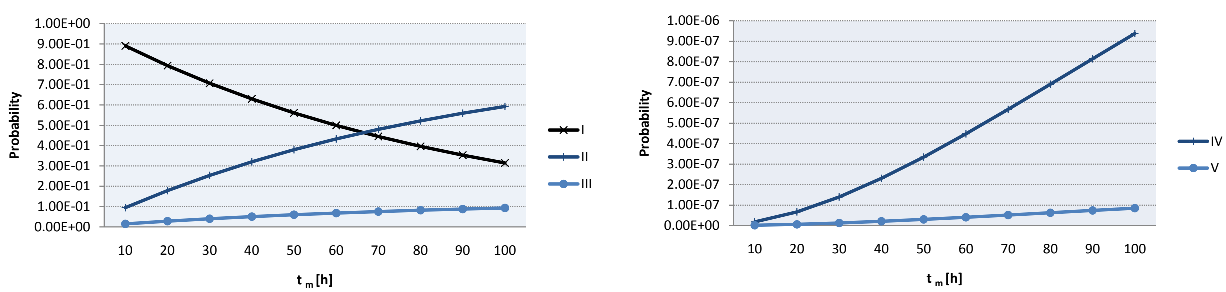


Figure 8.11: *OT: Scenario probabilities against observation time t_m from 0 to 100 h*

8.2.3 Discussion

The sensitivity analyses based on the analytical description of the MPS model confirm the interrelations observed in the case studies and show no unexpected dependence of the results on the input parameters. The suppression effect across failure modes and triggering events (Section 4.1.4), can be followed in detail. Most notably, the sensitivity analyses identify the input parameters that require particular attention due to their distinct impact on the results and show the effect to be expected by optimisation measures such as the improvement of components with regard to reliability.

The findings of the sensitivity analyses can be summarised as follows:

- **SD:** The study reveals a high sensitivity on the system demand rate. Due to its order of magnitude (compared to the component failure rates), a change of the demand rate has a significant absolute impact on the *emergency beam dump* as well as the safety-relevant *missed emergency beam dump* scenario. The system demand rates used in the case studies lie in a range of nearly linear interrelation to these scenario probabilities. For the system demand rate between 5E-3 and 4E-2h^{-1}, an increase of the rate by factor 2 increases the probabilities by factors of around 1.8.

- **BLIND:** The blockwise increase of the component *blind* rates reveals a linear interrelation to the *missed emergency (false) beam dump* scenario, an increase of the rates by one order of magnitude increases the scenario probabilities by about factor 10.

- **OPTIM:** The increase of rate λ^F_{FEE} by one order of magnitude results in a *false beam dump* probability of almost 8%, approaching the *emergency beam dump* probability of 11%. The decrease of rate λ^F_{FEE} by one order of magnitude on the other hand yields a decrease of the false beam dump probability by about one-third, while a decrease of rate λ^B_{VME} by one order of magnitude almost halves the probability of missed emergency beam dumps.

- **OT:** In the range of the observation time used in the case studies (12 hours), the probabilities for the scenarios *emergency beam dump* and *false beam dump* show a linear dependency on the observation time. A decrease of the observation time by 2 hours from 12 to 10 hours results in a decrease of the false beam dumps by 16%. The probabilities of the *missed emergency (false) beam dump* scenario show non-linear dependency. The decrease of the observation time by 2 hours results in a decrease of the missed emergency beam dumps by about one-third.

In addition to the presented sensitivity analyses, two extreme scenarios with regard to the fail-safe measures included in the MPS, i.e. the failure mode *false* in the model, are addressed. The scenarios assume the absence of fail-safe measures, i.e. the presence of *blind* failures only. Scenario *BLINDBpF* acts on a pessimistic assumption, assigning the omitted *false* failures to failure mode *blind*. Scenario *BLINDB* represents the optimistic extreme, omitting the *false* failures without substitution. With regard to the component failure rates, the addressed scenarios (with *STANDARD A* for comparison) are defined as follows:

- **STANDARD A:** failure modes *blind* and *false* with rates λ^B_i and λ^F_i according to Table 7.3

- **BLINDBpF:** failure mode *blind* with rates $\lambda_i^{B,new}$= λ_i^B + λ_i^F, $\lambda_i^{F,new} = 0$
- **BLINDB:** failure mode *blind* with rates $\lambda_i^{B,new}$= λ_i^B, $\lambda_i^{F,new} = 0$

The resulting scenario probabilities are shown in Table 8.5. Obviously, scenario III, *(false beam dump*, disappears for the benefit of scenario I (*completed mission*). The pessimistic scenario *BLINDBpF* additionally entails an increase of the probability of missed emergency dumps by three orders of magnitudes up to 1E-5 (relating to one missed emergency beam dump in about 170 years), which corresponds to the related results of study *BLIND*.

Table 8.5: *No fail-safe measures: Scenario probabilities with omitted failure mode false*

Scenario	STANDARD A	BlindBpF	BlindB
I	8.70E-01	8.87E-01	8.87E-01
II	1.12E-01	1.13E-01	1.13E-01
III	1.76E-02	0	0
IV	2.56E-08	1.47E-05	2.60E-08
V	2.33E-09	0	0

Chapter 9

Model Extensions

The MPS model, which was introduced in Chapter 4 and underlies the case studies and sensitivity analysis, includes a series of simplifying assumptions (Section 4.3.1). Two of the major limitations are the simplified system demand pattern and the exclusion of common cause failures. The simulation approach allows for a straightforward implementation of model extensions to overcome these limitations. This chapter presents the implementation of a more sophisticated system demand pattern, combined with the masking of monitors, and common cause failures (Fig. 9.1).

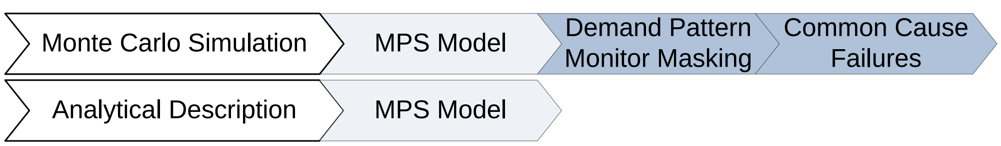


Figure 9.1: *Model extensions: Inclusion of weighted demand pattern, monitor masking and common cause failures*

9.1 Weighted Demand Pattern and Monitor Masking

In the LHC, beam losses are not equally distributed along a magnet. It is assumed that the first and second monitor of a magnet (downstream) encounter more beam loss than the third [65]. Furthermore, a beam loss event normally is not limited to one magnet but affects several magnets at different locations at the same time.

The more than 3000 beam loss monitors must ensure that each beam loss event is covered by at least one monitor. Since a beam loss event is often spread across several monitors (either related to the same magnet or to magnets at different locations), the number of monitors and their channels provide redundancy with regard to beam loss detection. On the other hand, the large number of monitors makes a significant contribution to the false beam dumps (Fig. 7.3). In order to reduce the number of false beam dumps triggered by monitors (or FEEs), the MPS offers the option of masking some of the monitor channels, i.e. ignoring the dump requests of selected monitors and their channels (Section 3.3.3).

This case study refers to these aspects by including a weighted demand pattern and the masking of monitors. The implementation includes major adaptations of the MPS model with regard to the system demand and the component model of the monitors.

9.1.1 System Demand

Table 9.1 shows the adaptations included in the weighted system demand pattern compared to the demand pattern of the MPS Model. The weighted demand pattern covers the whole machine and triggers the monitors of up to ten magnets at the same time. It triggers either a single or two subsequent monitors of the three monitors (per beam) covering a magnet.

Table 9.1: *Coverage of weighted system demand pattern*

	Common demand pattern (MPS Model)	**Weighted demand pattern (Extension)**
Sectors	1	all sectors
Magnets	1	1 to 10
Monitors	all 6	1 or 2 (one beam)

Figure 9.2 gives a schematic representation of the algorithm defining the weighted system demand pattern. Upon the occurence of a beam loss event, i.e. the expiry of object object *beamLossTimer* (Section 5.2.1.5), the number of affected magnets is determined (up to ten, uniformly distributed). In the next steps, the affected magnet (one out of 624, uniformly distributed), the beam (beam 1 or 2) and the number of affected monitors thereof is chosen. In 60% of the cases, two monitors are assumed to be affected, of which 90% concern the first and second monitor (down-stream) and 10% the second and third. In 40% of the cases, only one monitor is assumed to be affected, of which 70% concern the first, 20% the second and 10% the third (downstream). The procedure is repeated until the required number of magnets (and related monitors) are triggered. It is to be noted that the algorithm does not include time delay. All affected monitors are triggered at the same model time, e.g. the time of the beam loss event.

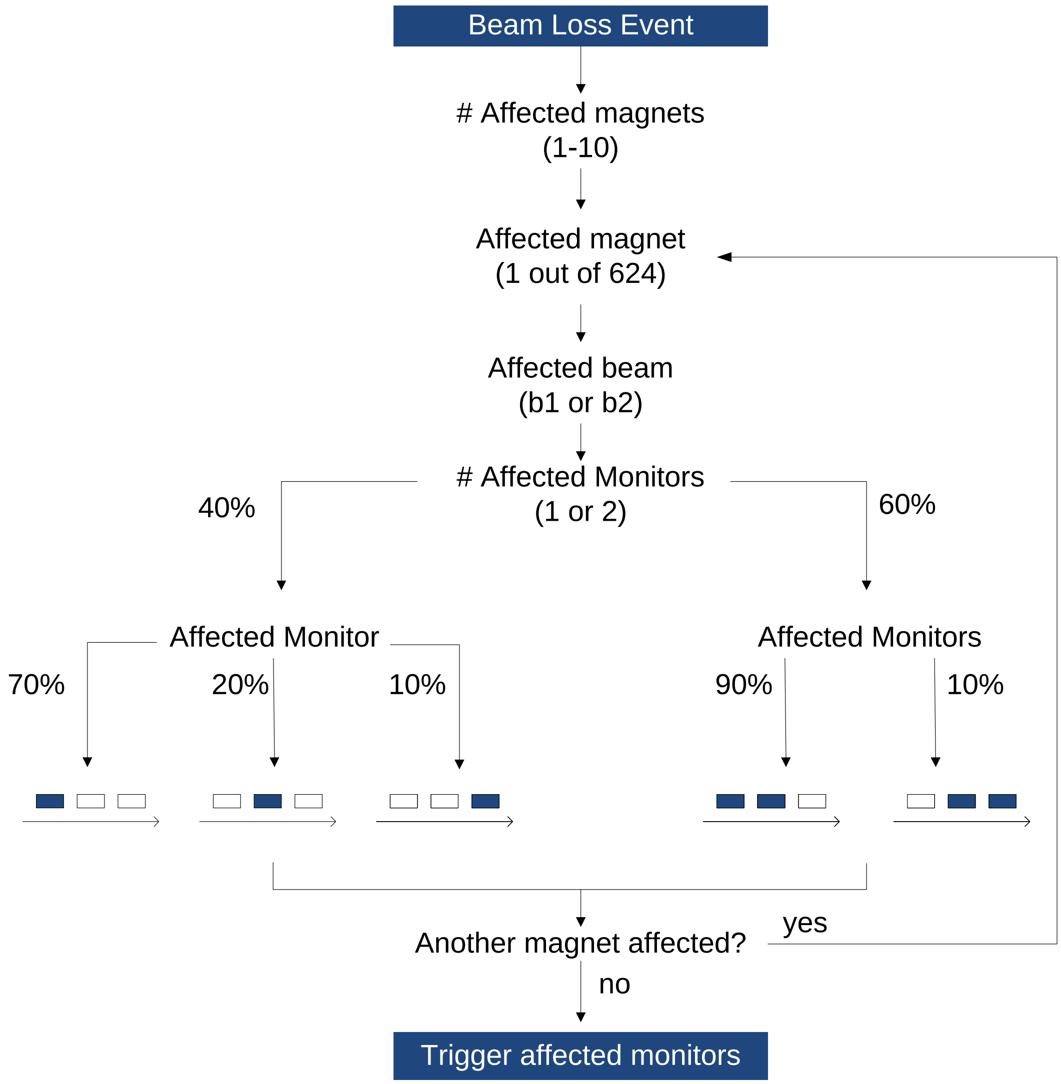


Figure 9.2: *Schematic algorithm representation for weighted system demand pattern*

The algorithm for the weighted system demand pattern is given in detail in Appendix B. The complete implementation of the weighted demand pattern entails slight changes in the logging of the data and therefore requires adaptations in the data analysis.

9.1.2 Component Model

Figure 9.3 presents the adapted monitor state diagram underlying the implementation of the monitor masking feature. It is implemented simply by adding an additional state (*masked* state). The *masked* state in its consequences corresponds to the *blind* state except for being set deterministically at the start of each simulated mission instead of upon a failure.

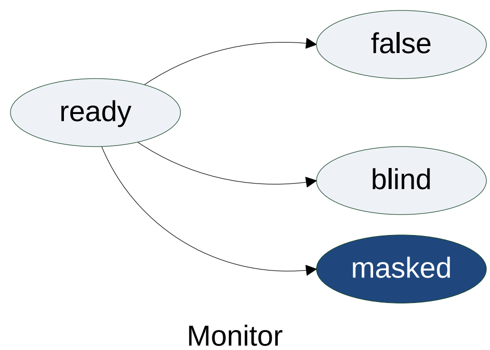


Figure 9.3: *Monitor Masking: Monitor state diagram with additional state masked*

9.1.3 Results

The results presented in this section are based on 10^5 simulated missions, using the system demand rate and component failure rates of study *STANDARD B* (Tab. 7.2 and 7.3). It includes the weighted system demand pattern described above. Figure 9.4 illustrates the resulting demand pattern on the basis of four magnets with three monitors each (one beam). It shows the trigger frequency of the monitors, with the weighting on the first and second monitor (downstream).

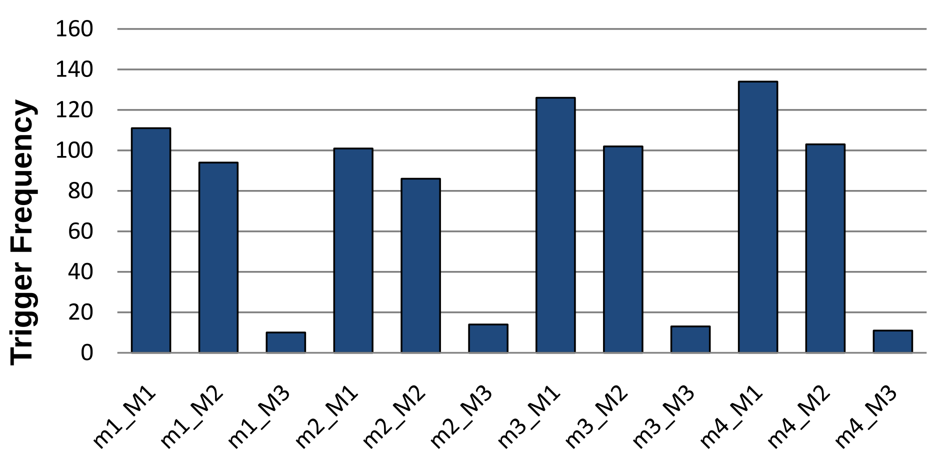


Figure 9.4: *Resulting demand pattern on the Monitors of four magnets (one beam)*

Table 9.2 shows the relative scenario frequencies resulting from simulations with the weighted demand pattern (third column). The fourth column presents the related figures if two thirds of the Monitors, namely the second and third monitor downstream, are masked [38]. The second column provides the scenario probabilities of case study *STANDARD B* for comparison.

Table 9.2: *Weighted demand pattern: Scenario probabilities with and without Monitor Masking*

Scenario	STANDARD B	Weighted Demand Pattern	
		No Monitors masked	Monitors masked
I	6.8466E-1	6.8405E-1	6.8750E-1
II	2.9966E-1	3.0085E-1	2.9263E-1
III	1.568E-2	1.510E-2	1.327E-2
IV	(1E-8)	0	6.60E-3
V	(1E-9)	0	0

The simulations with the weighted demand pattern yield a relative frequency of completed missions (I) of 68.4%, while 30.1% of the missions are ended due to emergency beam dumps (II) and 1.5% due to false beam dumps (III). Compared to the common demand pattern of case study *STANDARD B*, the weighted demand pattern involves less redundancy across the monitors of one magnet, since at most two are triggered at the same time (instead of six, Tab. 9.1). This decrease of redunancy however is compensated by the fact that in 90% of the beam loss events more than one magnet is assumed to be affected, resulting in an additional redundancy across the monitors of several magnets. In the very rare case that the same magnet is randomly determined more than once within an event, the redundancy across the monitors of a magnet is increased (to the extent of the common 1-out-of-6 redundancy at most). In view of this compensation, the results in the range of the scenario probablities of case study *STANDARD B* are reasonable. More extensive simulations, i.e. simulated missions, allow for a more thorough comparison.

The inclusion of Monitor Masking results in the expected effects, namely the decrease of false beam dumps (III) and a significant increase of missed emergency beam dumps (IV) up to a relative frequency of about 0.7% [38]. Thus with two-thirds of the monitors masked, several missed emergency beam dumps would have to be expected per year.

9.2 Common Cause Failures

The MPS model assumes the component failures to be independent (Section 4.3.1). This assumption disregards the occurrence of common cause failures, which, due to their extensive impact on system functionality, are perceived to be crucial with regard to system reliability.

The implementation of common cause failures is presented by means of two case studies, BEAM ENERGY SIGNAL (Section 9.2.1) and SAFE BEAM FLAG (Section 9.2.2). In both cases, the underlying common cause failure complies with the following definition:

Common Cause Failure Simultaneous failing of several components (of the primary signal path) caused by a faulty common secondary input.

Such common cause failures thus stand out by two characteristics, 1) they refer to failures that prevent several components at the same time from fulfilling their primary function and 2) the failures are external to the components, the malfunctioning of the components is a result of a nominal reaction of the (fault-free) component to a faulty input. The latter in particular excludes failure cascades across the components of the primary signal path, i.e. the failure of one component entailing the failure of others.

The implementation of the common cause failures includes major adaptations of the MPS model. It involves the extension of the model input parameters by additional rates defining the common cause failure, the definition of the failure pattern and the adaptation of the related component models, as described in the following sections.

9.2.1 Case Study: BEAM ENERGY SIGNAL

The Beam Energy Measurement System (BEMS) is a MPS subsystem whose primary function is to measure the current energy of the beams in the machine and provide the information as secondary input signal to a series of components of other subsystems. The BEEs of the BLMS are components with a crucial dependency on the Beam Energy signal. The beam energy defines the threshold that incoming (primary) beam loss signals are checked against (Section 3.3.1). If the beam loss is equal to or above the threshold for the given beam energy, a beam dump request is triggered. A faulty Beam Energy signal, i.e. not corresponding to the actual energy of the beams in the machine, results in a wrong threshold interfering with the BEEs functionality. The basic interrelations are shown in Table 9.3.

Table 9.3: *Beam Energy Signal: Basic interrelation with BEE function*

Energy signal	**Threshold**	**Consequence upon** non-nominal beam loss	nominal beam loss	**Failure mode**
Correct	Correct	Beam dump request	No beam dump request	NA (ready)
Too low	Too high	No beam dump request	NA (no dump request)	blind
Too high	Too low	NA (dump request)	Beam dump request	false

The consequences of a faulty Beam Energy signal (meeting a fully functional BEE) can directly be assigned to the two BEE failure modes. Too low a Beam Energy signal entails too high a threshold which means that a certain fraction of non-nominal beam loss detected by the related monitors (i.e. incoming emergency beam dump request) is ignored (absorbed) by the BEE. The same applies to incoming beam dump requests triggered upon an upstream component (FEE, Monitor) gone *false*. Too high a Beam Energy signal on the other hand entails too low a threshold which means that a certain fraction of nominal beam loss detected by the monitors (i.e. no incoming beam dump request) cause the triggering of a beam dump request by the BEE.

The extent to which the BEE reacts inaccurately depends on the scale of the deviation of the Beam Energy signal. For simplicity reasons, a faulty Beam Energy signal is assumed to leave the BEE in the related failure mode permanently. It represents a worst-case assumption particularly in terms of the blind mode, ignoring that an affected BEE is blind up to the (faulty) threshold only (i.e. working normally with regard to beam losses above the threshold).

Figure 9.5 shows the related model extensions, the inclusion of a Beam Energy signal fault (left) and the extension of the BEE component model by two additional states (right).

The Beam Energy signal is assumed to go *too low* (*too high*, due to a failure of the BEMS or related components), with rate λ^B_{Energy} (λ^F_{Energy}, Fig. 9.5, left), causing the affected BEE to go *falseEnergy* (*blindEnergy*). This transition is immediate and deterministicly linked to the transition of the Beam Energy signal. The failure pattern defines which BEEs are affected, it is presented in Table 9.4.

Table 9.4: *Beam Energy Signal: Failure pattern of common cause failure*

Energy signal	**BEE failure mode**	**Affected BEEs**
Too low	Blind	All BEEs in BLMS branch encountering beam loss
Too high	False	Last BEE of last VME in BLMS branch encountering beam loss

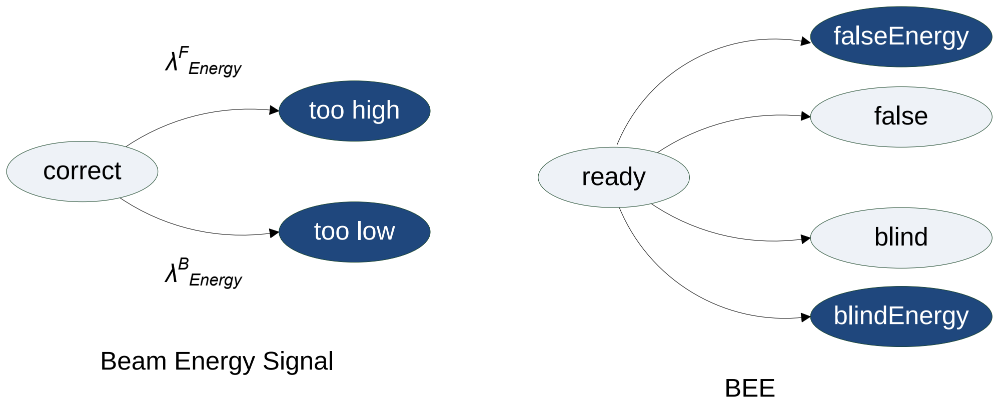


Figure 9.5: *Beam Energy Signal: State diagram (left) and BEE state diagram with additional states falseEnergy and blindEnergy (right)*

This pattern is an example out of a multitude of possible patterns. For case Beam Energy *too low*, the simultaneous impact on all BEEs (each going *blindEnergy*) in the one BMLS branch which encounters beam loss (Section 4.1.3.2) reflects the worst-case scenario with regard to machine safety. It means that in this case, any beam dump request triggered by any monitor or FEE of the BMLS branch is absorbed on BEE level. In case of Beam Energy going *too high*, a false beam dump request is triggered on BEE level of the BMLS branch encountering beam loss. For simulation reason, instead of all BEEs, the last BEE of the last crate (downstream) is set to *falseEnergy* only (entailing the trigger of a false beam dump request).

It is to be noted that the BEE component model (Fig. 9.5, right) implies that a common cause failure only affects the component if *ready*, thus corresponding to the specification given above. The algorithm for the failure pattern is given in Appendix B.

9.2.2 Case Study: SAFE BEAM FLAG

This section independently presents a case study which is linked to both the Beam Energy signal (Section 9.2.1) and the Monitor Masking (Section 9.1). The Safe Beam Flag (SBF) is a flag in the BICs categorising the current beam status as safe or unsafe, depending on beam energy and intensity (Section 3.3.3). In case of safe beam, i.e. low beam energy and intensity, the SBF is *true*, otherwise *false*. *SBF true* allows for the masking of certain monitor channels as a measure to reduce the number of false beam dumps. In case of SBF *true* and applied channel masking, the dump requests from the maskable channel (incorporating the monitor channels assigned to the maskable channel at BEE level) are ignored (i.e. the USER_PERMIT signal of the maskable channel is deterministically considered *true*).

Table 9.5: *Safe Beam Flag: Basic interrelation with Beam Status and Channel Masking at BIC*

	Beam safe	Beam unsafe
SBF true	Channel Masking (nominal)	Channel Masking (unsafe)
SBF false	No Channel Masking (safe)	No Channel Masking (nominal)

Table 9.5 shows the four basic scenarios with regard to Beam Status, SBF and Channel Masking. In terms of machine safety, the scenario *Beam unsafe/SBF true* is crucial. The masking of monitor channels at unsafe beam results in an increased risk of missing a non-nominal beam loss with high damage potential (due to high beam energy and intensity). This scenario is addressed with a model extension similar to the one presented in previous Section 9.2.1. Figure 9.6 shows the model for the SBF (left) and the extension of the BIC component model by the additional state *blindSBF* (right).

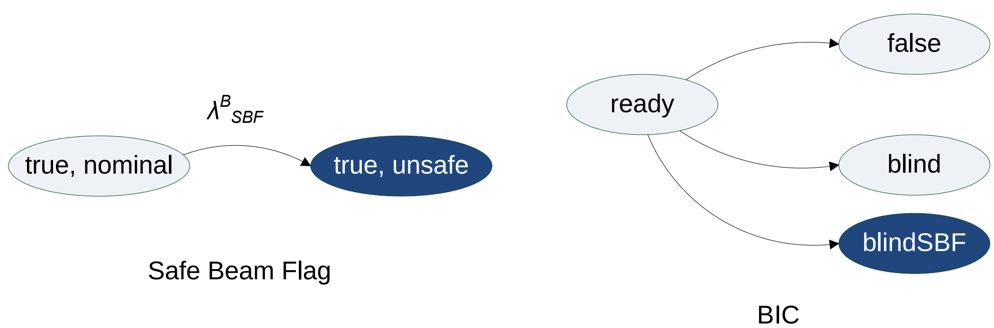


Figure 9.6: *Safe Beam Flag: State diagram (left) and BIC state diagram with additional state blindSBF (right)*

The model for the SBF reflects the scenario of the flag *true* (at beam safe) NOT switching to *false* upon the beam becoming unsafe, i.e. the virtual SBF change from *true nominal* to *true unsafe* (Fig. 9.6, left). This scenario is assumed to occur exponentially distributed with rate λ^B_{SBF}, due to a faulty (too low) beam energy or intensity signal. The transition deterministically entails the BICs going *blindSBF* (Fig. 9.6, right), according to the failure pattern presented in Table 9.6.

The presented model is a very first approach to the modelling of the SBF common cause failures. It involves a series of simplifications and assumptions, the most relevant of which are shown in Table 9.7.

Table 9.6: *Safe Beam Flag: Failure pattern of common cause failure*

Safe Beam Flag	BIC failure mode	Affected BICs
True, unsafe	Blind	All BICs

Table 9.7: *Safe Beam Flag: Failure model simplifications and assumptions*

Feature	Model	Reality
Time window	Whole mission	Up to moment of beam getting 'unsafe'
Failure Pattern	All BICs (32) affected	Part of the BICs affected at a time
Masked channels	All Monitor channels of BIC	Monitor channels assigned to maskable channel of BIC

The extended time window for the SBF common cause failure to occur compared to reality entails an increased failure probability at the given rate λ^{B}_{SBF}, thus relating to a conservative approach. In a nominal cycle, the beam reaches critical ('unsafe') intensity in the injection phase already. The assumption that all BICs are affected by the faulty SBF represents the worst-case scenario in terms of failure pattern. The same applies to the masking of all monitor channels (instead of the part of the monitor channels assigned to the maskable channels).

The subject of this case study to a certain extent overlaps with the one of studies *Monitor Masking* (Section 9.1) and *BEAM ENERGY SIGNAL* (Section 9.2.1). The study differs from *Monitor Masking* in the following characteristics:

- **System demand rate:** 1E-2h^{-1} vs. 3E-2 h^{-1}
- **Occurence of masking:** Stochastic (anytime during mission) vs. deterministic (at beginning of mission)
- **Masked part of the system:** All BLMS branches (due to blind BICs) vs. two-thirds of the monitors

The study differs from *BEAM ENERGY SIGNAL* common cause failures (Chapter 9.2.1) in the following characteristics:

- **Affected component:** BIC vs. BEE
- **Coverage:** All BLMS branches (all BICs affected) vs. BLMS branch encountering beam loss (all BEEs of branch)

- **Resulting failure mode:** *blind* vs. *blind* and *false*

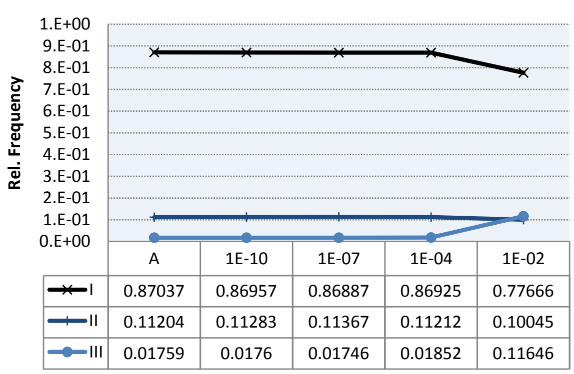
The algorithm for the failure pattern is given in Appendix B.

9.2.3 Results

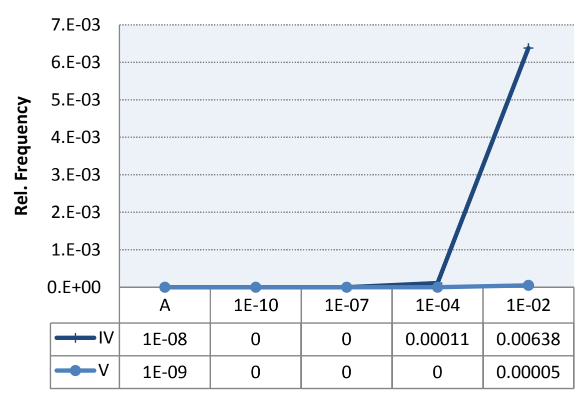
The results presented in this section base upon 100,000 simulated missions, using the system demand rate and component failure rates of study *STANDARD A* (Tab. 7.2 and 7.3).

9.2.3.1 Case Study: BEAM ENERGY SIGNAL

Figure 9.7 shows the relative scenario frequencies for rates λ^B_{Energy} and λ^F_{Energy} varying between 1E-10 and 1E-2h^{-1}. Set *A* represents the scenario probabilities of case study *STANDARD A* (Section 7.2) for comparison. Based on 10^5 simulated missions, no significant impact is observed for BEMS failures with rates λ^B_{Energy} and λ^F_{Energy} between 1E-10 and 1E-7h^{-1}. Increased rates of 1E-4h^{-1} result in a relative frequency of missed emergency beam dumps (IV) in the range of 1E-5 or above. The further increase of the rates by two orders of magnitudes entails a significant increase of false beam dumps (III) and missed emergency beam dumps (IV) by about one order of magnitude each, at the expense of completed missions (I).



	A	1E-10	1E-07	1E-04	1E-02
I	0.87037	0.86957	0.86887	0.86925	0.77666
II	0.11204	0.11283	0.11367	0.11212	0.10045
III	0.01759	0.0176	0.01746	0.01852	0.11646

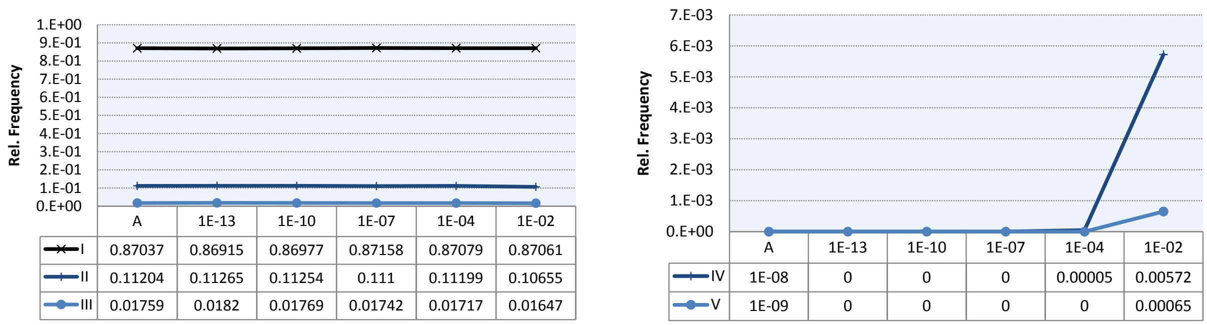


	A	1E-10	1E-07	1E-04	1E-02
IV	1E-08	0	0	0.00011	0.00638
V	1E-09	0	0	0	0.00005

Figure 9.7: *Beam Energy Signal: Relative scenario frequencies against rates λ^B_{Energy} and λ^F_{Energy} between 1E-10 and 1E-2h^{-1}. A: STANDARD A*

9.2.3.2 Case Study: SAFE BEAM FLAG

Figure 9.8 shows the relative scenario frequencies for rate λ^B_{SBF} varying between 1E-13 and 1E-2h^{-1}. Set *A* represents the scenario probabilities of case study *STANDARD A* (Section 7.2) for comparison. Based on 10^5 simulated missions, no significant impact is observed for flag faults based on a failure rate λ^B_{SBF} between 1E-13 and 1E-7h^{-1}. An increased rate of 1E-4h^{-1} results in a relative frequency of missed emergency beam dumps (IV) in the range of 1E-5. The further increase of the rate by two orders of magnitude entails a significant increase of missed emergency beam dumps (IV) by about two orders of magnitude.



	A	1E-13	1E-10	1E-07	1E-04	1E-02
I	0.87037	0.86915	0.86977	0.87158	0.87079	0.87061
II	0.11204	0.11265	0.11254	0.111	0.11199	0.10655
III	0.01759	0.0182	0.01769	0.01742	0.01717	0.01647

	A	1E-13	1E-10	1E-07	1E-04	1E-02
IV	1E-08	0	0	0	0.00005	0.00572
V	1E-09	0	0	0	0	0.00065

Figure 9.8: *Safe Beam Flag: Relative scenario frequencies against rate λ^B_{SBF} between 1E-13 and 1E-2h^{-1}. A: STANDARD A*

In comparison with study *BEAM ENERGY SIGNAL* (Fig. 9.7), the most significant difference appears in scenarios *false beam dumps* (III) and *missed false beam dumps* (V), for rates 1E-2 h^{-1}. The difference in the false beam dumps is due to additional false beam dump triggering included by rate λ^F_{Energy} in *BEAM ENERGY SIGNAL* only. The difference in missed false beam dumps is due to the fact that the flag fault affects all of the BICs (Tab. 9.6), i.e. *blind* BICs not only concern the BLMS branch encountering emergency beam dump triggers, but all of the branches and their false beam dump triggers.

9.3 Discussion

The presented case studies exemplify the potential of the modeling approach with regard to the inclusion of more sophisticated system features. The implementation of a weighted system demand pattern, the masking of components (e.g. Monitors) and common cause failures for simulations is straightforward and is not exhausted by the presented case studies. The simulations can be used both for adressing the impact of specific system configurations (e.g. the masking of a specific set of components) and for a more playful approach with rough assumptions (e.g. with regard to the components affected by a common cause failure) to get a first impression on principal interrelations which are to be further addressed.

The component masking feature provides a means for comparing different system configurations with respect to the balancing of machine safety and beam availability. The case study on Monitor Masking reveals that, as is to be expected given the underlying system demand pattern, that the masking of two-thirds of the monitors would result in a frequency of missed emergency beam dumps that is not acceptable.

The case studies on common cause failures as a first hint reveal that attention has to be paid to the common cause failures with rates from 1E-7 h^{-1} upward, since they rise the probability of the safety relevant scenario IV to a range of 1E-5 and beyond (a probability of 1E-4 relates to 1 missed emergency beam dump in 25 years, which is not acceptable). The impact of the common cause failures resulting in false beam dumps is less crucial. It confirms the previously made observation that they become significant when their rate approaches the range of the system demand rate (1E-2h^{-1}), Fig. 9.7).

The comparison of the simulation results with the *STANDARD* probabilities shows good correspondence given the limited number of simulated missions, and as such provides a rough verification of the simulations. The simulation time of about two hours for 100,000 missions still is a major limitation. However, continued studies have shown that weighted demand patterns and component masking can be included in the analytical description (Chapter 11).

Chapter 10

Conclusions

The LHC and its Machine Protection System (MPS) are highly complex and set new standards in terms of size, performance and requirements. The facility represents a prototype, being subject to ongoing enhancement and modification. However, each reliability analysis and notably the development of an appropriate method for an analysis requires an extensive comprehension of the system to be analysed. To gain the required system comprehension for this thesis, the author was given access to a wide range of documentation and specifications of the different subsystems and their parts, as well as the opportunity for discussions with the system experts. Due to the mentioned characteristics of the LHC and the MPS, this process accounts for a major part of this thesis. The result of the process, an overview on the LHC and the MPS in particular, is presented at the beginning of the thesis.

The thesis then introduces a quantitative method for the reliability analysis of the LHC MPS. It is based on an object-oriented modelling approach, that is the component-by-component representation of the primary signal path. The components' behaviour is described by a Markov model with two failure states, *blind* and *false*. Five scenarios and related component importances are quantified, including the *missed emergency beam dump* and the *false beam dump* scenario. The model input parameters comprise the component failure rates, a system demand rate and the model observation time (corresponding to the length of a LHC mission).

The presented MPS model covers the BLMS and the BIS, including almost 5000 individually modelled components. Two model implementations are shown, the first based on Monte Carlo simulation and the second based on an analytical model description. Case studies based on the analytical description yield a probability in the order of magnitude of 1E-2 that a LHC mission is ended early by a false beam dump. The probability of a missed emergency beam dump is determined to be in the range of 1E-8 (1E-6 in case of significantly increased failure rates for CIBUS). The results confirm expectations based on operational experience and previous studies. They are verified by the results of related Monte Carlo simulations, which show good accordance (relative error of the frequent-scenario probabilities in the range of 1E-4 to 1E-6).

A series of sensitivity analysis studies based on input parameter variation are presented. The studies do not reveal any unexpected dependence and as such confirm the stability of the model implementation. In addition, specific model extensions are suggested which allow to overcome some of the limitations of the initial model with regard to the real system. The extensions relate to the inclusion of common cause failures, weighted demand pattern and the masking of components.

The thesis shows the feasibility and useability of the developed modelling approach. Due to the direct imitation of the physical signal path and the related modular conception, the model is straight-forward and easily traceable, facilitating the information exchange between system engineers and reliability specialists, which is essential for a system analysis. The modular, bottom-up concept eases the analysis for large and complex systems and allows for a quick and easy realisation of selective model adaptations. This is a major advantage compared to the use of, for example, Markov models.

The explicit inclusion of the machine reliability (as system demand) makes the simultaneous quantification of five scenarios possible, which, contrary to the use of FTs, are all covered by the same model. The possibility to define specific system demand patterns furthermore allows for a weighting of the different system branches close to reality. Fail-safe measures included in the system design are taken into account by a second component failure mode (*false*). Redundancies within the components are included in the component failure rates, while redundancies on subsystem and system level are accounted for in the representation of the signal path.

Generally speaking, the method with its object-oriented modeling approach allows for the straight-forward inclusion of very specific system features. Its flexibility has been exemplified by the model extensions in terms of common cause failures, weighted demand pattern and the masking of components. The two options of implementation provide a powerful means for verification and also the choice to use the more suitable of the two for each specific application or adaption.

The limitations of the modeling approach are defined by the implementations. In terms of the Monte Carlo approach, the obvious limitation is the extensive simulation effort (i.e. time) needed to gain significant data, particularly with regard to the rare scenarios. In terms of the analytical description, the manual building of the equations sets the limits with regard to the size and complexity of the model. Also, the validation of the model is pending. It has to be performed along with future operational experience.

A reliability analysis always includes simplifications. Due to a model representation close to the physical system structure, the simplifications and assumptions underlying the developed approach are regarded as more easily to grasp compared to established, more abstract methods. The developed method is considered to be a useful means for the assessment of the MPS. It can help to gain a deeper understanding of specific dependencies and provide hints with regard to possible system optimisation. The model still holds big potential for extensions, since neither of the two implementations are exhausted yet. They may allow for the inclusion of additional model features.

From an application point of view, that is in terms of the analysis of the LHC MPS, it has to be noted that the presented model and the emerging results cover the BLMS and the BIS only. The MPS includes a range of other subsystems, which are not taken into account. Their inclusion into the model would add further thousands of components, depending on the considered level of detail.

Some of these systems are redundant to the BLMS and, to a lesser extent, to the BIS. Therefore, their inclusion is expected to have a lowering effect on the probability of missed emergency beam dumps and an increasing effect on the probability of false beam dumps (compared to the results obtained by the case studies). This however does not apply to the LBDS, which represents the direct continuation of the signal path from the BIS to the actuators (e.g. the extraction kicker magnets). The LBDS is crucial for the performance of any beam dump and therefore is safety relevant. Hence, while possibly resulting in an increase of the probability for false beam dumps, its inclusion into the model is expected to have an increasing effect on the probability of missed emergency beam dumps (depending on the reliability of the LBDS).

In view of the previously mentioned limitations of the model, it is considered reasonable to start a possible extension by additional subsystems at a lower level of detail in order to limit the number of components to be modelled.

Chapter 11

Outlook

In continuation of the parallel concept of model implementation, the question arises whether the analytical description is extendable by the weigthed demand pattern and the monitor masking (Fig. 11.1). In addition, considering the advantages and drawbacks of the two implementations, the question arises whether an automatic generation of the analytical description based on a graphical system representation is feasible. This would represent the combination of the advantages of the two implementations, that is straight-forward graphical system representation and fast and accurate calculation. At the same time, their major drawbacks, that is the simulation effort and the error-prone and costly manual development of the analytical description, would be removed.

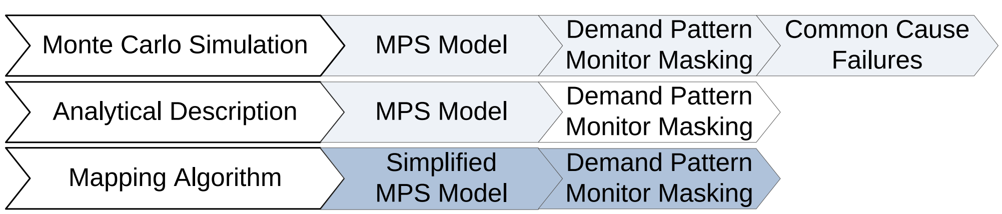


Figure 11.1: *Outlook: Mapping Algorithm*

Based on a feasibility study performed in cooperation with Nibali [66], the above questions can be answered with yes. A mapping algorithm has been developed and applied to a simplified MPS model with 20 components. The simplified model is implemented according to the MPS model features as well as with the additional features of weighted demand pattern and monitor masking. All models are also implemented based on Monte Carlo simulation for verification, with the results showing good accordance.

The algorithm is implemented using Maple. It bases upon a context-free grammar developed by Nibali to describe the system. The notation captures the system structure, the failure rates of the involved components and other parameters and characteristics such as weighting factors related to the system demand or indicators for component masking.

Each component is assigned a set of terms with predefined initial values. The algorithm identifies the parallel and serial elements of the system and reduces them in a recursive manner into one component. In each recursive step, two components are replaced by a single one whose terms are defined according to generic predefined rules for serial or parallel reduction, based on the terms of the replaced components. After the recursion, the equations of the five scenarios are built in a generic predefined manner, using the terms of the final component which include all information on the system.

The feasibility of a mapping algorithm for the automatic generation of the analytical description has been shown. The algorithm provides big potential for further development.

List of Tables

List of Figures

List of Abbreviations

ARW	Accelerator Reliability Workshop
BEE	Back-End Electronics
BEMS	Beam Energy Measurement System
BIC	Beam Interlock Controller
BIS	Beam Interlock System
BLM	Beam Loss Monitor
BLMS	Beam Loss Monitor System
BPM	Beam Position Monitor
CC	Combiner Card
CCF	Common Cause Failure
CERN	European Organization for Nuclear Research
CIBU	User Interface of the BIS
CIBUD	User Interface with double connection
CIBUS	User Interface with single connection
FEE	Front-End Electronics
FBCM	Fast Beam Current Change Monitor
FMCM	Fast Magnet Current Change Monitor
FMECA	Failure Mode, Effect and Criticality Analysis
FT	Fault Tree
FTA	Fault Tree Analysis
IC	Ionisation Chamber
IP	Interaction Point
IR	Insertion Region
LBDS	Beam Dumping System
LHC	Large Hadron Collider
MPS	Machine Protection System

NC	Normal conducting
PIC	Powering Interlock Controller
PSA	Probabilistic Safety Assessment
RBD	Reliablity Block Diagram
RF	Radio frequency
SBF	Safe Beam Flag
SC	Superconducting
SEM	Secondary Emission Monitor
SPS	Super Proton Synchrotron
VME	Versa Module Europa
WIC	Warm Interlock Controller

Bibliography

[1] B. Todd. *A beam interlock system for CERN high energy accelerators*. PhD thesis, Brunel University West London, 2006.

[2] M. Bajko, F. Bertinelli, N. Catalan-Lasheras, S. Claudet, P. Cruikshank, K. Dahlerup-Petersen, R. Denz, P. Fessia, C. Garion, J.M. Jimenez, G. Kirby, Ph. Lebrun, S. Le Naour, K.-H. Mess, M. Modena, V. Montabonnet, R. Nunes, V. Parma, A. Perin, G. de Rijk, A. Rijllart, L. Rossi, R. Schmidt, A. Siemko, P. Strubin, L. Tavian, H. Thiesen, J. Tock, E. Todesco, R. Veness, A. Verweij, L. Walckiers, R. Van Weelderen, S. Wolf, R .and Fehér, R. Flora, M. Koratzinos, P. Limon, and J. Strait. Report of the Task Force on the Incident of 19th September 2008 at the LCH, CERN, Geneva. 2009.

[3] CERN. LHC ends 2009 Run on a High Note. Press Release, PR20.09, 18 december 2009. 2008.

[4] http://public.web.cern.ch/public/en/LHC. CERN - The Large Hadron Collider .

[5] P. A. Tipler and R. A. Llewellyn. *Modern Physics. Fifth Edition.* W. H. Freeman and Company, New York, 2008.

[6] http://www.triumf.info/hosted/ARW/index.htm. Accelerator reliability workshop 2009.

[7] P. Bellomo and D. MacNair. SLAC Next-Generation High Availability Power Supply. In *Accelerator Reliability Workshop 2009 (ARW2009)* , Vancouver, Canada, 2009.

[8] R.S. Larsen. SLAC Accelerator Research High Availability Electronics R&D Program. In *Accelerator Reliability Workshop 2009 (ARW2009)* , Vancouver, Canada, 2009.

[9] M. Takao. Failure Postmortem Analysis at the SPring-8 Storage Ring. In *Accelerator Reliability Workshop 2009 (ARW2009)* , Vancouver, Canada, 2009.

[10] K.T. Hsu, K.H. Hu, J. Chen, C.Y. Wu, D. Lee, P.C. Chiu, Y.S. Cheng, C.H. Kuo, S.Y. Hsu, C.J. Wang, M.S. Yeh, and Ch. Wang. Post-Mortem Diagnosics for the Taiwan Light Source. In *Accelerator Reliability Workshop 2009 (ARW2009)* , Vancouver, Canada, 2009.

[11] P.W. Sampson. Infrastructure Improvement and RHIC Reliability. In *Accelerator Reliability Workshop 2009 (ARW2009)* , Vancouver, Canada, 2009.

[12] T. Kadowaki. Present Status and Future Plan for Automated Routine in HIMAC Operation. In *Accelerator Reliability Workshop 2009 (ARW2009)* , Vancouver, Canada, 2009.

[13] E. Matias, M. de Jong, and L. Dallin. CLS Control System Contribution to System Reliability. In *Accelerator Reliability Workshop 2009 (ARW2009)* , Vancouver, Canada, 2009.

[14] J.F. Lamarre, X. Deletoille, J.C. Besson, A. Bence, R. Cuoq, D. Pereira, E. Patry, S. Petit, G. Roux, and Y. Denis. Performances of the Soleil Synchrotron Light Source Operation. In *Accelerator Reliability Workshop 2009 (ARW2009)* , Vancouver, Canada, 2009.

[15] S. Krecic. New Reliability Aspects for ELETTRA after Booster Installation. In *Accelerator Reliability Workshop 2009 (ARW2009)* , Vancouver, Canada, 2009.

[16] M. Walter and W. Schneeweiss. *The Modeling World of Reliability/Safety Engineering*. LiLoLe, Hagen, 2005.

[17] O. P. M. Nusbaumer. *Analytical Solutions of Linked Fault Tree Probabilistic Risk Assessments using Binary Decision Diagrams with Emphasis on Nuclear Safety Applications*. PhD thesis, ETH Zurich, 2007.

[18] L. Podofillini, V. N. Dang, and K. Thomsen. Scoping-level Probabilistic Safety Assessment of a Complex Experimental Facility: Challenges and First Results from the Application to a Neutron Source Facility (MEGAPIE). *Nuclear Engineering and Design*, 238(10):2726–2738, 2008.

[19] L. Burgazzi and P. Pierini. Reliability studies of a high-power proton accelerator for accelerator-driven system applications for nuclear waste transmutation. *Reliability Engineering & System Safety*, 92(4):449–463, 2007.

[20] G. Guaglio. *Reliability of the Beam Loss Monitors System for the Large Hadron Collider at CERN*. PhD thesis, Université Clermont Ferrand II - Blaise Pascal, 2005.

[21] R. Filippini. *Dependability analysis of a safety critical system: The LHC Beam Dumping System at CERN*. PhD thesis, Università di Pisa, 2006.

[22] A. Vergara Fernandez. *Reliability of the Quench Protection System for the LHC superconducting elements*. PhD thesis, Universitat Politècnica de Catalunya, 2003.

[23] O. S. Brüning, P. Collier, P. Lebrun, S. Myers, R. Ostojic, J. Poole, and P. Proudlock. LHC Design Report. Vol. II: The LHC Infrastructure and General Services. CERN, Geneva. 2004.

[24] P. Halpern. *Collider: the search for the world's smallest particles*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2009.

[25] R. Schmidt, R. Assmann, E. Carlier, B. Dehning, R. Denz, B. Goddard, E. B. Holzer, V. Kain, B. Puccio, B. Todd, J. Uythoven, J. Wenninger, and M. Zerlauth. Protection of the CERN Large Hadron Collider. *New Journal of Physics*, 8:290. 31 p, 2006.

[26] A. Gomez Alonso. *Redundancy of the LHC machine protection systems in case of magnet failures*. PhD thesis, UPC Barcelona, 2009.

[27] O. S. Brüning, P. Collier, P. Lebrun, S. Myers, R. Ostojic, J. Poole, and P. Proudlock. LHC Design Report. Vol. I: The LHC Main Ring. CERN, Geneva. 2004.

[28] M. Conte and W. W. MacKay. *An Introduction to the Physics of Particle Accelerators*. World Scientific Publishing, Singapore, 2009.

[29] http://atlas.ch. ATLAS Experiment.

[30] http://cms.web.cern.ch/cms. CMS Experiment.

[31] http://aliceinfo.cern.ch/Public. ALICE - A Large Ion Collider Experiment.

[32] http://lhcb-public.web.cern.ch/lhcb-public. LHCb - Large Hadron Collider beauty experiment.

[33] http://public.web.cern.ch/public/en/LHC/TOTEM-en.html. CERN - LHC Experiments: TOTEM.

[34] http://public.web.cern.ch/public/en/LHC/LHCf-en.html. CERN - The LHC Experiments: LHCf.

[35] B. Todd. CERN and the LHC Machine Protection System. Presented at External Review of LHC Beam Interlock System. CERN. 2009.

[36] H. Wiedemann. *Particle Accelerator Physics, Third Edition*. Springer, Berlin Heidelberg, 2007.

[37] F. Sonnemann. *Resistive Transition and Protection of LHC Superconducting Cables and Magnets*. PhD thesis, TU Aachen, 2001.

[38] S. Wagner, R. Schmidt, and J. Wenninger. Reliability Analysis of the LHC Machine Protection System: Terminology and Methodology. In *European Particle Accelerator Conference 2008 (EPAC'08)*, Genoa, Italy, 2008.

[39] EN13306. *Maintenance Terminology*. Beuth, Berlin, 2001.

[40] EN61703. *Mathematische Ausdrücke für Begriffe der Funktionsfähigkeit, Verfügbarkeit, Instandhaltbarkeit und Instandhaltungsbereitschaft*. Beuth, Berlin, 2002.

[41] VDI4001. *Begriffsbestimmungen zum Gebrauch des VDI-Handbuches Technische Zuverlässigkeit, Blatt 2*. VDI, Düsseldorf, 1986.

[42] http://www.ieeetcsc.org. IEEE Technical Committee on Scalable Computing.

[43] A. Birolini. *Reliability Engineering. Theory and Practice. Fifth Edition*. Springer, Berlin, 2007.

[44] O. R. Jones. LHC Beam Intrumentation. In *Proceedings of the Particle Accelerator Conference 2007 (PAC07)*, Albuquerque, New Mexico, USA, 2007.

[45] M. Zerlauth. *Powering and Machine Protection of the Superconducting LHC Accelerator*. PhD thesis, Technische Universität Graz, 2004.

[46] J. Pukite and P. Pukite. *Modeling for Reliability Analysis. Markov Modeling for Reliability, Maintainability, Safety, and Supportability Analysis of Complex Computer Systems*. Institute of Electrical and Electronics Engineers, New York, 1998.

[47] D. L. McLeish. *Monte Carlo simulation and finance*. John Wiley & Sons Inc., 2005.

[48] B. Todd. personal communication, CERN. 2007.

[49] A. V. Borshchev, Y. Karpov, and V. Kharitonov. Distributed simulation of hybrid systems with AnyLogic and HLA. *Future Generation Computer Systems*, 18:829–839, 2002.

[50] http://www.xjtek.com. XJ Technologies.

[51] XJ Technologies. Anylogic enterprise library reference guide. 1992-2005.

[52] R. Nibali. Handling Guide: Monte Carlo Simulation (Lane 1). MPS Model Implementation in AnyLogic. ETH Zurich, Laboratory for Safety Analysis (Internal documentation). 2010.

[53] http://www.r-project.org. The R Project for Statistical Computing.

[54] R. Nibali. Handling Guide: Analytical Description (Lane 2). MPS Model Implementation in Maple. ETH Zurich, Laboratory for Safety Analysis (Internal documentation). 2010.

[55] R. Nibali. Handling Guide: Analytical Description (Lane 2). Digits, Precision and Epsilon in Maple. ETH Zurich, Laboratory for Safety Analysis (Internal documentation). 2010.

[56] K. Wittenburg. Quench levels and transient beam losses at hera. *Talk given on the CARE AMT Workshop on 'Beam Generated Heat Deposition and Quench Levels for LHC Magnets', CERN, 3-4 March 2005*.

[57] S. M. Ross. *Introduction to Probability and Statistics for Engineers and Scientists*. Elsevier, Oxford, 2009.

[58] B. Todd. personal communication, CERN. 2009.

[59] B. Dehning. personal communication, CERN. 2009.

[60] B. Todd. personal communication, CERN. 2009.

[61] S. Wagner, G. Guaglio, I. Eusgeld, and W. Kröger. Balancing Safety and Availability for an Electronic Protection System. In S. Martorell, C. Guedes Soares, and J. Barnett, editors, *Safety, Reliability and Risk Analysis: Theory, Methods and Applications*, pages 2541–2548, London, 2009. Taylor and Francis Group.

[62] K. Weltner, W. J. Weber, J. Grosjean, and P. Schuster. *Mathematics for Physicists and Engineers*. Springer, Berlin Heidelberg.

[63] S. Wagner, R. Nibali, R. Schmidt, and J. Wenninger. Reliability Analysis of the LHC Machine Protection System: Analytical Description. In *Particle Accelerator Conference 2009 (PAC'09)*, Vancouver, Canada, 2009.

[64] H. W. Coleman and W. G. Steele. *Experimentation, Validation, and Uncertainty Analysis for Engineers. Third edition.* Wiley, Hoboken, New Jersey, USA.

[65] B. Dehning. personal communication, CERN. 2008.

[66] R. Nibali. LHC Machine Protection System: Recursive Reduction Algorithm for Reliability Analysis. Feasibility Study. ETH Zurich, Laboratory for Safety Analysis (Internal documentation). 2010.

[67] DIN 820120. *Standardization, Part 120: Guidelines for the inclusion of safety aspects in standards (ISO/IEC Guide 51:1999)*. Beuth, Berlin, 2008-09.

Appendix A

Glossary

Availability	'Ability of an item to be in a state to perform a required function under given conditions at a given instant of time or during a given time interval (...)' [39].
Availability, beam	In the context of this work relates to the presence of colliding beams
Beam dump	Controlled extraction of the beam from the ring and deposition in absorbers
Beam dump, emergency	Beam dump triggered due to a non-nominal condition of machine equipment or beam (machine failure)
Beam dump, false	Beam dump triggered due to a failure in the MPS covered by fail-safe measures
Beam energy	Particle energy
Common Cause Failure	'Failures of different items resulting from the same direct cause and where these failures are not consequences of each other' [39]
Cycle, machine	Refers to one complete operation cycle of a machine, i.e. injection, ramp up, possible physics, beam dump and ramp down [27]
Electrical circuit	Sequence of connected electrical circuit elements, connecting the magnets to one or several power converters [45]

Electrical circuit element	Single component with two connection points in an electrical circuit (e.g. Power converter, warm cable, warm tube, current lead, cold busbar piece, magnet, energy extraction switch or energy extraction resistor) [45]
Dynamic aperture	'Maximum initial oscillation amplitude that guarantees stable particle motion over a given number of turns' [27].
Fail-safe principle	Design principle with the purpose of inducing a safe state upon failure of an item, thus controlling and limiting the consequences of the failure
Failure	'Termination of the ability of an item to perform a required function' [39]. 'After failure the item has a fault' (...) [39]. '*Failure* is an event, as distinguished from *fault*, which is a state' [39]
Failure, blind	Undetected failure
Failure, false	Failure covered by fail-safe measures, triggering a beam dump request
Fault	'State of an item characterized by the inability to perform a required function (...)' [39]
Function, primary	Essential function directly related to the main purpose of an item
Function, required	'Function or a combination of functions of an item which are considered necessary to provide a given service' [39]
Item	'Any part, component, device, subsystem, functional unit, equipment or system that can be individually considered' [39]. 'A number of items e.g. a population of items, or a sample, may itself be considered as an item' [39].
Machine	All LHC components and systems that represent the essential equipment with regard to the LHC function, i.e. providing colliding beams. The beams are regarded as part of the machine.
Masking, channel	Predefined ignoring of specific user permit signals in the BIC
Monitoring	Activity intended to observe the actual state of a component ('item'), usually carried out during operation ('in the operating state') [39]

Nominal condition	Condition related to safe operation
Nominal condition, non-	Dangerous condition, implying damage potential
Permit, beam	Clearing for operation with beam by the BIS
Permit, user	Clearing for operation with beam by the User Systems
Quench, magnet	Resistive transition from superconducting to normal conducting state
Redundancy	'In an item, the existence of more than one mean at a given instant of time for performing a required function [39]
Reliability	'Ability of an item to perform a required function under given conditions for a given time interval' [39]. 'The term *reliability* is also used as a measure of reliability performance and may also be defined as a probability' [39].
Risk	Combination of the probability of a damage causing event and its consequence ([67], translated)
Safety	Freedom from unacceptable risk ([67], translated)
Safety, machine	Implies machine operation under *nominal conditions*
Signal path, primary	Sequence of interconnected components transmitting beam dump request signals from the monitors (sensors) to the magnets of the LBDS (actuators)
System demand	Need of the MPS to perform a beam dump due to non-nominal machine conditions
Time, calculation	Actual time a calculation takes
Time, model	Time (span) simulated (by the model)
Time, simulation	Actual time a simulation run takes

Appendix B

Monte Carlo Simulation

B.1 Specifications of Simulation Server

The simulations presented in this work were performed on a server with the following specifications:

Server Hardware Intel S5000 Server System, 2x Quad Core Intel Xeon E5335 @ 2GHz, 8 GB of RAM

Operation System Ubuntu Linux (server-optimised SMP 2.6.26), x86_64 kernel and 64-bit user mode

Java Interpreter Java SE Runtime Environment (build 1.6.0_14-b08), HotSpot 64-Bit Server VM

B.2 Key of Log File

				BLMS					BIS	
1	2	3	4	5	6	7	8	9	10	11
mission	event type	trigger type	model time	rack index	vmecrate index ***	bee index	fee index	monitor index	component type	component index
1	*	0/1	getTime()	0	0	0	0	0	**	0
2	*	Main.triggerType	getTime()	numberOfRack-1	numberOfVME-1	numberOfBEE-1	numberOfFEE-1	numberOfMonitor-1	**	numberOfCIBUS numberOfBICbeam1 numberOfBICbeam2
3	*		getTime()						**	
4	*		getTime()						**	
5	*		getTime()						**	
6	*		getTime()						**	
7	*		getTime()						**	
8	*		getTime()						**	

Event type table (*)

Event	Event Code
beamLoss	0
falseGenerator	1
blind	2
missedDumpRequest	3
dump	4
missedDump	5
masked	6
beamLoss event	7

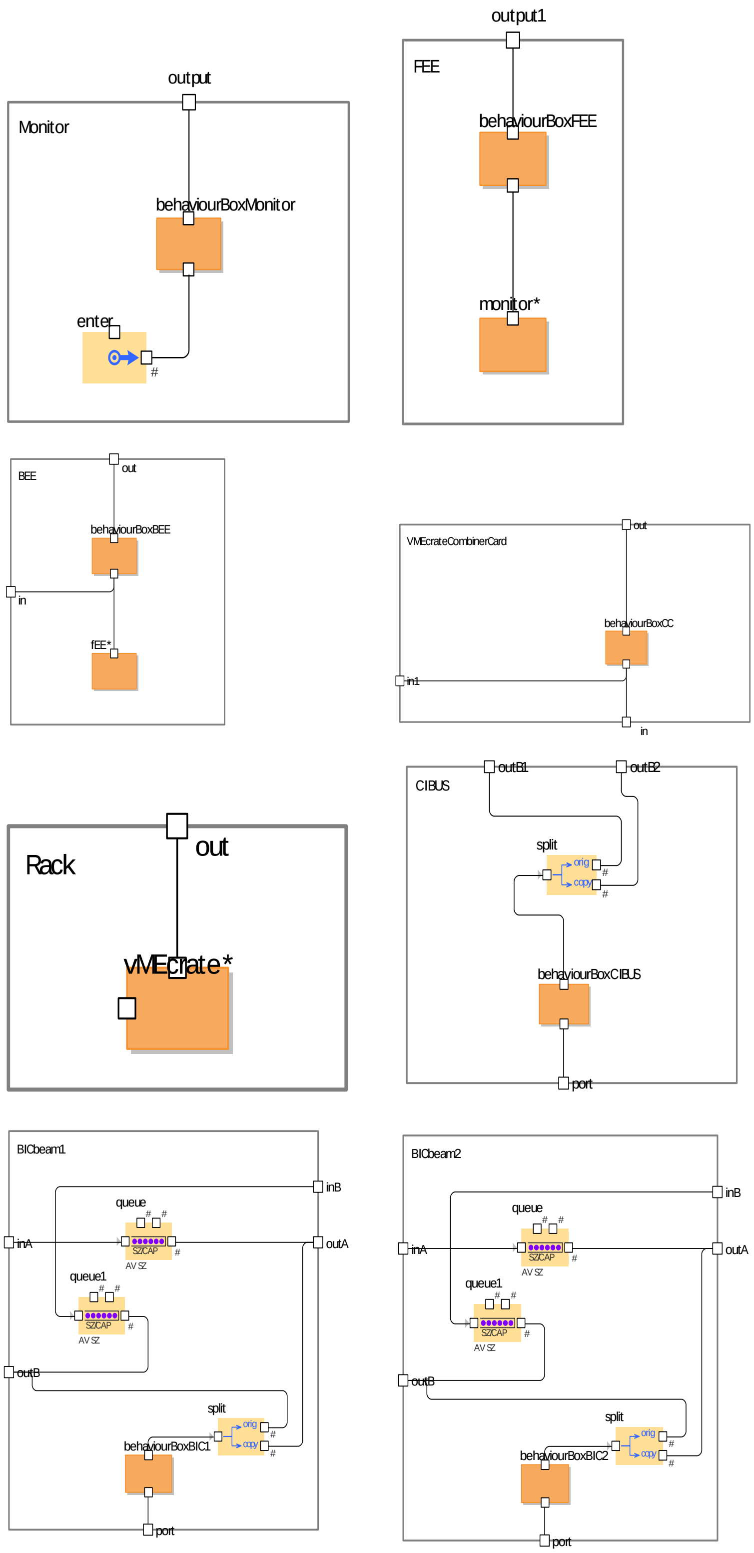
(***) index "-1" relates to CC!

Component type table (**)

Component	Component Code
CIBUS	0
BICbeam1	1
BICbeam2	2

B.3 Component Objects

Name	**Encapsulated objects**	Related to component
Monitor	BehaviourBoxMonitor	Monitor
FEE	BehaviourBoxFEE	FEE
BEE	BehaviourBoxBEE	BEE
VMEcrateCombinerCard	BehaviourBoxCC	CC
VMEcrate	BehaviourBoxCrate	VME
Rack	-	-
CIBUS	BehabiourBoxCIBUS	CIBUS
BICbeam1	BehaviourBoxBIC1	BIC1
BICbeam2	BehaviourBoxBIC2	BIC2



output
Monitor
behaviourBoxMonitor
enter
output1
FEE
behaviourBoxFEE
monitor*
BEE
out
behaviourBoxBEE
in
fEE*
VMEcrateCombinerCard
out
behaviourBoxCC
in1
in
Rack
out
vMEcrate*
CIBUS
outB1
outB2
split
orig
copy
behaviourBoxCIBUS
port
BICbeam1
inB
queue
inA
outA
queue1
outB
split
behaviourBoxBIC1
port
BICbeam2
inB
queue
inA
outA
queue1
outB
split
behaviourBoxBIC2
port

B.4 Auxiliary Model Parameters

Name	Type	Default value	Description
mission	integer	0	Mission count
triggerType	integer	Main.ET_NONE	Used for data logging
maxMissions	integer	150	# missions to be simulated
dump	boolean	false	Used for stop condition
missedDump	boolean	false	Used for stop condition
trigger	boolean	false	Used for stop condition
numberOfBICbeam1	integer	16	# BIC1s
numberOfBICbeam2	integer	16	# BIC2s
numberOfRack	integer	8	# Racks
numberOfVME	integer	3	# VMEs per Rack
numberOfCIBUS	integer	8	# CIBUSs
numberOfBEE	integer	13	# BEEs per VME
numberOfFEE	integer	2	# FEEs per BEE
numberOfMonitor	integer	6	# Monitors per FEE

B.5 Program Code

The provided program code is the original code of the model implementation used for the presented case studies. The structure evolved over the different phases of the implementation. The author is aware of the optimatisation potential of the code particularly in terms of a more compact programming.

B.5.1 System Demand Pattern of MPS Model

Expiry action of AnyLogic timer object beamLossTimer:

```
int m = uniform_discr(rack.item(0).vMEcrate.size()-1);
int i = uniform_discr(rack.item(0).vMEcrate.item(m).bEE.size()-1);
int j = uniform_discr(1);

if (Main.trigger == false) {
  if (j == 0) {
    for (int k=0; k<Main.numberOfMonitor; k++ ){
```

```
      Signal beam_loss = new Signal();
      rack.item(0).vMEcrate.item(m).bEE.item(i).fEE.item(0).monitor.item(k).
       enter.take(beam_loss);
      Main.trigger = true;
      Main.triggerType = Main.ET_BEAMLOSS;
    }
  }
  if (j == 1) {
   for (int k=0; k<Main.numberOfMonitor; k++ ){
      Signal beam_loss = new Signal();
      rack.item(0).vMEcrate.item(m).bEE.item(i).fEE.item(1).monitor.item(k).
       enter.take(beam_loss);
      Main.trigger = true;
      Main.triggerType = Main.ET_BEAMLOSS;
    }
  }
} else {
  Main.missedDump = true;
}
```

B.5.2 Monitor Masking Pattern

Program code in *Main*, defining the masking of two-thirds of the monitors:

```
//masking 4 of 6 Monitors, 2/3 and 5/6
for (int r=0; r<numberOfRack;r++) {
  for (int v=0; v<numberOfVME; v++) {
    for (int b=0; b<numberOfBEE; b++) {
      for (int f=0; f<numberOfFEE; f++) {
        rack.item(r).vMEcrate.item(v).bEE.item(b).fEE.item(f).monitor.
         item(1).behaviourBoxMonitor.statechartStateObject.
          fireEvent("Monitor_masked");
        rack.item(r).vMEcrate.item(v).bEE.item(b).fEE.item(f).monitor.
         item(2).behaviourBoxMonitor.statechartStateObject.
          fireEvent("Monitor_masked");
        rack.item(r).vMEcrate.item(v).bEE.item(b).fEE.item(f).monitor.
         item(4).behaviourBoxMonitor.statechartStateObject.
          fireEvent("Monitor_masked");
        rack.item(r).vMEcrate.item(v).bEE.item(b).fEE.item(f).monitor.
         item(5).behaviourBoxMonitor.statechartStateObject.
          fireEvent("Monitor_masked");
```

```
      }
    }
  }
}
```

B.5.3 Weighted System Demand Pattern

Expiry action of AnyLogic timer object beamLossTimer:

```
if (Main.trigger == false) {

 Main.trigger = true;
 Main.triggerType = Main.ET_BEAMLOSS;

int n =uniform_discr(9); //number of affected magnets, 1-10

int numberAffMagnets = n+1;
statslog.println(Main.mission + " "
    + Main.ET_LOSSEVENT + " "
    + numberAffMagnets + " "
    + getTime() + " "
    + Main.ET_NONE + " "
    + Main.ET_NONE + " "
    + Main.ET_NONE + " "
    + Main.ET_NONE + " "
    + Main.ET_NONE + " "
    + Main.ET_NONE + " "
    + Main.ET_NONE);

for(int a=0; a<=n; a++) {
  int nn =uniform_discr(rack.size()-1); //affected rack
  int m =uniform_discr(rack.item(nn).vMEcrate.size()-1); //affected crate
  int i = uniform_discr(rack.item(nn).vMEcrate.item(m).bEE.size()-1);
   //affected BEE
  int j = uniform_discr(1); //affected FEE
  int s = uniform_discr(1); //affected beam
  int k = uniform_discr(9);
   //number of affected monitors, in 60%: 2, in 40%: 1

  if (s==0) { //beam 1
  if (k>5) { //1 monitor affected
```

```
    int b=uniform_discr(9);

    if (b<=6) { //70 % monitor 1
        Entity beam_loss =  new Entity();
        rack.item(nn).vMEcrate.item(m).bEE.item(i).fEE.item(j).monitor.
         item(0).enter.take(beam_loss);
    }

    if (b>6 && b<=8) {//20% monitor 2
        Entity beam_loss =  new Entity();
        rack.item(nn).vMEcrate.item(m).bEE.item(i).fEE.item(j).monitor.
         item(1).enter.take(beam_loss);
       }

    if (b==9) {//10% monitor 3
        Entity beam_loss =  new Entity();
        rack.item(nn).vMEcrate.item(m).bEE.item(i).fEE.item(j).monitor.
         item(2).enter.take(beam_loss);
      }
    }

  if (k<=5) { // 2 monitors affected
    int b=uniform_discr(9);

      if (b<=8) { //90 % monitors 1/2
            Entity beam_loss =  new Entity();
            rack.item(nn).vMEcrate.item(m).bEE.item(i).fEE.item(j).monitor.
               item(0).enter.take(beam_loss);
            Entity beam_lossTwo =  new Entity();
            rack.item(nn).vMEcrate.item(m).bEE.item(i).fEE.item(j).monitor.
               item(1).enter.take(beam_lossTwo);

      }

      if (b==9) {//10% monitors 2/3
           Entity beam_loss =  new Entity();
           rack.item(nn).vMEcrate.item(m).bEE.item(i).fEE.item(j).monitor.
            item(1).enter.take(beam_loss);
           Entity beam_lossTwo =  new Entity();
           rack.item(nn).vMEcrate.item(m).bEE.item(i).fEE.item(j).monitor.
            item(2).enter.take(beam_lossTwo);
      }
    }
  }
```

```
if (s==1) {// beam 2
if (k>5) { //nur 1 IC betroffen
  int b=uniform_discr(9);

  if (b<=6) { //70 % IC 4
    //FEE
      Entity beam_loss =  new Entity();
      rack.item(nn).vMEcrate.item(m).bEE.item(i).fEE.item(j).monitor.
       item(3).enter.take(beam_loss);
     }

  if (b>6 && b<=8) {//20% IC 5
        Entity beam_loss =  new Entity();
        rack.item(nn).vMEcrate.item(m).bEE.item(i).fEE.item(j).monitor.
          item(4).enter.take(beam_loss);
     }

  if (b==9) {//10% IC 6
         Entity beam_loss =  new Entity();
         rack.item(nn).vMEcrate.item(m).bEE.item(i).fEE.item(j).monitor.
          item(5).enter.take(beam_loss);
    }
  }

if (k<=5) { // zwei ICs betroffen
  int b=uniform_discr(9);

    if (b<=8) { //90 % IC 4/5
          Entity beam_loss =  new Entity();
          rack.item(nn).vMEcrate.item(m).bEE.item(i).fEE.item(j).monitor.
             item(3).enter.take(beam_loss);
          Entity beam_lossTwo =  new Entity();
          rack.item(nn).vMEcrate.item(m).bEE.item(i).fEE.item(j).monitor.
             item(4).enter.take(beam_lossTwo);
    }

    if (b==9) {//10% IC 5/6
         Entity beam_loss =  new Entity();
         rack.item(nn).vMEcrate.item(m).bEE.item(i).fEE.item(j).monitor.
          item(4).enter.take(beam_loss);
         Entity beam_lossTwo =  new Entity();
         rack.item(nn).vMEcrate.item(m).bEE.item(i).fEE.item(j).monitor.
          item(5).enter.take(beam_lossTwo);
    }
  }
```

```
  }
}

}else {
  Main.missedDump = true;
}
```

B.5.4 Failure Pattern CCF BEAM ENERGY SIGNAL

Expiry action of AnyLogic timer object b_energy:

```
//BEEs of all crates in rack 0 affected

for (int v=0; v<Main.numberOfVME; v++ ){
   //whole rack 0 blind (each BEE set blind)
  for (int b=0; b<13; b++ ){
    rack.item(0).vMEcrate.item(v).bEE.item(b).behaviourBoxBEE.
      statechartStateObject.fireEvent("blindEnergy");
  }
}
```

Expiry action of Anylogic timer object f_energy:

```
//BEE 12 of crate 2 (last) in rack 0 affected

      rack.item(0).vMEcrate.item(2).bEE.item(12).behaviourBoxBEE.
         statechartStateObject.fireEvent("falseEnergy");
```

B.5.5 Failure Pattern CCF SAFE BEAM FLAG

Expiry action of AnyLogic timer object b_SBF:

```
// all 16 BIC pairs affected

  for (int i = 0; i < bICbeam1.size(); i++) {
    bICbeam1.item(i).behaviourBoxBIC1.statechartStateObject.
```

```
      fireEvent("blindSBF");
    bICbeam2.item(i).behaviourBoxBIC2.statechartStateObject.
      fireEvent("blindSBF");
}
```

Appendix C

Analytical Description

C.1 Specifications of Calculation Server

The calculation presented in this work were performed on a server with the following specifications:

Computer Hardware Intel Xeon CPU E5450 @ 3 GHz, 8 GB of RAM

Operation System Microsoft Windows Server 2003 R2, Enterprise x64 Edition, Service Pack 2

C.2 Auxiliary Model Parameters

Name	**Default value**	**Description**
num_cg	8	# component classes
n	[3744, 624, 312, 24, 24, 8, 16, 16]	# components n_i in class i, i=1..8
Monitor	6	# Monitors per FEE
FEE	2	# FEEs per BEE
BEE	13	# BEEs per VME
CombinerCard	1	# CC per VME
VMEcrate	3	# VME per Rack
Rack	8	# Racks
CIBUS	1	# CIBUS per Rack
BICbeam1	16	# BIC1
BICbeam2	16	# BIC2

C.3 Work Files

The following sections contain the essential parts of the indicated work files. The presented files result from a collaboration with Nibali [54].

C.3.1 Constants.txt

```
x := 0.01;
num_cg := 8;
lb := [1e-7, 1e-8, 1e-9, 1e-9, 1e-8, 1e-13, 1e-13, 1e-13];
lf := [1e-7, 1e-6, 1e-8, 1e-8, 1e-5, 1e-6, 1e-5, 1e-5];
n := [3744, 624, 312, 24, 24, 8, 16, 16];
Monitor := 6;
FEE := 2;
BEE := 13;
CombinerCard := 1;
Rack := 8;
CIBUS := 1;
BICbeam1 := 16;
BICbeam2 := 16;
VMEcrate := 3;
t0 := 0;
tf := 12;
```

C.3.2 TermsAndFunctions.txt

```
termD := proc()
  local term := x*exp(-x*t)/(VMEcrate*BEE*FEE);
  return term;
end proc:

termND := proc()
  local term := exp(-x*t);
  return term;
end proc:

termF := proc(cg)
  local term := lf[cg]*exp(-(lf[cg]+lb[cg])*t);
  return term;
end proc:
```

```
termB := proc(cg)
  local term := lb[cg]/(lb[cg]+lf[cg])*(1-exp(-(lb[cg]+lf[cg])*t));
  return term;
end proc:

termIB := proc(cg,pot)
  local term := exp(-(lb[cg]+lf[cg])*t*pot);
  return term;
end proc:

termEX := proc(cg,pot)
  local term := ((lb[cg]+lf[cg]*exp(-(lb[cg]+lf[cg])*t))/
      (lb[cg]+lf[cg]))^pot;
  return term;
end proc:
```

C.3.3 B7_not_demand_false.txt

```
myProbFuncB7 := proc(digits, precision, intMethod:="n/a")
  local i;
  local prob := 1.0;
  local st;
  local rtime;
  local numEvent;
  local nterms := 4768;

  NumericStatus(false);
  for i from 1 to num_cg do
    prob := prob * termEX(i,n[i]);
  end do;

  st := time();
  prob := evalf[digits](subs(t=tf,(termND() * prob)));
  rtime := time()-st;
  numEvent := checkNumEvent();

  return [prob, nterms, rtime, digits, precision, intMethod,
      [numEvent, NumericEventLocation]];
end proc:
```

C.3.4 B1_demand_triggered.txt

```
myProbFuncB1 := proc(digits, precision, intMethod:=_DEFAULT)
  local i;
  local t1;
  local integrand := 1.0;
  local prob;
  local nterms := 4768;
  local rtime;
  local numEvent;

  for i from 1 to num_cg do
    integrand := integrand * termEX(i,n[i]);
  end do;

  # Note: x*termND = termD without factor
  integrand := termND() * integrand;
  NumericStatus(false);
  t1 := time();
  prob := x * evalf[digits](Int(unapply(integrand,t),
      t0 .. tf, epsilon = 1.*10^(-precision), method=intMethod));
  rtime := time() - t1;
  numEvent := checkNumEvent();

  return [prob,nterms,rtime,digits,precision,intMethod,
      [numEvent,NumericEventLocation]];
end proc:
```

C.3.5 B2plus_class_false.txt

```
myProbFuncB2plus := proc(digits, precision, intMethod:=_DEFAULT)
local i;
local st;
local int;
local expo;
local numEvent;
local integrand := Vector[row](num_cg, 1.0);
local rtimecomp := Vector[row](num_cg);
local probcomp := Vector[row](num_cg);
local nterms := Vector[row](num_cg, 4768);
local prob := Vector[row](num_cg+1);
```

```
local rtime := Vector[row](num_cg+1);

for int from 1 to num_cg do
  for i from 1 to num_cg do
    if (i=int) then
      expo:=n[i]-1;
    else
      expo:=n[i];
    end if;
    integrand[int] := integrand[int] * termEX(i,expo);
  end do;
  integrand[int] := lf[int]*exp(-(lf[int]+lb[int]+x)*t)*integrand[int];
end do;

NumericStatus(false);
for i from 1 to num_cg do
  st := time();
  probcomp[i] := n[i] * evalf[digits](Int(unapply(integrand[i], t),
      t0 .. tf, epsilon = 1.*10^(-precision), method=intMethod));
  rtimecomp[i] := time() - st;
end do;

numEvent := checkNumEvent();

prob := convert(Vector[row]([add(probcomp[i], i = 1 .. num_cg),
      probcomp]), listlist);
rtime := convert(Vector[row]([add(rtimecomp[i], i = 1 .. num_cg),
      rtimecomp]), listlist);
nterms := convert(nterms, listlist);

return [prob, nterms, rtime, digits, precision, intMethod,
      [numEvent, NumericEventLocation]];
end proc:
```

C.3.6 B5A_demand_end.txt

```
myProbFuncB5A := proc(digits, precision, intMethod:=_DEFAULT)
local prob;
local i;
local integrand := 0.0;
local numEvent;
local nterms := 4768;
local link2bee;
```

```
local link2cc;
local st;
local rtime;

# D --> E
for link2cc from 1 to VMEcrate do
  for link2bee from 1 to BEE do
    integrand := integrand + FEE *
        termD() *
        ((termIB(1,1) + termB(1))**Monitor - termB(1)**Monitor) *
        termIB(2,1) *
      termIB(3,BEE-link2bee+1) *
      termIB(4,VMEcrate-link2cc+1) *
      termIB(5,VMEcrate-link2cc+1) *
      termIB(6,1) *
      (termIB(7,1) * termIB(8,1) + termIB(7,1) * termB(8) +
            termB(7) * termIB(8,1)) *
        termEX(1,n[1]-Monitor) *
      termEX(2,n[2]-1) *
      termEX(3,n[3]-(BEE-link2bee+1)) *
      termEX(4,n[4]-(VMEcrate-link2cc+1)) *
      termEX(5,n[5]-(VMEcrate-link2cc+1)) *
      termEX(6,n[6]-1) *
      termEX(7,n[7]-1) *
      termEX(8,n[8]-1);
  end do; #link2bee
end do; #link2cc

NumericStatus(false);
st := time();
prob := evalf[digits](Int(unapply(integrand, t), t0 .. tf,
      epsilon = 1.*10^(-precision), method=intMethod));
rtime := time() - st;
numEvent := checkNumEvent();

return [prob, nterms, rtime, digits, precision, intMethod,
      [numEvent,NumericEventLocation]];
end proc:
```

C.3.7 B5B_false_end.txt

```
myProbFuncB5B := proc(digits, precision, intMethod:=_DEFAULT)
local i;
```

```
local vmecrate;
local cc;
local bee;
local ic2bee;
local pos_f;
local link2bee;
local link2cc;
local st;
local numEvent;
local integrand := Vector[row](num_cg);
local rtimecomp := Vector[row](num_cg);
local probcomp := Vector[row](num_cg);
local nterms := Vector[row](num_cg, 4768);
local prob := Vector[row](num_cg+1);
local rtime := Vector[row](num_cg+1);

for pos_f from 1 to num_cg do

# 1 --> E
if (pos_f=1) then
  for link2cc from 1 to VMEcrate do
    for link2bee from 1 to BEE do
      integrand[pos_f] := integrand[pos_f] +
      Rack * FEE * Monitor *
        termF(pos_f) * termND() *
        termIB(2,1) *
      termIB(3,BEE-link2bee+1) *
      termIB(4,VMEcrate-link2cc+1) *
      termIB(5,VMEcrate-link2cc+1) *
      termIB(6,1) *
      (termIB(7,1) * termIB(8,1) + termIB(7,1) * termB(8,1) +
            termB(7,1) * termIB(8,1)) *
        termEX(1,n[1]-1) *
      termEX(2,n[2]-1) *
      termEX(3,n[3]-(BEE-link2bee+1)) *
      termEX(4,n[4]-(VMEcrate-link2cc+1)) *
      termEX(5,n[5]-(VMEcrate-link2cc+1)) *
      termEX(6,n[6]-1) *
      termEX(7,n[7]-1) *
      termEX(8,n[8]-1);
    end do; #link2bee
  end do; #link2cc

# 2 --> E
elif (pos_f=2) then
```

```
  for link2cc from 1 to VMEcrate do
    for link2bee from 1 to BEE do
      integrand[pos_f] := integrand[pos_f] +
      Rack * FEE *
        termF(pos_f) * termND() *
        termIB(3,BEE-link2bee+1) *
      termIB(4,VMEcrate-link2cc+1) *
      termIB(5,VMEcrate-link2cc+1) *
      termIB(6,1) *
      (termIB(7,1) * termIB(8,1) + termIB(7,1) * termB(8,1) +
            termB(7,1) * termIB(8,1)) *
        termEX(1,n[1]) *
      termEX(2,n[2]-1) *
      termEX(3,n[3]-(BEE-link2bee+1)) *
      termEX(4,n[4]-(VMEcrate-link2cc+1)) *
      termEX(5,n[5]-(VMEcrate-link2cc+1)) *
      termEX(6,n[6]-1) *
      termEX(7,n[7]-1) *
      termEX(8,n[8]-1);
    end do; #link2bee
  end do; #link2cc

# 3 --> E
elif (pos_f=3) then
  for link2cc from 1 to VMEcrate do
    for bee from 1 to BEE do
      integrand[pos_f] := integrand[pos_f] +
      Rack *
        termF(pos_f) * termND() *
        termIB(3,BEE-bee) *
      termIB(4,VMEcrate-link2cc+1) *
      termIB(5,VMEcrate-link2cc+1) *
      termIB(6,1) *
      (termIB(7,1) * termIB(8,1) + termIB(7,1) * termB(8,1)
            + termB(7,1) * termIB(8,1)) *
        termEX(1,n[1]) *
      termEX(2,n[2]) *
      termEX(3,n[3]-(BEE-bee+1)) *
      termEX(4,n[4]-(VMEcrate-link2cc+1)) *
      termEX(5,n[5]-(VMEcrate-link2cc+1)) *
      termEX(6,n[6]-1) *
      termEX(7,n[7]-1) *
      termEX(8,n[8]-1);
    end do; #bee
  end do; #link2cc
```

```
# 4 --> E
elif (pos_f=4) then
  for cc from 1 to VMEcrate do
    integrand[pos_f] := integrand[pos_f] +
     Rack *
      termF(pos_f) * termND() *
      termIB(4,VMEcrate-cc) *
     termIB(5,VMEcrate-cc+1) *
     termIB(6,1) *
     (termIB(7,1) * termIB(8,1) + termIB(7,1) * termB(8,1)
         + termB(7,1) * termIB(8,1)) *
      termEX(1,n[1]) *
     termEX(2,n[2]) *
     termEX(3,n[3]) *
     termEX(4,n[4]-(VMEcrate-cc+1)) *
     termEX(5,n[5]-(VMEcrate-cc+1)) *
     termEX(6,n[6]-1) *
     termEX(7,n[7]-1) *
     termEX(8,n[8]-1);
  end do; #cc

# 5 --> E
elif (pos_f=5) then
  for vmecrate from 1 to VMEcrate do
    integrand[pos_f] := integrand[pos_f] +
     Rack *
      termF(pos_f) * termND() *
      termIB(4,VMEcrate-vmecrate) *
     termIB(5,VMEcrate-vmecrate) *
     termIB(6,1) *
     (termIB(7,1) * termIB(8,1) + termIB(7,1) * termB(8,1) +
         termB(7,1) * termIB(8,1)) *
      termEX(1,n[1]) *
     termEX(2,n[2]) *
     termEX(3,n[3]) *
     termEX(4,n[4]-(VMEcrate-vmecrate)) *
     termEX(5,n[5]-(VMEcrate-vmecrate+1)) *
     termEX(6,n[6]-1) *
     termEX(7,n[7]-1) *
     termEX(8,n[8]-1);
  end do; #vmecrate

# 6 --> E
elif (pos_f=6) then
```

```
  integrand[pos_f] := integrand[pos_f] +
   Rack *
    termF(pos_f) * termND() *
    (termIB(7,1) * termIB(8,1) + termIB(7,1) * termB(8,1) +
         termB(7,1) * termIB(8,1)) *
    termEX(1,n[1]) *
   termEX(2,n[2]) *
   termEX(3,n[3]) *
   termEX(4,n[4]) *
   termEX(5,n[5]) *
   termEX(6,n[6]-1) *
   termEX(7,n[7]-1) *
   termEX(8,n[8]-1);

# 7 --> E
elif (pos_f=7) then
  integrand[pos_f] := integrand[pos_f] +
   n[7] *
    termF(pos_f) * termND() *
    termEX(1,n[1]) *
   termEX(2,n[2]) *
   termEX(3,n[3]) *
   termEX(4,n[4]) *
   termEX(5,n[5]) *
   termEX(6,n[6]) *
   termEX(7,n[7]-1) *
   termEX(8,n[8]);

# 8 --> E
elif (pos_f=8) then
  integrand[pos_f] := integrand[pos_f] +
   n[8] *
    termF(pos_f) * termND() *
    termEX(1,n[1]) *
   termEX(2,n[2]) *
   termEX(3,n[3]) *
   termEX(4,n[4]) *
   termEX(5,n[5]) *
   termEX(6,n[6]) *
   termEX(7,n[7]) *
   termEX(8,n[8]-1);

end if;
end do; #pos_f
```

```
NumericStatus(false);
for i from 1 to num_cg do
  st := time();
  probcomp[i] := evalf[digits](Int(unapply(integrand[i], t), t0 .. tf,
      epsilon = 1.*10^(-precision), method=intMethod));
  rtimecomp[i] := time() - st;
end do;

numEvent := checkNumEvent();

prob := convert(Vector[row]([add(probcomp[i], i = 1 .. num_cg),
      probcomp]), listlist);
rtime := convert(Vector[row]([add(rtimecomp[i], i = 1 .. num_cg),
      rtimecomp]), listlist);
nterms := convert(nterms, listlist);

return [prob, nterms, rtime, digits, precision, intMethod,
      [numEvent, NumericEventLocation]];
end proc:
```

C.3.8 B6A_demand_absorbed.txt

```
myProbFuncB6A := proc(digits, precision, intMethod:=_DEFAULT)
local i;
local vmecrate;
local cc;
local bee;
local ic2bee;
local pos_f;
local pos_b;
local link2bee;
local link2cc;
local Bbee;
local Fbee;
local Bcc;
local Fcc;
local Bvmecrate;
local Fvmecrate;
local st;
local numEvent;
local integrand := Vector[row](num_cg);
local rtimecomp := Vector[row](num_cg);
local probcomp := Vector[row](num_cg);
```

```
local nterms := Vector[row](num_cg, 4768);
local prob := Vector[row](num_cg+1);
local rtime := Vector[row](num_cg+1);

NumericStatus(false);
for pos_b from 1 to num_cg do

# D --> 1
if (pos_b=1) then
  integrand[pos_b] := integrand[pos_b] +
   VMEcrate * BEE * FEE *
    termB(pos_b)**Monitor * termD() *
    termEX(1,n[1]-Monitor) *
   termEX(2,n[2]) *
   termEX(3,n[3]) *
   termEX(4,n[4]) *
   termEX(5,n[5]) *
   termEX(6,n[6]) *
   termEX(7,n[7]) *
   termEX(8,n[8]);

# D --> 2
elif (pos_b=2) then
  integrand[pos_b] := integrand[pos_b] +
   VMEcrate * BEE * FEE *
    termB(pos_b) * termD() *
    ((termIB(1,1) + termB(1))**Monitor - termB(1)**Monitor) *
    termEX(1,n[1]-Monitor) *
   termEX(2,n[2]-1) *
   termEX(3,n[3]) *
   termEX(4,n[4]) *
   termEX(5,n[5]) *
   termEX(6,n[6]) *
   termEX(7,n[7]) *
   termEX(8,n[8]);

# D --> 3
elif (pos_b=3) then
  for bee from 1 to BEE do
    for link2bee from 1 to bee do
      integrand[pos_b] := integrand[pos_b] +
      VMEcrate * FEE *
        termB(pos_b) * termD() *
        ((termIB(1,1) + termB(1))**Monitor - termB(1)**Monitor) *
        termIB(2,1) *
```

```
      termIB(3,bee-link2bee) *
        termEX(1,n[1]-Monitor) *
      termEX(2,n[2]-1) *
      termEX(3,n[3]-(bee-link2bee+1)) *
      termEX(4,n[4]) *
      termEX(5,n[5]) *
      termEX(6,n[6]) *
      termEX(7,n[7]) *
      termEX(8,n[8]);
    end do; #link2bee
  end do; #bee

# D --> 4
elif (pos_b=4) then
  for cc from 1 to VMEcrate do
    for link2cc from 1 to cc do
      for link2bee from 1 to BEE do
        integrand[pos_b] := integrand[pos_b] +
        FEE *
          termB(pos_b) * termD()*
          ((termIB(1,1) + termB(1))**Monitor - termB(1)**Monitor) *
          termIB(2,1) *
        termIB(3,BEE-link2bee+1) *
        termIB(4,cc-link2cc) *
        termIB(5,cc-link2cc) *
          termEX(1,n[1]-Monitor) *
        termEX(2,n[2]-1) *
        termEX(3,n[3]-(BEE-link2bee+1)) *
        termEX(4,n[4]-(cc-link2cc+1)) *
        termEX(5,n[5]-(cc-link2cc)) *
        termEX(6,n[6]) *
        termEX(7,n[7]) *
        termEX(8,n[8]);
      end do; #linkbee
    end do; #link2cc
  end do; #cc

# D --> 5
elif (pos_b=5) then
  for vmecrate from 1 to VMEcrate do
    for link2cc from 1 to vmecrate do
      for link2bee from 1 to BEE do
        integrand[pos_b] := integrand[pos_b] +
        FEE *
          termB(pos_b) * termD() *
```

```
          ((termIB(1,1) + termB(1))**Monitor - termB(1)**Monitor) *
          termIB(2,1) *
        termIB(3,BEE-link2bee+1) *
        termIB(4,vmecrate-link2cc+1) *
        termIB(5,vmecrate-link2cc) *
          termEX(1,n[1]-Monitor) *
        termEX(2,n[2]-1) *
        termEX(3,n[3]-(BEE-link2bee+1)) *
        termEX(4,n[4]-(vmecrate-link2cc+1)) *
        termEX(5,n[5]-(vmecrate-link2cc+1)) *
        termEX(6,n[6]) *
        termEX(7,n[7]) *
        termEX(8,n[8]);
      end do; #link2bee
    end do; #link2cc
  end do; #vmecrate

# D --> 6
elif (pos_b=6) then
  for link2cc from 1 to VMEcrate do
    for link2bee from 1 to BEE do
      integrand[pos_b] := integrand[pos_b] +
      FEE *
        termB(pos_b) * termD() *
        ((termIB(1,1)+termB(1))**Monitor - termB(1)**Monitor) *
        termIB(2,1) *
      termIB(3,BEE-link2bee+1) *
      termIB(4,VMEcrate-link2cc+1) *
      termIB(5,VMEcrate-link2cc+1) *
        termEX(1,n[1]-Monitor) *
      termEX(2,n[2]-1) *
      termEX(3,n[3]-(BEE-link2bee+1)) *
      termEX(4,n[4]-(VMEcrate-link2cc+1)) *
      termEX(5,n[5]-(VMEcrate-link2cc+1)) *
      termEX(6,n[6]-1) *
      termEX(7,n[7]) *
      termEX(8,n[8]);
    end do; #link2bee
  end do; #link2cc

# D --> 7
elif (pos_b=7) then
  for link2cc from 1 to VMEcrate do
    for link2bee from 1 to BEE do
      integrand[pos_b] := integrand[pos_b] +
```

```
      FEE *
        termB(pos_b) * termB(pos_b + 1) * termD() *
        ((termIB(1,1) + termB(1))**Monitor - termB(1)**Monitor) *
        termIB(2,1) *
      termIB(3,BEE-link2bee+1) *
      termIB(4,VMEcrate-link2cc+1) *
      termIB(5,VMEcrate-link2cc+1) *
      termIB(6,1) *
        termEX(1,n[1]-Monitor) *
      termEX(2,n[2]-1) *
      termEX(3,n[3]-(BEE-link2bee+1)) *
      termEX(4,n[4]-(VMEcrate-link2cc+1)) *
      termEX(5,n[5]-(VMEcrate-link2cc+1)) *
      termEX(6,n[6]-1) *
      termEX(7,n[7]-1) *
      termEX(8,n[8]-1);
    end do; #link2bee
  end do; #link2cc

end if;
end do;

NumericStatus(false);
for i from 1 to num_cg do
  st := time();
  probcomp[i] := evalf[digits](Int(unapply(integrand[i], t), t0 .. tf,
      epsilon = 1.*10^(-precision), method=intMethod));
  rtimecomp[i] := time() - st;
end do;

numEvent := checkNumEvent();

prob := convert(Vector[row]([add(probcomp[i], i = 1 .. num_cg),
      probcomp]), listlist);
rtime := convert(Vector[row]([add(rtimecomp[i], i = 1 .. num_cg),
      rtimecomp]), listlist);
nterms := convert(nterms, listlist);

return [prob, nterms, rtime, digits, precision, intMethod,
      [numEvent, NumericEventLocation]];
end proc:
```

C.3.9 B6B_false_absorbed.txt

```
myProbFuncB6B := proc(digits, precision, intMethod:=_DEFAULT)
local i;
local vmecrate;
local cc;
local bee;
local ic2bee;
local pos_f;
local pos_b;
local link2bee;
local link2cc;
local Bbee;
local Fbee;
local Bcc;
local Fcc;
local Bvmecrate;
local Fvmecrate;
local st;
local numEvent;
local integrand := Vector[row](num_cg);
local rtimecomp := Vector[row](num_cg);
local probcomp := Vector[row](num_cg);
local nterms := Vector[row](num_cg, 4768);
local prob := Vector[row](num_cg+1);
local rtime := Vector[row](num_cg+1);

for pos_f from 1 to num_cg do
for pos_b from (pos_f - 1) to num_cg do

# 1 --> 2
if (pos_f=1 and pos_b=2) then
  integrand[pos_f] := integrand[pos_f] +
   Rack * VMEcrate * BEE * FEE * Monitor *
    termF(pos_f) * termB(pos_b) * termND() *
    termEX(1,n[1]-1) *
   termEX(2,n[2]-1) *
   termEX(3,n[3]) *
   termEX(4,n[4]) *
   termEX(5,n[5]) *
   termEX(6,n[6]) *
   termEX(7,n[7]) *
   termEX(8,n[8]);
```

```
# 1 --> 3
elif (pos_f=1 and pos_b=3) then
  for bee from 1 to BEE do
    for link2bee from 1 to bee do
      integrand[pos_f] := integrand[pos_f] +
      Rack * VMEcrate * FEE * Monitor *
        termF(pos_f) * termB(pos_b) * termND() *
        termIB(2,1) *
      termIB(3,bee-link2bee) *
        termEX(1,n[1]-1) *
      termEX(2,n[2]-1) *
      termEX(3,n[3]-(bee-link2bee+1)) *
      termEX(4,n[4]) *
      termEX(5,n[5]) *
      termEX(6,n[6]) *
      termEX(7,n[7]) *
      termEX(8,n[8]);
    end do; #link2bee
  end do; #bee

# 1 --> 4
elif (pos_f=1 and pos_b=4) then
  for cc from 1 to VMEcrate do
    for link2cc from 1 to cc do
      for link2bee from 1 to BEE do
        integrand[pos_f] := integrand[pos_f] +
        Rack * FEE * Monitor *
          termF(pos_f) * termB(pos_b) * termND()*
          termIB(2,1) *
        termIB(3,BEE-link2bee+1) *
        termIB(4,cc-link2cc) *
        termIB(5,cc-link2cc) *
          termEX(1,n[1]-1) *
        termEX(2,n[2]-1) *
        termEX(3,n[3]-(BEE-link2bee+1)) *
        termEX(4,n[4]-(cc-link2cc+1)) *
        termEX(5,n[5]-(cc-link2cc)) *
        termEX(6,n[6]) *
        termEX(7,n[7]) *
        termEX(8,n[8]);
      end do; #linkbee
    end do; #link2cc
  end do; #cc

# 1 --> 5
```

```
elif (pos_f=1 and pos_b=5) then
  for vmecrate from 1 to VMEcrate do
    for link2cc from 1 to vmecrate do
      for link2bee from 1 to BEE do
        integrand[pos_f] := integrand[pos_f] +
        Rack * FEE * Monitor *
          termF(pos_f) * termB(pos_b) * termND() *
          termIB(2,1) *
        termIB(3,BEE-link2bee+1) *
        termIB(4,vmecrate-link2cc+1) *
        termIB(5,vmecrate-link2cc) *
          termEX(1,n[1]-1) *
        termEX(2,n[2]-1) *
        termEX(3,n[3]-(BEE-link2bee+1)) *
        termEX(4,n[4]-(vmecrate-link2cc+1)) *
        termEX(5,n[5]-(vmecrate-link2cc+1)) *
        termEX(6,n[6]) *
        termEX(7,n[7]) *
        termEX(8,n[8]);
      end do; #link2bee
    end do; #link2cc
  end do; #vmecrate

# 1 --> 6
elif (pos_f=1 and pos_b=6) then
  for link2cc from 1 to VMEcrate do
    for link2bee from 1 to BEE do
      integrand[pos_f] := integrand[pos_f] +
      Rack * FEE * Monitor *
        termF(pos_f) * termB(pos_b) * termND() *
        termIB(2,1) *
      termIB(3,BEE-link2bee+1) *
      termIB(4,VMEcrate-link2cc+1) *
      termIB(5,VMEcrate-link2cc+1) *
        termEX(1,n[1]-1) *
      termEX(2,n[2]-1) *
      termEX(3,n[3]-(BEE-link2bee+1)) *
      termEX(4,n[4]-(VMEcrate-link2cc+1)) *
      termEX(5,n[5]-(VMEcrate-link2cc+1)) *
      termEX(6,n[6]-1) *
      termEX(7,n[7]) *
      termEX(8,n[8]);
    end do; #link2bee
  end do; #link2cc
```

```
# 1 --> 7
elif (pos_f=1 and pos_b=7) then
  for link2cc from 1 to VMEcrate do
    for link2bee from 1 to BEE do
      integrand[pos_f] := integrand[pos_f] +
      Rack * FEE * Monitor *
        termF(pos_f) * termB(pos_b) * termB(pos_b + 1) * termND() *
        termIB(2,1) *
      termIB(3,BEE-link2bee+1) *
      termIB(4,VMEcrate-link2cc+1) *
      termIB(5,VMEcrate-link2cc+1) *
      termIB(6,1) *
        termEX(1,n[1]-1) *
      termEX(2,n[2]-1) *
      termEX(3,n[3]-(BEE-link2bee+1)) *
      termEX(4,n[4]-(VMEcrate-link2cc+1)) *
      termEX(5,n[5]-(VMEcrate-link2cc+1)) *
      termEX(6,n[6]-1) *
      termEX(7,n[7]-1) *
      termEX(8,n[8]-1);
    end do; #link2bee
  end do; #link2cc

# 2 --> 3
elif (pos_f=2 and pos_b=3) then
  for bee from 1 to BEE do
    for link2bee from 1 to bee do
      integrand[pos_f] := integrand[pos_f] +
      Rack * VMEcrate * FEE *
        termF(pos_f) * termB(pos_b) * termND() *
        termIB(3,bee-link2bee) *
        termEX(1,n[1]) *
      termEX(2,n[2]-1) *
      termEX(3,n[3]-(bee-link2bee+1)) *
      termEX(4,n[4]) *
      termEX(5,n[5]) *
      termEX(6,n[6]) *
      termEX(7,n[7]) *
      termEX(8,n[8]);
    end do; #link2bee
  end do; #bee

# 2 --> 4
elif (pos_f=2 and pos_b=4) then
  for cc from 1 to VMEcrate do
```

```
    for link2cc from 1 to cc do
      for link2bee from 1 to BEE do
        integrand[pos_f] := integrand[pos_f] +
        Rack * FEE *
          termF(pos_f) * termB(pos_b) * termND() *
          termIB(3,BEE-link2bee+1) *
        termIB(4,cc-link2cc) *
        termIB(5,cc-link2cc)*
          termEX(1,n[1]) *
        termEX(2,n[2]-1) *
        termEX(3,n[3]-(BEE-link2bee+1)) *
        termEX(4,n[4]-(cc-link2cc+1)) *
        termEX(5,n[5]-(cc-link2cc)) *
        termEX(6,n[6]) *
        termEX(7,n[7]) *
        termEX(8,n[8]);
      end do; #link2bee
    end do; #link2cc
  end do; #cc

# 2 --> 5
elif (pos_f=2 and pos_b=5) then
  for vmecrate from 1 to VMEcrate do
    for link2cc from 1 to vmecrate do
      for link2bee from 1 to BEE do
        integrand[pos_f] := integrand[pos_f] +
        Rack * FEE *
          termF(pos_f) * termB(pos_b) * termND() *
          termIB(3,BEE-link2bee+1) *
        termIB(4,vmecrate-link2cc+1) *
        termIB(5,vmecrate-link2cc) *
          termEX(1,n[1]) *
        termEX(2,n[2]-1) *
        termEX(3,n[3]-(BEE-link2bee+1)) *
        termEX(4,n[4]-(vmecrate-link2cc+1)) *
        termEX(5,n[5]-(vmecrate-link2cc+1)) *
        termEX(6,n[6]) *
        termEX(7,n[7]) *
        termEX(8,n[8]);
      end do; #link2bee
    end do; #link2cc
  end do; #vmecrate

# 2 --> 6
elif (pos_f=2 and pos_b=6) then
```

```
  for link2cc from 1 to VMEcrate do
    for link2bee from 1 to BEE do
      integrand[pos_f] := integrand[pos_f] +
      Rack * FEE *
        termF(pos_f) * termB(pos_b) * termND() *
        termIB(3,BEE-link2bee+1) *
      termIB(4,VMEcrate-link2cc+1) *
      termIB(5,VMEcrate-link2cc+1) *
        termEX(1,n[1]) *
      termEX(2,n[2]-1) *
      termEX(3,n[3]-(BEE-link2bee+1)) *
      termEX(4,n[4]-(VMEcrate-link2cc+1)) *
      termEX(5,n[5]-(VMEcrate-link2cc+1)) *
      termEX(6,n[6]-1) *
      termEX(7,n[7]) *
      termEX(8,n[8]);
    end do; #link2bee
  end do; #link2cc

# 2 --> 7
elif (pos_f=2 and pos_b=7) then
  for link2cc from 1 to VMEcrate do
    for link2bee from 1 to BEE do
      integrand[pos_f] := integrand[pos_f] +
      Rack * FEE *
        termF(pos_f) * termB(pos_b) * termB(pos_b+1) * termND() *
        termIB(3,BEE-link2bee+1) *
      termIB(4,VMEcrate-link2cc+1) *
      termIB(5,VMEcrate-link2cc+1) *
      termIB(6,1)*
        termEX(1,n[1]) *
      termEX(2,n[2]-1) *
      termEX(3,n[3]-(BEE-link2bee+1)) *
      termEX(4,n[4]-(VMEcrate-link2cc+1)) *
      termEX(5,n[5]-(VMEcrate-link2cc+1)) *
      termEX(6,n[6]-1) *
      termEX(7,n[7]-1) *
      termEX(8,n[8]-1);
    end do; #link2bee
  end do; #link2cc

# 3 --> 3
elif (pos_f=3 and pos_b=3) then
  for Bbee from 2 to BEE do
    for Fbee from 1 to Bbee-1 do
```

```
      integrand[pos_f] := integrand[pos_f] +
      Rack * VMEcrate *
        termF(pos_f) * termB(pos_b) * termND() *
        termIB(3,Bbee-Fbee-1) *
        termEX(1,n[1]) *
      termEX(2,n[2]) *
      termEX(3,n[3]-(Bbee-Fbee+1)) *
      termEX(4,n[4]) *
      termEX(5,n[5]) *
      termEX(6,n[6]) *
      termEX(7,n[7]) *
      termEX(8,n[8]);
    end do; #Fbee
  end do; #Bbee

# 3 --> 4
elif (pos_f=3 and pos_b=4) then
  for cc from 1 to VMEcrate do
    for link2cc from 1 to cc do
      for bee from 1 to BEE do
        integrand[pos_f] := integrand[pos_f] +
        Rack *
          termF(pos_f) * termB(pos_b) * termND() *
          termIB(3,BEE-bee) *
        termIB(4,cc-link2cc) *
        termIB(5,cc-link2cc)*
          termEX(1,n[1]) *
        termEX(2,n[2]) *
        termEX(3,n[3]-(BEE-bee+1)) *
        termEX(4,n[4]-(cc-link2cc+1)) *
        termEX(5,n[5]-(cc-link2cc)) *
        termEX(6,n[6]) *
        termEX(7,n[7]) *
        termEX(8,n[8]);
      end do; #bee
    end do; #link2cc
  end do; #cc

# 3 --> 5
elif (pos_f=3 and pos_b=5) then
  for vmecrate from 1 to VMEcrate do
    for link2cc from 1 to vmecrate do
      for bee from 1 to BEE do
        integrand[pos_f] := integrand[pos_f] +
        Rack *
```

```
          termF(pos_f) * termB(pos_b) * termND() *
          termIB(3,BEE-bee) *
        termIB(4,vmecrate-link2cc+1) *
        termIB(5,vmecrate-link2cc) *
          termEX(1,n[1]) *
        termEX(2,n[2]) *
        termEX(3,n[3]-(BEE-bee+1)) *
        termEX(4,n[4]-(vmecrate-link2cc+1)) *
        termEX(5,n[5]-(vmecrate-link2cc+1)) *
        termEX(6,n[6]) *
        termEX(7,n[7]) *
        termEX(8,n[8]);
      end do; #bee
    end do; #link2cc
  end do; #vmecrate

# 3 --> 6
elif (pos_f=3 and pos_b=6) then
  for link2cc from 1 to VMEcrate do
    for bee from 1 to BEE do
      integrand[pos_f] := integrand[pos_f] +
      Rack *
        termF(pos_f) * termB(pos_b) * termND() *
        termIB(3,BEE-bee) *
      termIB(4,VMEcrate-link2cc+1) *
      termIB(5,VMEcrate-link2cc+1) *
        termEX(1,n[1]) *
      termEX(2,n[2]) *
      termEX(3,n[3]-(BEE-bee+1)) *
      termEX(4,n[4]-(VMEcrate-link2cc+1)) *
      termEX(5,n[5]-(VMEcrate-link2cc+1)) *
      termEX(6,n[6]-1) *
      termEX(7,n[7]) *
      termEX(8,n[8]);
    end do; #bee
  end do; #link2cc

# 3 --> 7
elif (pos_f=3 and pos_b=7) then
  for link2cc from 1 to VMEcrate do
    for bee from 1 to BEE do
      integrand[pos_f] := integrand[pos_f] +
      Rack *
        termF(pos_f) * termB(pos_b) * termB(pos_b+1) * termND() *
        termIB(3,BEE-bee) *
```

```
      termIB(4,VMEcrate-link2cc+1) *
      termIB(5,VMEcrate-link2cc+1) *
      termIB(6,1)*
        termEX(1,n[1]) *
      termEX(2,n[2]) *
      termEX(3,n[3]-(BEE-bee+1)) *
      termEX(4,n[4]-(VMEcrate-link2cc+1)) *
      termEX(5,n[5]-(VMEcrate-link2cc+1)) *
      termEX(6,n[6]-1) *
      termEX(7,n[7]-1) *
      termEX(8,n[8]-1);
    end do; #bee
  end do; #link2cc

# 4 --> 4
elif (pos_f=4 and pos_b=4) then
  for Bcc from 2 to VMEcrate do
    for Fcc from 1 to Bcc-1 do
      integrand[pos_f] := integrand[pos_f] +
      Rack *
        termF(pos_f) * termB(pos_b) * termND() *
        termIB(4,Bcc-Fcc-1) *
      termIB(5,Bcc-Fcc)*
        termEX(1,n[1]) *
      termEX(2,n[2]) *
      termEX(3,n[3]) *
      termEX(4,n[4]-(Bcc-Fcc+1)) *
      termEX(5,n[5]-(Bcc-Fcc)) *
      termEX(6,n[6]) *
      termEX(7,n[7]) *
      termEX(8,n[8]);
    end do; #Fcc
  end do; #Bcc

# 4 --> 5
elif (pos_f=4 and pos_b=5) then
  for vmecrate from 1 to VMEcrate do
    for cc from 1 to vmecrate do
      integrand[pos_f] := integrand[pos_f] +
      Rack *
        termF(pos_f) * termB(pos_b) * termND() *
        termIB(4,vmecrate-cc) *
      termIB(5,vmecrate-cc) *
        termEX(1,n[1]) *
      termEX(2,n[2]) *
```

```
      termEX(3,n[3]) *
      termEX(4,n[4]-(vmecrate-cc+1)) *
      termEX(5,n[5]-(vmecrate-cc+1)) *
      termEX(6,n[6]) *
      termEX(7,n[7]) *
      termEX(8,n[8]);
    end do; #cc
  end do; #vmecrate

# 4 --> 6
elif (pos_f=4 and pos_b=6) then
  for cc from 1 to VMEcrate do
    integrand[pos_f] := integrand[pos_f] +
     Rack *
      termF(pos_f) * termB(pos_b) * termND() *
      termIB(4,VMEcrate-cc) *
     termIB(5,VMEcrate-cc+1) *
      termEX(1,n[1]) *
     termEX(2,n[2]) *
     termEX(3,n[3]) *
     termEX(4,n[4]-(VMEcrate-cc+1)) *
     termEX(5,n[5]-(VMEcrate-cc+1)) *
     termEX(6,n[6]-1) *
     termEX(7,n[7]) *
     termEX(8,n[8]);
  end do; #cc

# 4 --> 7
elif (pos_f=4 and pos_b=7) then
  for cc from 1 to VMEcrate do
    integrand[pos_f] := integrand[pos_f] +
     Rack *
      termF(pos_f) * termB(pos_b) * termB(pos_b+1) * termND() *
      termIB(4,VMEcrate-cc) *
     termIB(5,VMEcrate-cc+1) *
     termIB(6,1) *
      termEX(1,n[1]) *
     termEX(2,n[2]) *
     termEX(3,n[3]) *
     termEX(4,n[4]-(VMEcrate-cc+1)) *
     termEX(5,n[5]-(VMEcrate-cc+1)) *
     termEX(6,n[6]-1) *
     termEX(7,n[7]-1) *
     termEX(8,n[8]-1);
  end do; #cc
```

```
# 5 --> 4 (super tricky)
elif (pos_f=5 and pos_b=4) then
  for vmecrate from 1 to (VMEcrate-1) do
    for cc from (vmecrate+1) to VMEcrate do
      integrand[pos_f] := integrand[pos_f] +
      Rack *
        termF(pos_f) * termB(pos_b) * termND() *
        termIB(4,cc-vmecrate-1) *
      termIB(5,cc-vmecrate-1) *
        termEX(1,n[1]) *
      termEX(2,n[2]) *
      termEX(3,n[3]) *
      termEX(4,n[4]-(cc-vmecrate)) *
      termEX(5,n[5]-(cc-vmecrate)) *
      termEX(6,n[6]) *
      termEX(7,n[7]) *
      termEX(8,n[8]);
    end do; #cc
  end do; #vmecrate

# 5 --> 5
elif (pos_f=5 and pos_b=5) then
  for Bvmecrate from 2 to VMEcrate do
    for Fvmecrate from 1 to Bvmecrate-1 do
      integrand[pos_f] := integrand[pos_f] +
      Rack *
        termF(pos_f) * termB(pos_b) * termND() *
        termIB(4,Bvmecrate-Fvmecrate) *
      termIB(5,Bvmecrate-Fvmecrate-1) *
        termEX(1,n[1]) *
      termEX(2,n[2]) *
      termEX(3,n[3]) *
      termEX(4,n[4]-(Bvmecrate-Fvmecrate)) *
      termEX(5,n[5]-(Bvmecrate-Fvmecrate+1)) *
      termEX(6,n[6]) *
      termEX(7,n[7]) *
      termEX(8,n[8]);
    end do; #Fvmecrate
  end do; #Bvmecrate

# 5 --> 6
elif (pos_f=5 and pos_b=6) then
  for vmecrate from 1 to VMEcrate do
    integrand[pos_f] := integrand[pos_f] +
```

```
      Rack *
       termF(pos_f) * termB(pos_b) * termND() *
       termIB(4,VMEcrate-vmecrate) *
      termIB(5,VMEcrate-vmecrate) *
       termEX(1,n[1]) *
      termEX(2,n[2]) *
      termEX(3,n[3]) *
      termEX(4,n[4]-(VMEcrate-vmecrate)) *
      termEX(5,n[5]-(VMEcrate-vmecrate+1)) *
      termEX(6,n[6]-1) *
      termEX(7,n[7]) *
      termEX(8,n[8]);
  end do; #vmecrate

# 5 --> 7
elif (pos_f=5 and pos_b=7) then
  for vmecrate from 1 to VMEcrate do
     integrand[pos_f] := integrand[pos_f] +
      Rack *
       termF(pos_f) * termB(pos_b) * termB(pos_b+1) * termND() *
       termIB(4,VMEcrate-vmecrate) *
      termIB(5,VMEcrate-vmecrate) *
      termIB(6,1) *
       termEX(1,n[1]) *
      termEX(2,n[2]) *
      termEX(3,n[3]) *
      termEX(4,n[4]-(VMEcrate-vmecrate)) *
      termEX(5,n[5]-(VMEcrate-vmecrate+1)) *
      termEX(6,n[6]-1) *
      termEX(7,n[7]-1) *
      termEX(8,n[8]-1);
  end do; #vmecrate

# 6 --> 7
elif (pos_f=6 and pos_b=7) then
  integrand[pos_f] := integrand[pos_f] +
   Rack *
    termF(pos_f) * termB(pos_b) * termB(pos_b+1) * termND() *
    termEX(1,n[1]) *
   termEX(2,n[2]) *
   termEX(3,n[3]) *
   termEX(4,n[4]) *
   termEX(5,n[5]) *
   termEX(6,n[6]-1) *
   termEX(7,n[7]-1) *
```

```
   termEX(8,n[8]-1);

end if;

end do; #pos_b
end do; #pos_f

NumericStatus(false);
for i from 1 to num_cg do
  st := time();
  probcomp[i] := evalf[digits](Int(unapply(integrand[i], t), t0 .. tf,
      epsilon = 1.*10^(-precision), method=intMethod));
  rtimecomp[i] := time() - st;
end do;

numEvent := checkNumEvent();

prob := convert(Vector[row]([add(probcomp[i], i = 1 .. num_cg),
      probcomp]), listlist);
rtime := convert(Vector[row]([add(rtimecomp[i], i = 1 .. num_cg),
      rtimecomp]), listlist);
nterms := convert(nterms, listlist);

return [prob, nterms, rtime, digits, precision, intMethod, [numEvent,
      NumericEventLocation]];
end proc:
```

C.3.10 B6X_false_absorbed.txt

```
myProbFuncB6X := proc(digits, precision, intMethod:=_DEFAULT)
local i;
local vmecrate;
local cc;
local bee;
local ic2bee;
local pos_f;
local pos_b;
local link2bee;
local link2cc;
local Bbee;
local Fbee;
local Bcc;
```

```
local Fcc;
local Bvmecrate;
local Fvmecrate;
local st;
local numEvent;
local integrand := Vector[row](num_cg);
local rtimecomp := Vector[row](num_cg);
local probcomp := Vector[row](num_cg);
local nterms := Vector[row](num_cg, 4768);
local prob := Vector[row](num_cg+1);
local rtime := Vector[row](num_cg+1);

for pos_f from 1 to num_cg do
for pos_b from (pos_f - 1) to num_cg do

# 1 --> 2
if (pos_f=1 and pos_b=2) then
  integrand[pos_b] := integrand[pos_b] +
   Rack * VMEcrate * BEE * FEE * Monitor *
    termF(pos_f) * termB(pos_b) * termND() *
    termEX(1,n[1]-1) *
   termEX(2,n[2]-1) *
   termEX(3,n[3]) *
   termEX(4,n[4]) *
   termEX(5,n[5]) *
   termEX(6,n[6]) *
   termEX(7,n[7]) *
   termEX(8,n[8]);

# 1 --> 3
elif (pos_f=1 and pos_b=3) then
  for bee from 1 to BEE do
    for link2bee from 1 to bee do
      integrand[pos_b] := integrand[pos_b] +
      Rack * VMEcrate * FEE * Monitor *
        termF(pos_f) * termB(pos_b) * termND() *
        termIB(2,1) *
      termIB(3,bee-link2bee) *
        termEX(1,n[1]-1) *
      termEX(2,n[2]-1) *
      termEX(3,n[3]-(bee-link2bee+1)) *
      termEX(4,n[4]) *
      termEX(5,n[5]) *
      termEX(6,n[6]) *
      termEX(7,n[7]) *
```

```
      termEX(8,n[8]);
    end do; #link2bee
  end do; #bee

# 1 --> 4
elif (pos_f=1 and pos_b=4) then
  for cc from 1 to VMEcrate do
    for link2cc from 1 to cc do
      for link2bee from 1 to BEE do
        integrand[pos_b] := integrand[pos_b] +
        Rack * FEE * Monitor *
          termF(pos_f) * termB(pos_b) * termND()*
          termIB(2,1) *
        termIB(3,BEE-link2bee+1) *
        termIB(4,cc-link2cc) *
        termIB(5,cc-link2cc) *
          termEX(1,n[1]-1) *
        termEX(2,n[2]-1) *
        termEX(3,n[3]-(BEE-link2bee+1)) *
        termEX(4,n[4]-(cc-link2cc+1)) *
        termEX(5,n[5]-(cc-link2cc)) *
        termEX(6,n[6]) *
        termEX(7,n[7]) *
        termEX(8,n[8]);
      end do; #linkbee
    end do; #link2cc
  end do; #cc

# 1 --> 5
elif (pos_f=1 and pos_b=5) then
  for vmecrate from 1 to VMEcrate do
    for link2cc from 1 to vmecrate do
      for link2bee from 1 to BEE do
        integrand[pos_b] := integrand[pos_b] +
        Rack * FEE * Monitor *
          termF(pos_f) * termB(pos_b) * termND() *
          termIB(2,1) *
        termIB(3,BEE-link2bee+1) *
        termIB(4,vmecrate-link2cc+1) *
        termIB(5,vmecrate-link2cc) *
          termEX(1,n[1]-1) *
        termEX(2,n[2]-1) *
        termEX(3,n[3]-(BEE-link2bee+1)) *
        termEX(4,n[4]-(vmecrate-link2cc+1)) *
        termEX(5,n[5]-(vmecrate-link2cc+1)) *
```

```
        termEX(6,n[6]) *
        termEX(7,n[7]) *
        termEX(8,n[8]);
      end do; #link2bee
    end do; #link2cc
  end do; #vmecrate

# 1 --> 6
elif (pos_f=1 and pos_b=6) then
  for link2cc from 1 to VMEcrate do
    for link2bee from 1 to BEE do
      integrand[pos_b] := integrand[pos_b] +
      Rack * FEE * Monitor *
        termF(pos_f) * termB(pos_b) * termND() *
        termIB(2,1) *
      termIB(3,BEE-link2bee+1) *
      termIB(4,VMEcrate-link2cc+1) *
      termIB(5,VMEcrate-link2cc+1) *
        termEX(1,n[1]-1) *
      termEX(2,n[2]-1) *
      termEX(3,n[3]-(BEE-link2bee+1)) *
      termEX(4,n[4]-(VMEcrate-link2cc+1)) *
      termEX(5,n[5]-(VMEcrate-link2cc+1)) *
      termEX(6,n[6]-1) *
      termEX(7,n[7]) *
      termEX(8,n[8]);
    end do; #link2bee
  end do; #link2cc

# 1 --> 7
elif (pos_f=1 and pos_b=7) then
  for link2cc from 1 to VMEcrate do
    for link2bee from 1 to BEE do
      integrand[pos_b] := integrand[pos_b] +
      Rack * FEE * Monitor *
        termF(pos_f) * termB(pos_b) * termB(pos_b + 1) * termND() *
        termIB(2,1) *
      termIB(3,BEE-link2bee+1) *
      termIB(4,VMEcrate-link2cc+1) *
      termIB(5,VMEcrate-link2cc+1) *
      termIB(6,1) *
        termEX(1,n[1]-1) *
      termEX(2,n[2]-1) *
      termEX(3,n[3]-(BEE-link2bee+1)) *
      termEX(4,n[4]-(VMEcrate-link2cc+1)) *
```

```
      termEX(5,n[5]-(VMEcrate-link2cc+1)) *
      termEX(6,n[6]-1) *
      termEX(7,n[7]-1) *
      termEX(8,n[8]-1);
    end do; #link2bee
  end do; #link2cc

# 2 --> 3
elif (pos_f=2 and pos_b=3) then
  for bee from 1 to BEE do
    for link2bee from 1 to bee do
      integrand[pos_b] := integrand[pos_b] +
      Rack * VMEcrate * FEE *
        termF(pos_f) * termB(pos_b) * termND() *
        termIB(3,bee-link2bee) *
        termEX(1,n[1]) *
      termEX(2,n[2]-1) *
      termEX(3,n[3]-(bee-link2bee+1)) *
      termEX(4,n[4]) *
      termEX(5,n[5]) *
      termEX(6,n[6]) *
      termEX(7,n[7]) *
      termEX(8,n[8]);
    end do; #link2bee
  end do; #bee

# 2 --> 4
elif (pos_f=2 and pos_b=4) then
  for cc from 1 to VMEcrate do
    for link2cc from 1 to cc do
      for link2bee from 1 to BEE do
        integrand[pos_b] := integrand[pos_b] +
        Rack * FEE *
          termF(pos_f) * termB(pos_b) * termND() *
          termIB(3,BEE-link2bee+1) *
        termIB(4,cc-link2cc) *
        termIB(5,cc-link2cc)*
          termEX(1,n[1]) *
        termEX(2,n[2]-1) *
        termEX(3,n[3]-(BEE-link2bee+1)) *
        termEX(4,n[4]-(cc-link2cc+1)) *
        termEX(5,n[5]-(cc-link2cc)) *
        termEX(6,n[6]) *
        termEX(7,n[7]) *
        termEX(8,n[8]);
```

```
      end do; #link2bee
    end do; #link2cc
  end do; #cc

# 2 --> 5
elif (pos_f=2 and pos_b=5) then
  for vmecrate from 1 to VMEcrate do
    for link2cc from 1 to vmecrate do
      for link2bee from 1 to BEE do
        integrand[pos_b] := integrand[pos_b] +
        Rack * FEE *
          termF(pos_f) * termB(pos_b) * termND() *
          termIB(3,BEE-link2bee+1) *
        termIB(4,vmecrate-link2cc+1) *
        termIB(5,vmecrate-link2cc) *
          termEX(1,n[1]) *
        termEX(2,n[2]-1) *
        termEX(3,n[3]-(BEE-link2bee+1)) *
        termEX(4,n[4]-(vmecrate-link2cc+1)) *
        termEX(5,n[5]-(vmecrate-link2cc+1)) *
        termEX(6,n[6]) *
        termEX(7,n[7]) *
        termEX(8,n[8]);
      end do; #link2bee
    end do; #link2cc
  end do; #vmecrate

# 2 --> 6
elif (pos_f=2 and pos_b=6) then
  for link2cc from 1 to VMEcrate do
    for link2bee from 1 to BEE do
      integrand[pos_b] := integrand[pos_b] +
      Rack * FEE *
        termF(pos_f) * termB(pos_b) * termND() *
        termIB(3,BEE-link2bee+1) *
      termIB(4,VMEcrate-link2cc+1) *
      termIB(5,VMEcrate-link2cc+1) *
        termEX(1,n[1]) *
      termEX(2,n[2]-1) *
      termEX(3,n[3]-(BEE-link2bee+1)) *
      termEX(4,n[4]-(VMEcrate-link2cc+1)) *
      termEX(5,n[5]-(VMEcrate-link2cc+1)) *
      termEX(6,n[6]-1) *
      termEX(7,n[7]) *
      termEX(8,n[8]);
```

```
    end do; #link2bee
  end do; #link2cc

# 2 --> 7
elif (pos_f=2 and pos_b=7) then
  for link2cc from 1 to VMEcrate do
    for link2bee from 1 to BEE do
      integrand[pos_b] := integrand[pos_b] +
      Rack * FEE *
        termF(pos_f) * termB(pos_b) * termB(pos_b+1) * termND() *
        termIB(3,BEE-link2bee+1) *
      termIB(4,VMEcrate-link2cc+1) *
      termIB(5,VMEcrate-link2cc+1) *
      termIB(6,1)*
        termEX(1,n[1]) *
      termEX(2,n[2]-1) *
      termEX(3,n[3]-(BEE-link2bee+1)) *
      termEX(4,n[4]-(VMEcrate-link2cc+1)) *
      termEX(5,n[5]-(VMEcrate-link2cc+1)) *
      termEX(6,n[6]-1) *
      termEX(7,n[7]-1) *
      termEX(8,n[8]-1);
    end do; #link2bee
  end do; #link2cc

# 3 --> 3
elif (pos_f=3 and pos_b=3) then
  for Bbee from 2 to BEE do
    for Fbee from 1 to Bbee-1 do
      integrand[pos_b] := integrand[pos_b] +
      Rack * VMEcrate *
        termF(pos_f) * termB(pos_b) * termND() *
        termIB(3,Bbee-Fbee-1) *
        termEX(1,n[1]) *
      termEX(2,n[2]) *
      termEX(3,n[3]-(Bbee-Fbee+1)) *
      termEX(4,n[4]) *
      termEX(5,n[5]) *
      termEX(6,n[6]) *
      termEX(7,n[7]) *
      termEX(8,n[8]);
    end do; #Fbee
  end do; #Bbee

# 3 --> 4
```

```
elif (pos_f=3 and pos_b=4) then
  for cc from 1 to VMEcrate do
    for link2cc from 1 to cc do
      for bee from 1 to BEE do
        integrand[pos_b] := integrand[pos_b] +
        Rack *
          termF(pos_f) * termB(pos_b) * termND() *
          termIB(3,BEE-bee) *
        termIB(4,cc-link2cc) *
        termIB(5,cc-link2cc)*
          termEX(1,n[1]) *
        termEX(2,n[2]) *
        termEX(3,n[3]-(BEE-bee+1)) *
        termEX(4,n[4]-(cc-link2cc+1)) *
        termEX(5,n[5]-(cc-link2cc)) *
        termEX(6,n[6]) *
        termEX(7,n[7]) *
        termEX(8,n[8]);
      end do; #bee
    end do; #link2cc
  end do; #cc

# 3 --> 5
elif (pos_f=3 and pos_b=5) then
  for vmecrate from 1 to VMEcrate do
    for link2cc from 1 to vmecrate do
      for bee from 1 to BEE do
        integrand[pos_b] := integrand[pos_b] +
        Rack *
          termF(pos_f) * termB(pos_b) * termND() *
          termIB(3,BEE-bee) *
        termIB(4,vmecrate-link2cc+1) *
        termIB(5,vmecrate-link2cc) *
          termEX(1,n[1]) *
        termEX(2,n[2]) *
        termEX(3,n[3]-(BEE-bee+1)) *
        termEX(4,n[4]-(vmecrate-link2cc+1)) *
        termEX(5,n[5]-(vmecrate-link2cc+1)) *
        termEX(6,n[6]) *
        termEX(7,n[7]) *
        termEX(8,n[8]);
      end do; #bee
    end do; #link2cc
  end do; #vmecrate
```

```
# 3 --> 6
elif (pos_f=3 and pos_b=6) then
  for link2cc from 1 to VMEcrate do
    for bee from 1 to BEE do
      integrand[pos_b] := integrand[pos_b] +
      Rack *
        termF(pos_f) * termB(pos_b) * termND() *
        termIB(3,BEE-bee) *
      termIB(4,VMEcrate-link2cc+1) *
      termIB(5,VMEcrate-link2cc+1) *
        termEX(1,n[1]) *
      termEX(2,n[2]) *
      termEX(3,n[3]-(BEE-bee+1)) *
      termEX(4,n[4]-(VMEcrate-link2cc+1)) *
      termEX(5,n[5]-(VMEcrate-link2cc+1)) *
      termEX(6,n[6]-1) *
      termEX(7,n[7]) *
      termEX(8,n[8]);
    end do; #bee
  end do; #link2cc

# 3 --> 7
elif (pos_f=3 and pos_b=7) then
  for link2cc from 1 to VMEcrate do
    for bee from 1 to BEE do
      integrand[pos_b] := integrand[pos_b] +
      Rack *
        termF(pos_f) * termB(pos_b) * termB(pos_b+1) * termND() *
        termIB(3,BEE-bee) *
      termIB(4,VMEcrate-link2cc+1) *
      termIB(5,VMEcrate-link2cc+1) *
      termIB(6,1)*
        termEX(1,n[1]) *
      termEX(2,n[2]) *
      termEX(3,n[3]-(BEE-bee+1)) *
      termEX(4,n[4]-(VMEcrate-link2cc+1)) *
      termEX(5,n[5]-(VMEcrate-link2cc+1)) *
      termEX(6,n[6]-1) *
      termEX(7,n[7]-1) *
      termEX(8,n[8]-1);
    end do; #bee
  end do; #link2cc

# 4 --> 4
elif (pos_f=4 and pos_b=4) then
```

```
  for Bcc from 2 to VMEcrate do
    for Fcc from 1 to Bcc-1 do
      integrand[pos_b] := integrand[pos_b] +
      Rack *
        termF(pos_f) * termB(pos_b) * termND() *
        termIB(4,Bcc-Fcc-1) *
      termIB(5,Bcc-Fcc)*
        termEX(1,n[1]) *
      termEX(2,n[2]) *
      termEX(3,n[3]) *
      termEX(4,n[4]-(Bcc-Fcc+1)) *
      termEX(5,n[5]-(Bcc-Fcc)) *
      termEX(6,n[6]) *
      termEX(7,n[7]) *
      termEX(8,n[8]);
    end do; #Fcc
  end do; #Bcc

# 4 --> 5
elif (pos_f=4 and pos_b=5) then
  for vmecrate from 1 to VMEcrate do
    for cc from 1 to vmecrate do
      integrand[pos_b] := integrand[pos_b] +
      Rack *
        termF(pos_f) * termB(pos_b) * termND() *
        termIB(4,vmecrate-cc) *
      termIB(5,vmecrate-cc) *
        termEX(1,n[1]) *
      termEX(2,n[2]) *
      termEX(3,n[3]) *
      termEX(4,n[4]-(vmecrate-cc+1)) *
      termEX(5,n[5]-(vmecrate-cc+1)) *
      termEX(6,n[6]) *
      termEX(7,n[7]) *
      termEX(8,n[8]);
    end do; #cc
  end do; #vmecrate

# 4 --> 6
elif (pos_f=4 and pos_b=6) then
  for cc from 1 to VMEcrate do
    integrand[pos_b] := integrand[pos_b] +
     Rack *
      termF(pos_f) * termB(pos_b) * termND() *
      termIB(4,VMEcrate-cc) *
```

```
     termIB(5,VMEcrate-cc+1) *
      termEX(1,n[1]) *
     termEX(2,n[2]) *
     termEX(3,n[3]) *
     termEX(4,n[4]-(VMEcrate-cc+1)) *
     termEX(5,n[5]-(VMEcrate-cc+1)) *
     termEX(6,n[6]-1) *
     termEX(7,n[7]) *
     termEX(8,n[8]);
  end do; #cc

# 4 --> 7
elif (pos_f=4 and pos_b=7) then
  for cc from 1 to VMEcrate do
    integrand[pos_b] := integrand[pos_b] +
     Rack *
      termF(pos_f) * termB(pos_b) * termB(pos_b+1) * termND() *
      termIB(4,VMEcrate-cc) *
     termIB(5,VMEcrate-cc+1) *
     termIB(6,1) *
      termEX(1,n[1]) *
     termEX(2,n[2]) *
     termEX(3,n[3]) *
     termEX(4,n[4]-(VMEcrate-cc+1)) *
     termEX(5,n[5]-(VMEcrate-cc+1)) *
     termEX(6,n[6]-1) *
     termEX(7,n[7]-1) *
     termEX(8,n[8]-1);
  end do; #cc

# 5 --> 4 (super tricky)
elif (pos_f=5 and pos_b=4) then
  for vmecrate from 1 to (VMEcrate-1) do
    for cc from (vmecrate+1) to VMEcrate do
      integrand[pos_b] := integrand[pos_b] +
      Rack *
        termF(pos_f) * termB(pos_b) * termND() *
        termIB(4,cc-vmecrate-1) *
      termIB(5,cc-vmecrate-1) *
        termEX(1,n[1]) *
      termEX(2,n[2]) *
      termEX(3,n[3]) *
      termEX(4,n[4]-(cc-vmecrate)) *
      termEX(5,n[5]-(cc-vmecrate)) *
      termEX(6,n[6]) *
```

```
      termEX(7,n[7]) *
      termEX(8,n[8]);
    end do; #cc
  end do; #vmecrate

# 5 --> 5
elif (pos_f=5 and pos_b=5) then
  for Bvmecrate from 2 to VMEcrate do
    for Fvmecrate from 1 to Bvmecrate-1 do
      integrand[pos_b] := integrand[pos_b] +
      Rack *
        termF(pos_f) * termB(pos_b) * termND() *
        termIB(4,Bvmecrate-Fvmecrate) *
      termIB(5,Bvmecrate-Fvmecrate-1) *
        termEX(1,n[1]) *
      termEX(2,n[2]) *
      termEX(3,n[3]) *
      termEX(4,n[4]-(Bvmecrate-Fvmecrate)) *
      termEX(5,n[5]-(Bvmecrate-Fvmecrate+1)) *
      termEX(6,n[6]) *
      termEX(7,n[7]) *
      termEX(8,n[8]);
    end do; #Fvmecrate
  end do; #Bvmecrate

# 5 --> 6
elif (pos_f=5 and pos_b=6) then
  for vmecrate from 1 to VMEcrate do
    integrand[pos_b] := integrand[pos_b] +
     Rack *
      termF(pos_f) * termB(pos_b) * termND() *
      termIB(4,VMEcrate-vmecrate) *
     termIB(5,VMEcrate-vmecrate) *
      termEX(1,n[1]) *
     termEX(2,n[2]) *
     termEX(3,n[3]) *
     termEX(4,n[4]-(VMEcrate-vmecrate)) *
     termEX(5,n[5]-(VMEcrate-vmecrate+1)) *
     termEX(6,n[6]-1) *
     termEX(7,n[7]) *
     termEX(8,n[8]);
  end do; #vmecrate

# 5 --> 7
elif (pos_f=5 and pos_b=7) then
```

```
  for vmecrate from 1 to VMEcrate do
    integrand[pos_b] := integrand[pos_b] +
     Rack *
      termF(pos_f) * termB(pos_b) * termB(pos_b+1) * termND() *
      termIB(4,VMEcrate-vmecrate) *
     termIB(5,VMEcrate-vmecrate) *
     termIB(6,1) *
      termEX(1,n[1]) *
     termEX(2,n[2]) *
     termEX(3,n[3]) *
     termEX(4,n[4]-(VMEcrate-vmecrate)) *
     termEX(5,n[5]-(VMEcrate-vmecrate+1)) *
     termEX(6,n[6]-1) *
     termEX(7,n[7]-1) *
     termEX(8,n[8]-1);
  end do; #vmecrate

# 6 --> 7
elif (pos_f=6 and pos_b=7) then
  integrand[pos_b] := integrand[pos_b] +
   Rack *
    termF(pos_f) * termB(pos_b) * termB(pos_b+1) * termND() *
    termEX(1,n[1]) *
   termEX(2,n[2]) *
   termEX(3,n[3]) *
   termEX(4,n[4]) *
   termEX(5,n[5]) *
   termEX(6,n[6]-1) *
   termEX(7,n[7]-1) *
   termEX(8,n[8]-1);

end if;

end do; #pos_b
end do; #pos_f

NumericStatus(false);
for i from 1 to num_cg do
  st := time();
  probcomp[i] := evalf[digits](Int(unapply(integrand[i], t),
      t0 .. tf, epsilon = 1.*10^(-precision), method=intMethod));
  rtimecomp[i] := time() - st;
end do;

numEvent := checkNumEvent();
```

```
prob := convert(Vector[row]([add(probcomp[i], i = 1 .. num_cg),
      probcomp]), listlist);
rtime := convert(Vector[row]([add(rtimecomp[i], i = 1 .. num_cg),
      rtimecomp]), listlist);
nterms := convert(nterms, listlist);

return [prob, nterms, rtime, digits, precision, intMethod,
      [numEvent, NumericEventLocation]];
end proc:
```

C.3.11 Generate_Library.mw

```
> #Build the Complete Library;
> lname := "LHC_MPS.mla";
> lib := cat(".", kernelopts(dirsep), lname);
>
> read "B1_demand_triggered.txt";
> read "B2plus_class_false.txt";
> read "B3plus_class_blind.txt";
> read "B5A_demand_end.txt";
> read "B5B_false_end.txt";
> read "B6A_demand_absorbed.txt";
> read "B6B_false_absorbed.txt";
> read "B6X_false_absorbed.txt";
> read "B7_not_demand_false.txt";
> read "TermsAndFunctions.txt";
> read "Constants.txt";
```

C.3.12 cc.txt

```
myCrossCheck1 := proc(pB1, pB2plus, pB7)
  local cc;
  cc := 1.0 - pB1[1] - pB2plus[1,1] - pB7[1];
  return cc;
end proc:

myCrossCheck2 := proc(pB1, pB5A, pB6A)
  local cc;
```

```
  cc := pB1[1] - pB5A[1] - pB6A[1,1];
  return cc;
end proc:

myCrossCheck3 := proc(pB2plus, pB5B, pB6B)
  local cc;
  cc := pB2plus[1,1] - pB5B[1,1] - pB6B[1,1];
  return cc;
end proc:

myCrossCheck4 := proc(pB5A, pB6A, pB5B, pB6B, pB7)
  local cc;
  cc := 1.0 - pB5A[1] - pB6A[1,1] - pB5B[1,1] - pB6B[1,1] - pB7[1];
  return cc;
end proc:
```

C.3.13 ParameterVariation.mw

```
> #Load Library
> lname := "LHC_MPS.mla";
> lib := cat(lname);
> march(open, lib);
               "C:\Program Files\Maple 12/lib", "LHC_MPS.mla"
>
> # Load Crosscheck Procedures
> read "cc.txt";
>
> # Global Parameters
> Precision := 6;
> Digits := 10;
>
> # Parameter Variation Set
> SetName := "-base";
>
> # Perform Calculations and Crosschecks
retvec:=myCrossChecks(Precision, _DEFAULT, SetName):
```

Appendix D

Results

The following sections contain the detailed results of the case studies (A-F, based on the implementation of the analytical description).

D.1 Case Studies: Scenario probabilities

	I	II	III	IV	V
A	0.87036978563158	0.112042244404586	0.017587942025044	0.000000025612023	0.000000002326764
B	0.684657122808324	0.299662852016367	0.015679957479801	0.00000006570579	0.000000001989715
C	0.829769210364626	0.109468648224062	0.060762113778744	0.000000024826357	0.000000002806208
D	0.652719579128008	0.293058308830633	0.054222045907798	0.000000063732267	0.000000002401292
E	0.870369786132913	0.112041587722634	0.017587859956242	0.000000682314745	0.000000083873464
F	0.684657123202687	0.299661167344522	0.015679887299189	0.000001750429803	0.000000071723797

D.2 Case Studies: Importances, false trigger

	A		B	
	III	V	III	V
Monitor	0.004194859171812	0.000000000958913	0.003739790290628	0.000000000820007
FEE	0.006991436050846	0.000000001188405	0.00623298732194	0.000000001016256
BEE	0.000034957182303	0.000000000005737	0.000031164938361	0.000000000004906
CC	0.000002689014133	0.00000000000033	0.000002397303045	0.000000000000283
VME	0.002689014149406	0.000000000173377	0.002397303058731	0.000000000148262
CIBUS	0.000089633816013	0	0.000079910111392	0
BIC1	0.001792676320264	0	0.00159820222785	0
BIC2	0.001792676320264	0	0.00159820222785	0
Total	0.01758794202504	0.0000000232676	0.01567995747980	0.0000000198971

	C		D	
	III	V	III	V
Monitor	0.004098503806842	0.000000000929498	0.00365736565534	0.000000000795378
FEE	0.006830843649181	0.000000001152051	0.006095612823677	0.000000000985818
BEE	0.003415422023147	0.00000000055617	0.003047806581744	0.000000000475918
CC	0.000002627247817	0.0000000000032	0.000002344466692	0.00000000000274
VME	0.002627247832301	0.000000000168167	0.002344466705899	0.000000000143901
CIBUS	0.008757493843524	0	0.007814889934574	0
BIC1	0.017514987687965	0	0.015629779869934	0
BIC2	0.017514987687965	0	0.015629779869934	0
Total	0.06076211377874	0.00000000280621	0.05422204590779	0.00000000240129

	E		F	
	III	V	III	V
Monitor	0.004194834585659	0.000000025545843	0.00373976926594	0.000000021845347
FEE	0.006991395073892	0.000000042166655	0.006232952280766	0.000000036058515
BEE	0.000034956977418	0.000000000210628	0.000031164763155	0.000000000180117
CC	0.000002688998373	0.000000000016091	0.000002397289567	0.0000000001376
VME	0.002688998389037	0.000000015934244	0.002397289581355	0.000000013626056
CIBUS	0.000089633290667	0	0.000079909662146	0
BIC1	0.001792676320597	0	0.001598202228128	0
BIC2	0.001792676320597	0	0.001598202228128	0
Total	0.01758785995624	0.00000008387346	0.01567988729919	0.00000007172380

D.3 Case Studies: Importances, trigger absorber

	A		B	
	IV	V	IV	V
Monitor	0	0	0	0
FEE	0.000000006567055	0.00000000024587	0.000000016847311	0.000000000210254
BEE	0.00000000459692	0.000000000460185	0.000000011793071	0.000000000393524
CC	0.000000001313405	0.000000000147316	0.000000003369448	0.000000000125976
VME	0.000000013134575	0.000000001473383	0.00000003369579	0.000000001259952
CIBUS	0.000000000000065	0.000000000000008	0.000000000000168	0.000000000000006
BIC1	0	0	0	0
BIC2	0	0	0	0
Total	0.00000002561202	0.0000000232676	0.00000006570579	0.0000000198971

	C		**D**	
	IV	V	IV	V
Monitor	0	0	0	0
FEE	0.000000006363986	0.000000000238267	0.00000001633713	0.000000000203887
BEE	0.00000000445479	0.000000000563899	0.00000001143599	0.000000000482533
CC	0.000000001272792	0.000000000182075	0.000000003267413	0.000000000155803
VME	0.000000012728421	0.000000001820978	0.000000032675388	0.000000001558223
CIBUS	0.000000000006366	0.000000000000987	0.000000000016343	0.000000000000844
BIC1	0	0	0	0
BIC2	0	0	0	0
Total	0.00000002482636	0.00000000280621	0.00000006373226	0.00000000240129

	E		**F**	
	IV	V	IV	V
Monitor	0	0	0	0
FEE	0.000000006567055	0.00000000024587	0.000000016847311	0.000000000210254
BEE	0.00000000459692	0.000000000460185	0.000000011793071	0.000000000393524
CC	0.000000001313405	0.000000000147316	0.000000003369448	0.000000000125976
VME	0.000000013134575	0.000000001473383	0.00000003369579	0.000000001259952
CIBUS	0.000000656702787	0.000000081546708	0.000001684724181	0.000000069734088
BIC1	0	0	0	0
BIC2	0	0	0	0
Total	0.00000068231474	0.00000008387346	0.00000175042980	0.00000007172379