Abstract

Plans for the future construction of the first fusion power plant are already in movement. But before constructing a demonstration power plant (DEMO) there is much work ahead. A crucial part of this work is to choose and test the material the reactor will be constructed with. For this purpose the International Fusion Irradiation Facility (IFMIF) is being developed. IFMIF recreates the same conditions a material would suffer inside of the fusion reactor. It performs this via a D-Li (Deuterium-Lithium) neutron flux with energy of 14 MeV. This flux is provided by an accelerator. One of the greatest challenges of the IFMIF project is that the test materials condition must be as similar as possible as in a real fusion reactor. For this reason IFMIF must be able to operate continuously to provide as much test data as possible.

For this reason Reliability, Availability, Maintainability and Inspectability (RAMI) analysis are required to provide the design team with enough data to reach the availability desired. The RAMI Team has created models of IFMIF in order to proper study it and improve its performance; these models have been analyzed using the professional software RiskSpectrum®. But some aspects became difficult to be modeled with RiskSpectrum® as the model grew and the complexity increased. When failure acceptance, beam degradation operation and first maintenance policies appeared, a simulation of the whole performance of the accelerator became needful.

Availsim has been created in Stanford University in order to provide availability data for the design of the International Linear Collider (ILC). This software included the features needed by the RAMI team and needed to be adapted to perform with the IFMIF model. Availsim included degraded operation and allowed to repair more than one component at a time. It also allowed including manpower as a restriction to the repairs.

But before its utilization by the RAMI team Availsim had to be adapted to accept and simulate the IFMIF model. New features have been added during the adaptation. Availsim is now able to perform multiple iterations during one simulation providing more reliable results. Originally Availsim only took into consideration the most common failure mode for a component. Thanks to the addition of functions multiple failure modes can be studied for each component. Availsim is now also able to decide whether to continue degraded operation or to stop operation to maximize the beam effectiveness.

Degraded operation simulation and maintenance strategies allow the availability simulation to be closer to reality. The beam availability result obtained with Availsim was 84,95%, superior to the 81,25% beam availability obtained with RiskSpectrum® and closer to IFMIF availability requirements of 87,00% beam availability.

ETSEIB

Fusion Energy Engineering Laboratory

Summary

1 Glossary

BA Beam Availability

BE Beam Effectiveness

CIEMAT Centro de Investigaciones Energéticas, Mediambientales y Tecnológicas

CSV Coma separated value

D Deuterium

DEMO Demonstration Power Plant

ECRIS Electron Cyclotron Resonance Ion Source

EF Error Factor

FEEL Fusion Energy Engineering Laboratory

HA Hardware Availability

HEBT High Energy Beam Transport

HFTM High Flux Test Module

IFMIF International Fusion Materials Irradiation Facility

ITER International Thermonuclear Experimental Reactor

LEBT Low Energy Beam Transport

Li Lithium

MDT Mean Down Time

MEBT Middle Energy Beam Transport

MTBF Mean Time Between Failures

MTTR Mean Time To Repair

NERG Nuclear Engineering Research Group

Fr Failure Rate

RAMI Reliability, Availability, Maintenance and Inspectability

RFQ Radio Frequency Quadrupole

SRF Superconducting Radio Frequency

TA Target Assembly

TC Test Cell

2 Preface

This Project has been developed in the Fusion Energy Engineering Laboratory (FEEL) which belongs to the Nuclear Engineering Research Group (NERG). This group work in cooperation with the Centro de Investigacioness Energéticas, Medioambientales y Tecnológicas (CIEMAT). This research group has been involved in the International Fusion Material Irradiation Facility (IFMIF) project since 2007. The FEEL task has been to create a complete set of Reliability, Availability, Maintainability and Inspectability data for each component of IFMIF. Their goal is to provide improvements and recommendations to the design team to reach the high availability requirements for IFMIF.

This project is about the adaptation of an already existing software called Availsim to IFMIF. This software had already been successfully used in the availability analysis of the International Linear Collider (ILC) and the RAMI team considered that after some adaptation it could provide new data for their studies.

3 Introduction to IFMIF

DEMO is the future demonstration fusion power reactor. Its purpose is to demonstrate the feasibility of fusion as an electrical power source. For this reason 2 parallel projects have been developed. ITER and IFMIF.

ITER (international thermonuclear experimental reaction) is a joint scientific experiment developed by China, the European Union, India, Korea, Japan and the United States. The main goal of ITER is to build a reactor able to provide more energy than consumed. It is projected that consuming 50 MW of power it will be able to produce 500 MW of output power.

The second project aim is to test materials in order to provide enough data to optimize the design of DEMO (demonstration fusion power plant). This project is IFMIF (International Fusion Material Irradiation Facility) [1].

3.1 The International Fusion Irradiation Facility

IFMIF is a scientific research facility with the purpose of testing and choosing suitable material for the construction of the future fusion reactor DEMO. This project is planned by Japan, the European Union, the United States and Russia, and managed by the International Atomic Energy Agency (IAEA).

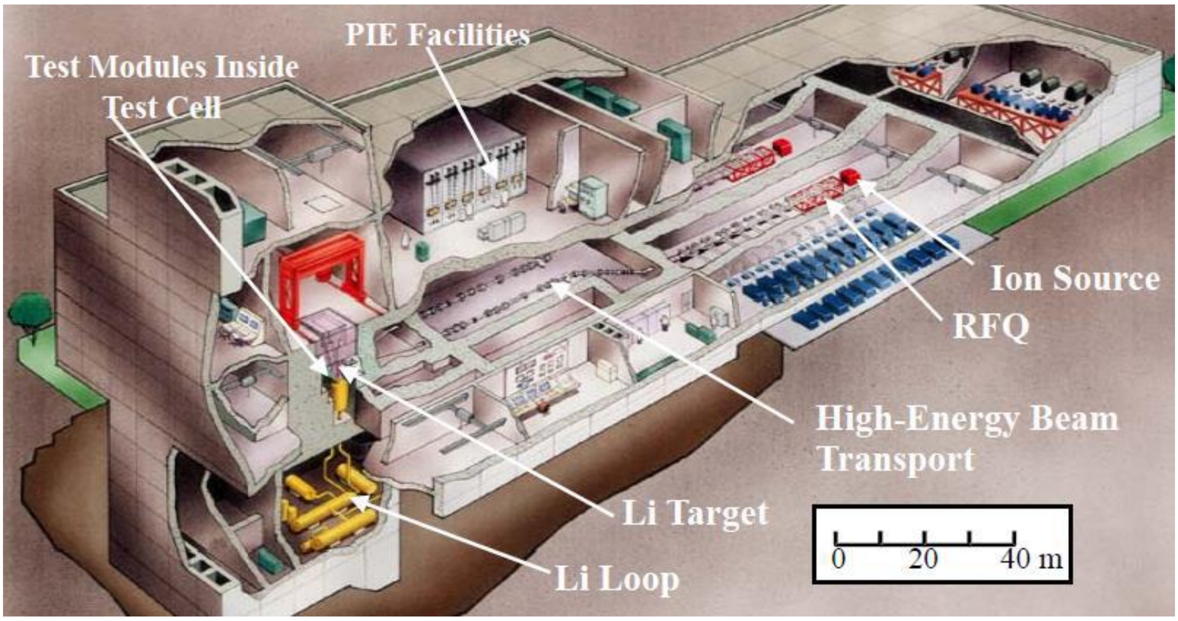


Fig. 1 Overview of the IFMIF design

3.1.1 IFMIF parameters

To test the materials IFMIF has to simulate the same conditions the material would suffer in a real fusion reactor. To do so the materials are bombarded with a neutron flux provided by a particle accelerator-based neutron source. The materials are subjected to the following conditions[2]:

- 14 MeV
- A neutron flux of 10^{17} $n \cdot m \cdot s^{-2}$
- 20 dpa/fpy (displacement per atom/full power year).

Those are the operation conditions the materials will be subjected in the fusion reactor. Since the aim of the whole project is to build fusion energy power plant, materials have to be test for extensive periods of time to recreate the operation of a real power plant.

For that reason an overall availability of 70% is required for IFMIF. This availability is different for each facility composing IFMIF. The availability requested for every facility is the following[3]:

Facility	Availability
Accelerator facility	87 %
Test facility	96%
Target facility	94 %
Conventional facilities	98 %
Central control system and common instrumentation	98 %
Total availability required	75 %

Table-1 IFMIF availability requirements

These availability requirements contain two maintenance periods. A long maintenance period scheduled for the end of one operation year with duration of 20 days. And a short maintenance period scheduled at half an operational day with duration of 3 days[4].

As stated before IFMIF is composed of different facilities each one with a different purpose. The 3 main facilities will be now briefly exposed.

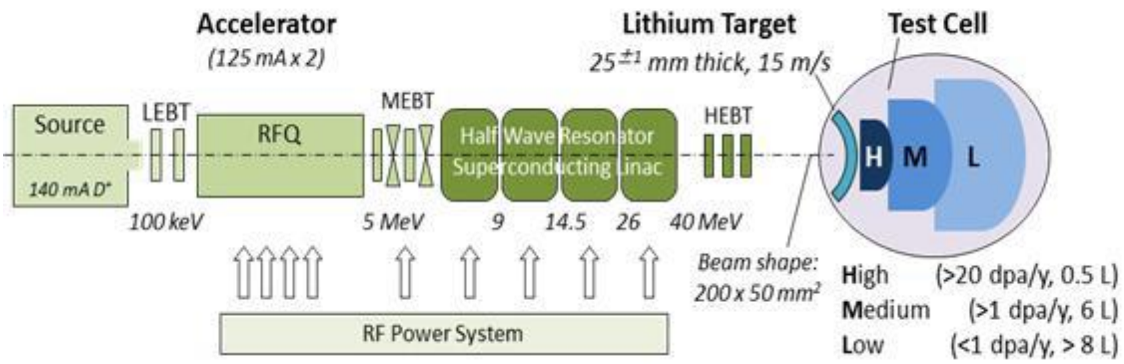


Fig. 2 Schematic principle of IFMIF project[7]

3.1.2 Accelerator facility

The accelerator facility is in fact composed by two accelerators. Each one delivers a 40 MeV, 125 mA deuteron beam to the target facility. Each IFMIF accelerator is a sequence of acceleration and beam transport stages. The deuteron beam is produced and extracted from an Electron Cyclotron Resonance Ion Source (ECRIS) at 100keV. A Low Energy Beam Transport (LEBT) section guides the deuteron beam from the source to a Radio Frequency Quadrupole (RFQ). The RFQ bunches the beam and accelerates 125mA to 5MeV. The RFQ output beam is injected through a matching section called Medium Energy Beam Transport line (MEBT), which guides the beam up to the next accelerating system: Superconducting Radio Frequency linac (SRF), composed of four cryomodules totalizing 42 superconducting cavities and 21 solenoids, bring the beam energy to 40MeV, and finally a High Energy Beam Transport line (HEBT) guides and shapes the beam to produce a rectangular and uniform footprint at the level of the lithium target[5].

3.1.2.1 **Injector**

The IFMIF ion injector, consist of the ECRIS and the LEBT section. Different kinds of ion source have been studied, but the ion source finally selected is an Electron Cyclotron Resonance (ECR) at a frequency of 2,45 GHz at 875 Gauss and will deliver a deuteron beam of 140mA at 100keV in CW.

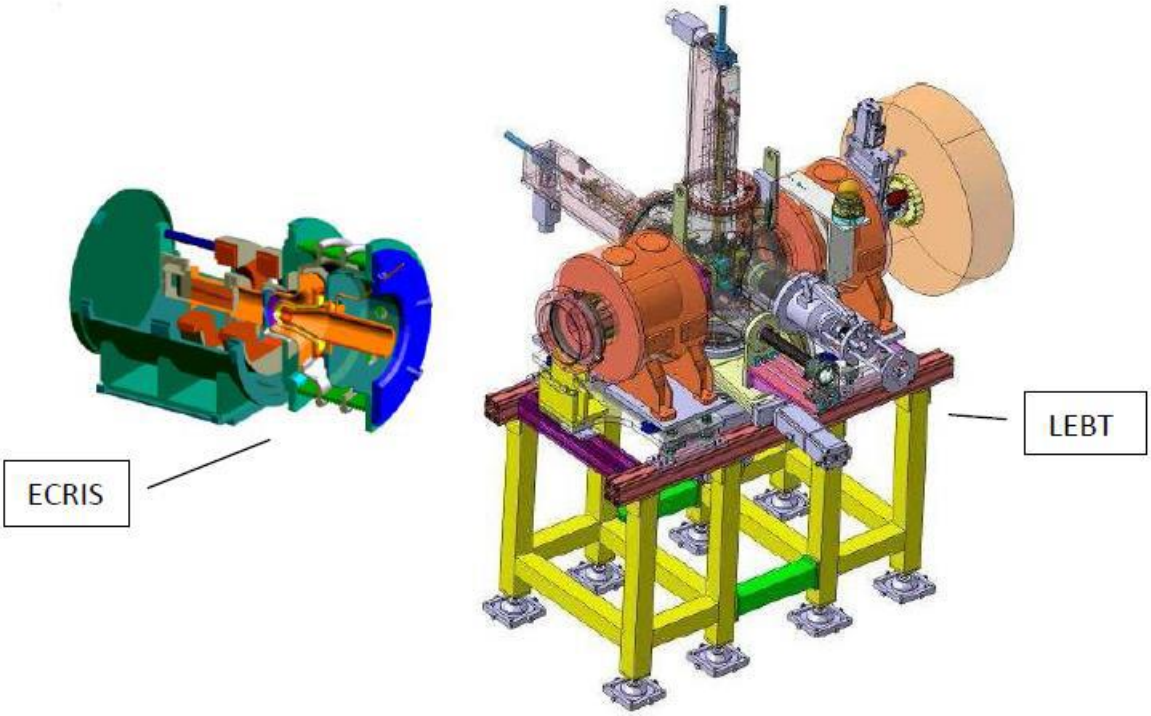


Fig. 3 Injector

The Low Energy Beam Transport (LEBT) is essentially a pair of weak focusing magnets (solenoids) which have to match the beam to the RFQ input needs. This is necessary to provide optimal acceleration and to avoid activation of the RFQ. There is also a couple of quadrupoles or steerers which are optical elements used to focus the beam in the transverse directions if it deviate[5].

3.1.2.2 **Radio Frequency Quadrupole**

The RFQ will be the largest ever built, with 18 modules (~12.5m). It will accelerate the beam from 100keV to 5MeV while strongly focuses and bunches the DC beam from the injector as required for injection into the SRF. The aim of this pre-acceleration is the optimization of the SRF Linac section, which needs an input with this energy for taking profit of the 175 MHz frequency it is fed with[5].

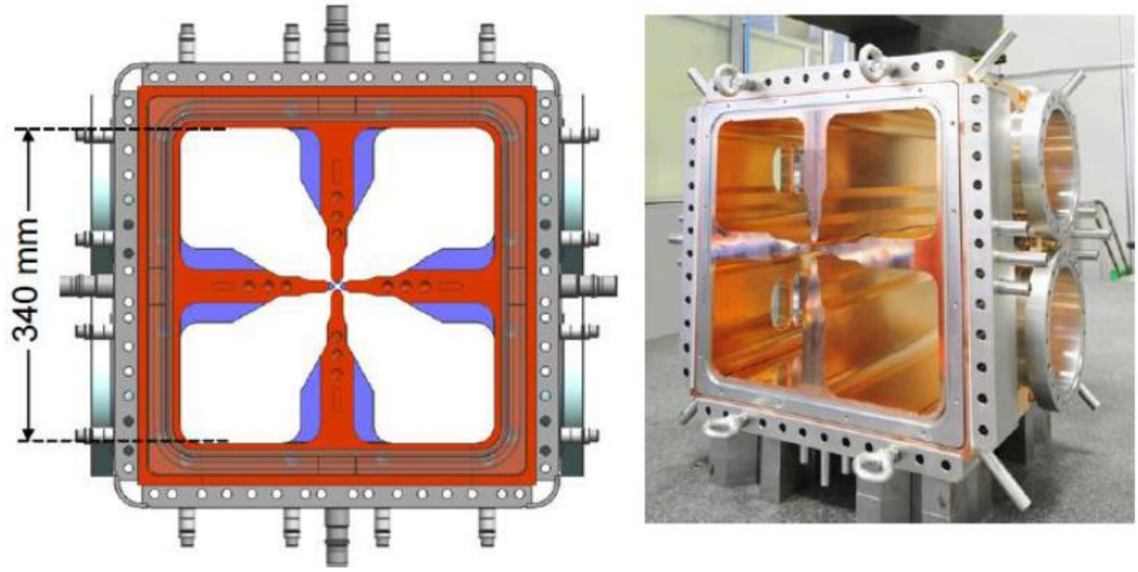


Fig. 4 Radio Frequency Quadrupole cavity

3.1.2.3 **Medium Energy Beam Transport**

MEBT focuses the beam in transverse with five quadrupoles (1triplet and 1 doublet) and in longitudinal with 2 buncher cavities. There is also a pair of collimators (scrapers) between the first and second magnet in order to absorb any deviation of the beam and properly matched into the SRF linac[5].

3.1.2.4 **Superconducting Radio Frequency Linear Accelerator**

This is the main part of the accelerator, where the beam is accelerated from 5 to 40 MeV. It is composed of 4 cryomodules. They focus the beam with solenoids and they accelerate it using superconducting Half Wave Resonators. A Half-wave resonator is a cavity made to match its measures with half the wavelength of the electric field in it. This way a resonance is generated and the amplitude is enhanced, and the energy associated to it (and transmitted to the particles) is much higher. The cryomodules have in total 21 solenoids and 42 resonators[5].

3.1.2.5 **High Energy Beam Transport**

Finally, a HEBT line focuses the beam by means of quadrupoles and homogenizes the beam density by means of higher order multipoles, bends it by means of two dipoles and expands and matches it to the required rectangular and uniform footprint at the level of the lithium target[5].

3.1.3 Target facility

Every deuteron beam with a power of up to 5 MW (40MeV, 125mA) collides on a common beam footprint with a height of 50 mm and a width of 200 mm on a free surface of liquid Li flow of 25mm thickness. This may induce a reaction in which high energy neutrons are produced, in a range peaked around 14MeV. Typical reaction are: 7Li(d,2n)7Be, 6Li(d,n)7Be, 6Li(n,T)4He[6].

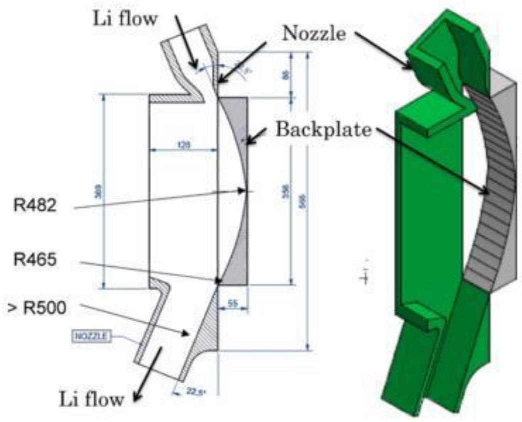


Fig. 5 Concept of the back-plate and the nozzle

To avoid boiling and significant vaporization of the liquid Li even under a high power density of up to 1 GW/m2 (10 MW in the area of 50 mm x 200 mm) and a vacuum condition for the accelerators, a concept of liquid Li target flowing at high speed (15m/s) along with a concave channel increasing a boiling point due to a centrifugal force has been employed.

These facilities have to purify, chill and monitor the Li flux constantly, to prevent any radiological hazard, and structure erosion. Moreover, Li has to be perfectly isolated from air and water, to avoid combustion. Vacuum conditions around this system are designed to prevent this from happening.

3.1.4 Test facility

The conceptual and engineering design of the IFMIF contemplates the Test Facility as three main parts: Test Cell (TC), Access Cell (AC), and Test Module Handling Cells (TMHCs). However this facility is still very susceptible to design changes.

Test cells provide the space for secure and reliable interaction of the deuteron beams, the lithium Target Assembly (TA) and the Test Modules (TMs)[5].

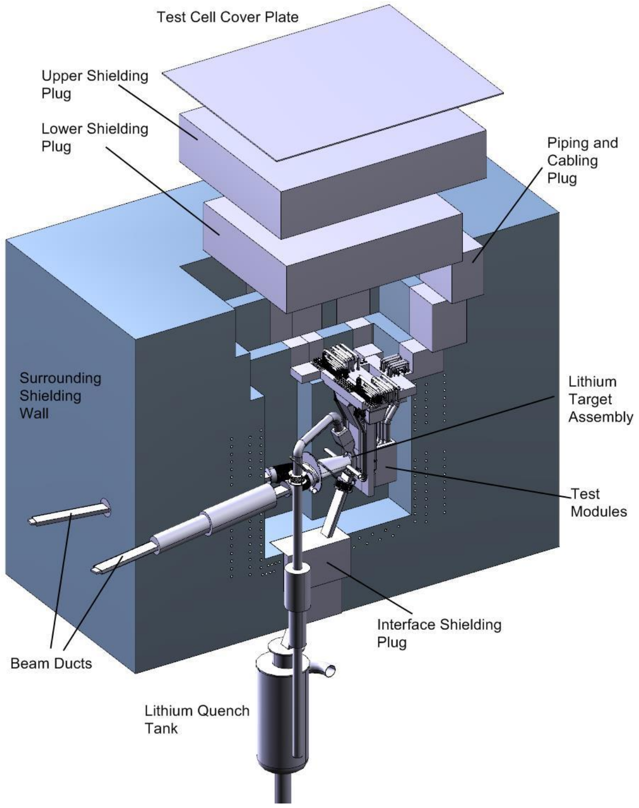


Fig. 6 IFMIF test facility

Since materials inside and around the test cells will be highly activated, remote handling is needed for manipulation and maintenance operations. The range of temperatures also implies cryogenic system necessarily.

The Access Cell provides transport capacity, space and logistics for deposition of the Test Cell cover plate and shielding plugs. It is equipped with an infrastructure for the safe transfer of Test Modules and Target Assembly to and from Test Module Handling Cells.

The Test Module Handling Cells (TMHCs) are subdivided in a chain of cells according to their functions. Decontamination, heat removal and clean the specimens to be transported to the Post Irradiation Examination Facilities (PIE).

3.2 RAMI

RAMI stands for Reliability, Availability, Maintainability and Inspectability. It describes a process whose primary purpose is to make sure that all the systems of the ITER machine will be reliable during the operation phase and maintain their performance under operational conditions with the best possible availability. Failure of only one small function might result in the machine being halted for long periods of time and result in high costs for repairs and replacements. It is therefore important that every system undergoes a technical risk analysis to evaluate what can go wrong, where and when, and to recommend spare components, back-up systems, increased frequency maintenance schedules, component standardization, systems design optimization etc, to reduce the risk level of a main function breakdown to a minimum and to decrease the time to repair to a maximum.

3.2.1 Availability

The availability of an item is expressed by the expected fraction of time it will be operational, i.e., time to perform its specified functions under given conditions at a given time t, assuming that the required external resources needed are provided. It is often expressed as (up-time)/(up-time + downtime) with many different variants. Up-time refers to a capability to perform the task and downtime refers to not being able to perform the task. The inherent availability is expressed in the following equation[8]:

$$A_i = \frac{MTBF}{MTBF + MDT}$$

(Eq. 1)

Where:

- A_i is the inherent availability.
- $MTBF$ is the mean time between failures.
- MDT is the mean down time for failure.

Inherent availability reflects the fraction of time a system would be available if no scheduled maintenance time is taken into account, which means the availability over the scheduled operation time.

3.2.2 Reliability

Reliability describes the frequency of failures over a time interval. It measures the probability for failure-free operation during a given interval. It is sometimes used for measuring the success for a failure free operation. Is expressed by the following equation[5]:

$$R(t) = e^{\left(\frac{-t}{MTBF}\right)}$$

(Eq. 2)

Where:

- $MTBF$ is the mean time between failures.
- t is the time in hours.

3.2.3 Maintainability

If a system is to have high availability, it should very rarely fail but it should also be able to be quickly repaired. In this context, the repair activity must encompass all the actions leading to system restoration, including logistics. The aptitude of a system to be repaired is therefore measured by its Maintainability.

Maintainability engineering is regarded with the implementing basic principles to future equipment repair while equipment is being designed, developed and/or fabricated. It must be a part of design planning. Maintainability characteristics must be specified and incorporated during system design and concurrent with development. The objective of Maintainability is to develop equipment and systems which can be maintained in the least time, at the least cost, with a minimum expenditure of support resources, without adversely affecting the item's performance or/and its safety characteristics[5].

3.2.4 Inspectability

The last basic tools used in RAMI engineering is the Inspectability. It is a term recently added to RAMI because that characteristic becomes essential when the component reliability cannot be improved enough. It is one of the characteristics of maintainability with a preventive objective. It is in fact defined as that characteristic of design and integration that allows in situ monitoring of equipment performance in regard to the amount of usable lifetime remaining. Furthermore, passive systems, usually safety systems, need to be inspected periodically due to their operation behavior. This includes the accessibility to equipment, removable samples to evaluate the material degradation and diagnostics to determine incipient failure. The Inspectability concerns also the monitoring aspect during the various stages of production and testing period for the inspection processes. Test engineering as a provision and access of test points, should be involved at an early stage to define test requirements and design the test approach[5].

3.2.5 RAMI goals

As has been mentioned before the required operational availability for IFMIF is 70%. The hardware availability requirements for the accelerator facility are 87% which is a great challenge.

The 87% availability requirement for the accelerator facility is related to *dpa* (displacement per atom) that both accelerators could produce in a determinate period. Taking into account this direct relation it has been assumed that [4]:

- If both accelerators are working: 100% of availability
- If one accelerator is not working is assumed a 50% of availability.
- If none are working: 0%

Meaning that:

$$\frac{Availability_{facility1} + Availability_{facility2}}{2} = Availability_{total}$$

(Eq. 3)

The requirements for IFMIF are given in terms of availability. No specific reliability requirement have been established, only the reliability requirements derived from availability ones. The mission of IFMIF is to produce a number of dpa in a period of time. In other words, to achieve a total facility operational availability of 70% in order to reach accumulated damage levels around 100 dpa in a few years of operation[4].

The previous case is valid as long as the accelerator runs at 100% operation parameters. But it has been accepted that due to failures in components the beam can be degraded. If degraded operation is accepted the beam availability must be calculated using 2 different parameters[3]:

$$HA \cdot BE = BA \qquad \text{(Eq. 4)}$$

Where:

- HA is the hardware availability of the accelerator. It only represents the proportion of time the accelerator is running not taking onto consideration if the operation is degraded.
- BE is the beam effectiveness of the beam. It expresses the proportion of dpa the beam is supplying the target in comparison with the design operation. If no component ever failed this parameter would be 100%.
- BA is the beam availability and the goal of the studies. It represent the amount of dpa that are being effectively supplied to the target.

3.2.6 Prediction

The availability prediction requires an iterative process with designers. It is where the RAMI modeling of the facility is carried out. The model represents the behavior of the failures at each functional level—facility, system, subsystem, component—as well as the equipment configuration and operating modes. The starting point for developing the models is the understanding of each process system, in order to develop a Plant Break-down Structure (PBS), followed by a Failure Mode Analysis (FMEA), which is performed to identify all significant component failure modes and the effects of failure on the operation of the system.

A model is a logical way of showing the interrelationships between the items that make up an equipment system and the attendant response as a result of failed items and other events. Many equivalent ways exist to model a given system, the most popular being event-tree analysis, fault-tree analysis, reliability block diagrams, truth (or state) tables, and Markov state diagrams. Any method that depicts relevant information in a form that is condensed, logical, and accurate is acceptable. A fault tree analysis has been chosen. The tool used to perform it is RiskSpectrum®[5].

RiskSpectrum® is a tool specifically conceived to make PRAs/PSAs, widely used in nuclear power plant industry. So it has good capabilities to cope with similar studies for a facility like IFMIF. It allows a complete organization analysis and presentation of risk and reliability information. It is a powerful analysis tool that helps to analyze complex models in a few moments and calculate availability measures by using Boolean combination of failures modes.

But due to the nature of PRAs/PSAs it was unable to take into consideration some aspects of the accelerator like the possibility to operate with degraded output. For this reason Availsim was adapted.

4 Introduction to original Availsim

The orginal Availsim is a free software tool developed by the Stanford University that has been used in the design of the ILC. The RAMI team involved in the IFMIF project thought that this tool could prove useful to reach the availability requirements demanded for the project. For this reason this project was created.

The RAMI team had already been working with the software RiskSpectrum®. Professional software widely used in this field. But RiskSpectrum® proved to have some limitations that could be included in Availsim[9].

4.1 Availsim

Availsim was created in order to provide availability calculations for the design of the International Linear Collider. The ILC is going to be one of the most complex machines ever built. Typical high energy physics accelerator availability ranges from 75% to 85%. The ILC contains an order of magnitude more parts than other accelerator, meaning that the availability would be unacceptably low unless enough effort is invested towards component reliability.

Spreadsheet calculations and commercial reliability software packages are often used to estimate the availability. But both spreadsheet and commercial software packages have some limitations. For this reason Stanford University professor Tom Himel and his team decided to write a simulation in order to include certain complexities. The simulation, named Availsim, takes an input list of components, their quantities, mean time between failures, mean time to repair and the effect of their failure. It then simulates the failure and repair of components[9].

4.1.1 Availsim vs RiskSpectrum®

Availsim and RiskSpectrum® calculate availability by different means. RiskSpectrum® performs PRAs/PSAs using fault tree analysis and obtains the probability of unavailability for a system. Availsim performs a simulation recreating the operation of the accelerator. For this reason some features can be included in Availsim that are unable to introduce in RiskSpectrum®:

- Degraded operation: Availsim has been purposely created in order to calculate the availability of systems where degraded operation is accepted. RiskSpectrum® does not contemplate this option.

– Maintenance strategy: Availsim allows to personalize maintenance shut downs to certain degree. Allowing performing scheduled repairs in previous shut down in order to save down time.

– Multiple repairs: As a consequence of the two previous features availsim allows to perform additional repairs during down times in order to repair the most number of components taking advantage of the down time.

– Operation parameters: The failed components affect operation parameters so Availsim can provide a mean and history of the parameter desired.

– Manpower: Another feature provided for availsim is that takes into consideration the manpower required and available for the repairs. Meaning that it can affect the duration of a downtime.

These features could provide additional data to the RAMI team that in conjunction with the data obtained from RiskSpectrum® would allow the RAMI studies to be more complete.

4.1.2 Features included in Availsim

Many features are introduced in the simulation to make it as realistic as possible. Each component fails at a random time with an exponential distribution determined by its MTBF. When a component fails the accelerator is degrades in some fashion. Components can be specified as hot swappable meaning that they can be replaced without further degrading the accelerator, repairable without accessing the accelerator tunnel , or repairable with access to the accelerator tunnel. Devices which are not hot swappable are only repaired when the accel is down[9].

4.1.2.1 **Repairs**

The simulation detects which parameter has been degraded too much an plans to fix things that degrade that parameter. Based on the required repairs it calculates how long the downtime must be to repair necessary items. It then schedules other items for repair allowing downtime to be extended as much as 50 to 100%.the devices chosen for repair are those who give the most bang for the buck (most improvement in the parameter per hour of repair time). Thinks that break during downtime are just ignored.

4.1.2.2 Scheduled maintenance shutdown

There are no regularly scheduled maintenance shutdowns, except an annual 3 month shutdown. In real life maintenance could be planned in case the operation parameters were getting low without inducing a shutdown. However the simulation doesn't penalize for unplanned downtimes so it does not impact the results.

4.1.2.3 Recovery

The simulation assumes that all repairs are completed on schedule. It seemed an unnecessary complication to throw random numbers to distribute the repair times around the MTTR as the simulation integrates over a long enough time period to average for such variations.

Recovery of the beam is modeled after the qualitative experience obtained from many accelerators. The longer the accelerator is down, the longer it takes to recover. The extension of the recovery time can be due to:

- Hardware failures
- Environmental factors
- Human error
- Parameter drifts
- Commissioning

Rather than modeling each of the previous recovery procedures Availsim assumes the time it takes to recover the beam after a repair is proportional to the time the beam has been down. The constants of proportionality used for the damping rings interaction regions were 20% and for rest 10% was used.

4.1.2.4 Machine development

Machine development (MD) is the time spent to the operating efficiency of the accelerator. It includes better characterization of the machine, developing new tuning procedures, and test future improvements. For the simulation it is assumed that 10% of the time MD is performed. This MD can be performed on an opportunistic basis. Some regions can finish repairs sooner than others. For this reason when one region is repaired but waiting for other region to be repaired to start tuning up, machined development can be performed in that region.

4.1.2.5 **Kludge repairs**

Kludge repairs can be simulated. This is done when proper repair would take too long to finish and a quicker work-a-round is performed in order to keep operations. The proper repair would be performed later on a down time.

4.1.3 Implementation

Availsim is written in the MATLAB® scripting language. The machine defendant input data is contained in a spreadsheet which is read by the MATLAB® program. It also contains macros in order to add the handling of the amount of data needed by the simulation. The output data obtained from the simulation is stored in another spreadsheet. It contains which component caused a downtime and how much downtime has each region of the accelerator induced[9].

4.1.4 ILC Results

Availsim results have been used to help make several ILC design decisions and establish unavailability budgets for systems and components. For example if all the damping rings were in one tunnel it would decrease the downtime by 1%, but at the same time if the rest of devices were contained in the same tunnel it would decrease uptime by 14%. Since it would be risky and too expensive to improve the reliability of individual components to regain the 14% lost the baseline ILC design has two tunnels. At the same times Availsim gives the downtime caused by each type of component. That means some components can be tuned to improve their downtime. Thank to this the RAMI team can point the Hardware R&D teams to develop higher availability versions of components in order to improve overall availability.

4.1.5 Availsim conclusions

The availability simulation has been a valuable tool in the design in the ILC. It has been used to make major design decisions and determine which components needed to have their reliability improved. Its general purpose can be extended to other accelerators[9].

4.2 Objectives of this project

The objective of this project is to adequate and modify Availsim in order to be able to provide useful and trustful data to the IFMIF RAMI team. For this reason Availsim would need to accept the data used by the RAMI team and process it. At the same time some data results that were not considered important for the ILC could prove useful for the IFMIF team so Availsim must be able to provide more results than originally was intended.

4.3 Scope of the project

The scope of the project is to adapt existing software, Availsim, to satisfy the needs of the IFMIF project. The objective of Availsim is not to simulate the operation of an accelerator but to simulate its availability. For this reason the results of the simulation are entirely subjected to the input data provided by the RAMI team.

While the project is to adapt the software to IFMIF requirements, it is desired to make Availsim as global as possible. The goal is to make Availsim a potent simulation tool for accelerators in general not a single project. For this reason it will avoided (if possible) to encode IFMIF specific operation parameters into the software.

5 Availsim 2.0 simulation

Availsim has been substantially modified in order to provide the data required by the IFMIF RAMI team. The core of the software remains basically the same only altering the parts that were incompatible with the specifications of IFMIF RAMI analysis.

5.1 Simulation elements

Availsim simulates the development of three elements which are linked between them. Events represent the physical components of the accelerator in their failure modes. Functions represent the physical parameters of operation for the accelerator and the state of their systems. Facilities represent the buildings and their state defines the availability.

5.1.1 Events

Components are the basic part of every system. All availability studies revolve around the failure of components and the effect they have over a system. For this reason Availsim needs a certain amount of information for each component in order to be able to simulate the effect I has over a system and how to deal with the consequence in a realistic way[3].

Every component can fail in more than one way. Every one of these failures may have different consequences on the system and different repair times. It is possible that one failure allows the system to keep operation while another failure mode from the same component forces a system shut down and repairs.

Usually when the repair times or the mean down times derived from the component failures are similar, only the most probable failure mode is studied. This is due to many components being replaced rather than repaired. Also in many cases every failure mode from a component prevents the operation of the component and apply the same mean down time to the system.

But if different failure modes for a component have different effect on the system there has to be a distinction between them in order to obtain reliable results and improve the performance of the system.

For that reason Availsim2.0 abandons the “component” designation for the lower parts of a system and uses “events” instead. An event is a possible failure mode for a component. Each event has its own effect on the system. This way studying which failure modes for each event have more impact on the availability in order to improve the reliability of the component is possible.

5.1.1.1 **Group of events**

Every component belongs to a bigger system which is directly affected by it. The system itself can be the physical system where the component is located or simply a grouping of components.

The aim of the system parameter is to prevent an already failed component from further affecting the accelerator. This is induced by using events instead of components. This way when and event happens the group is marked as down. This doesn't have any effect on the operation other than preventing any other event inside that group to further affect the accelerator. When a group is marked down, any event inside that group will be ignored until the original event is repaired.

If an event doesn't belong to any group its failure will never be ignored.

5.1.1.2 **Quantity**

In the original Availsim every component had a quantity associated and the components where treated as a whole. The adapted version treats every component individually. To be more precise treats every component failure mode (event) individually.

The point of this change was to increase the realism of the simulation. Originally for each component the mean time between failures was split between the number of components. For example if there were 5 component with a mtbf of 1000 hours, that would effectively give that component a mtbf of 200 hours. This procedure meant that components from the same type failed one after another.

By treating every event individually situations cane be introduced when same type events fail at similar periods. The downside of this methodology is that hugely increases the number of elements to treat increasing a lot the time needed to perform the simulation.

Nevertheless Availsim 2.0 still supports same type components treated as one.

5.1.1.3 **Time of the failure**

The time of the failure is the time in the simulation period when an event will happen. Since Availsim is a Montecarlo simulation[9] the time of the next failure will be always random centered on the mean time between failures of the corresponding event.

This is calculated generating a random number with a flat distribution between 0 and 1 and using the inverse of the integrated probability density function (p.d.f.) of an exponential to turn the flat p.d.f. into an exponential p.d.f centered on the mean time between failures.:

$$t = \frac{-MTBF}{\log(1-x)}$$

(Eq. 5)

Where:

- t is the time of the next failure
- $MTBF$ is the mean time between failures.
- x is a random number with a flat distribution between 0 and 1.

The failure rate or mtbf used by Availsim is the one the expected in steady operation. That means early failures and wear off failures are not included[5].

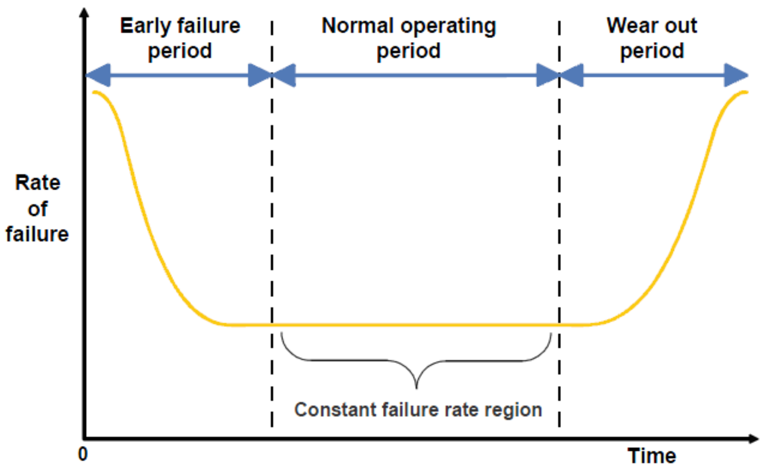


Fig.7 Bathtube curve of the failure rate

5.1.1.4 **Degradation**

When an event happens it applies a certain amount of degradation for each function the event affects. Events can affect more than one function and each one differently. The degradation can be applied as a multiplicative effect or as an additive one. The value affected is the function current value.

5.1.2 Functions

A function is a parameter that allows evaluating the state of the accelerator. Functions were introduced as a mean to allow redundancies in Availsim. In the original Availsim every component had a direct effect on a parameter. The addition of functions allows the introduction of an intermediate step that is able to check if the broken component (event) has an effect on the facility operation. This functions can be either a real operation parameter e.g. Intensity , energy … or a proxy parameter (redundancies). The utility of the functions can be explained with an example. Given a system that is supplied by three power supplies, being one of them a redundancy. In the event of a failure of the system that would imply a degradation on an operational parameter. An event occurs that brings down one power supply. The function affected is a proxy function with a design value of 3 and minimum value of 2. The event adds a -1 to the proxy function, leaving it with a value of 2. Since its value it's not below the minimum function value the function will not affect any operational parameter. However if another power supply failed it would add another -1 to the proxy function leaving it by 1, which would be below its minimum value. This time the proxy function would degrade an operational function.

There are 3 types of functions.

5.1.2.1 **Normal functions**

Normal functions affect other functions. They have a minimum and a design value and will apply degradation to the target function if their value is below its minimum.

5.1.2.2 **Critical functions**

Critical functions are the ones that ultimately define the state of the facility. If one critical function value is below its minimal it means we are below our minimum performance allowed, therefore we must shutdown the facility to make repairs. Critical functions don't affect other functions. In the case of IFMIF the critical functions were beam, energy and intensity.

5.1.2.3 **Special function**

Special functions are treated as critical functions but they can bring down the facility even when they are above their minimum value. The reason is that special functions mean value have to be maximized when possible. To do so the facility must be shut down in order to bring the function's value to an acceptable operation.

5.1.2.4 **Degradation**

Functions affect other functions the same way events do. Besides applying additive and multiplicative degradation functions can set the target function value to a specific value.

If a function is set to a value that function cannot be further degraded in multiplicative or additive way. However, the function value can be set to a new value by another function.

Functions that set other functions value have a specified hierarchy level. Once a function has been set to a value it can only be changed by a function of an inferior level.

For example, if the function Sol25 goes below its minimum value it would set the Intensity to 100 mA with a level of 3. The intensity could now only be affected by functions with a level 3,2 or 1. If the function Sol65 fails, it would set the intensity to 65 because it is a level 2 function. Once the intensity has been set to a value, it can't be degraded by addition or multiplication.

This methodology was originated while trying to calculate the degradation cryomodules from SRF linac applied to the intensity. This was a complex matter due to the way each cavity affected the beam. The degradation of each cavity was dependent on its position and the state of other cavities. To model this in Availsim a great number of functions would be required. To avoid excessive complexity the "set" method was devised as an acceptable simplification to how the intensity would react in case of cavity failures[10].

5.1.3 Facilities

A facility is the element whose state determinates the availability of the accelerator. It can be compared to a building. The simulations performed during this project only comprised 2 facilities simulating the parallel accelerators.

During the simulation facilities are independent one of another. That means an event from one facility has no means to affect in any way another facility. So in the end each simulation is effectively one simulation for each facility[3].

There is only one parameter that allows interaction between facilities. The manpower is shared by all the facilities. Therefore there could happen that all personnel were performing repairs on one facility when the other facility suffered a shut down. That would force the newly down facility to wait until enough workers are available to perform repairs, thus extending the down time.

5.2 Simulation methodology

5.2.1 Failures

Availsim treats events not components. An event is produced when a component fails in one of its failure modes. Availsim detects the failure and applies the corresponding effect on the facility taking the measures required.

As long as the accelerator is operating every failure will be dealt with. But if the facility is down, events are ignored. This is an important point, because it means that during long periods without operation many failures will be ignored. The purpose of this simplification is to avoid the complexity of dealing with failures during down times. If during a down time an event happened and required repairs in order to resume operation that would mean redirection of manpower to those critical repairs. Manpower that could be already performing non-critical reparation. At the same time, the new down time produced by the new failure would let room to more failures.

When one event is ignored a new failure time is immediately assigned to this event the same way any other event. This way although the failure is not completely ignore but postponed.

This methodology means that a potentially long down period could be ignored. The same way an event that could be repaired during the same down period it failed could provoke a new shut down in future time. The simplification was accepted but is has to be taken into consideration when studying the results.

In order to reflect the effect the failure of components have on the tuning time a randomized proportional law that adds extra tuning time in base of how long the facility has been down was adopted. This law was devised by the original creators of Availsim and it extends the down time by a mean factor of 1.2 the original down time[9].

5.2.2 Facility state

When the facility is operating the state is up. When the facility is not operating the state is down. It doesn't matter if the down time is provoked by a failure or a scheduled maintenance the state of the facility is always down. The recovery and tuning of the facility is still considered down. Only when the accelerator resumes normal operation it is considered up. Normal operation is assumed when the performance parameters (intensity, energy...) are above its minimum values.

Every time an event happens the state of the facility is checked. Once the degradation provoked has been updated, the critical functions are checked. If they are all above its minimum value then the facility is up. If one or more are below its minimum value then the state is down and repairs must be performed.

Once the repairs have finished and the facility has ended is recovery time the state is checked again in order to update the new function values.

5.2.2.1 **Shut down**

When a facility operation parameter has suffered too much degradation to continue operation the facility shuts down. The facility state goes from up to down and repairs must be performed in order to bring it up again.

5.2.2.2 **Performance shut down**

If a function is marked as a special function, there are some considerations to be taken. A special function means that whenever possible optimal performance for this function must be achieved. In IFMIF case the Intensity output is a special function.

Normally Availsim prioritizes availability over degradation. That means that as long as the value of the function is above the minimum it will not shut down to perform the repairs. But the value of the intensity is as important as the availability itself because it determines the beam effectiveness. The beam availability is the product between the hardware availability and the beam effectiveness and it is the target parameter to improve. To do this a balance between hardware availability and beam effectiveness must be reached. This is a new feature introduced in Availsim.

This is where Availsim comes in. In the event that a component fails and degrades the intensity Availsim has to take a decision. It has to check if it is more profitable to stop and perform enough repairs to bring up the intensity to its maximum value or if on the contrary is more profitable to continue degraded operation until the next scheduled long maintenance period. To do this it performs a simple operation. Availsim calculates the mean value the intensity would acquire in both situations and decide which is higher.

The mean without shutting down the accelerator to perform repair is the actual value of the intensity (I'). The mean if we decide to stop and repair is:

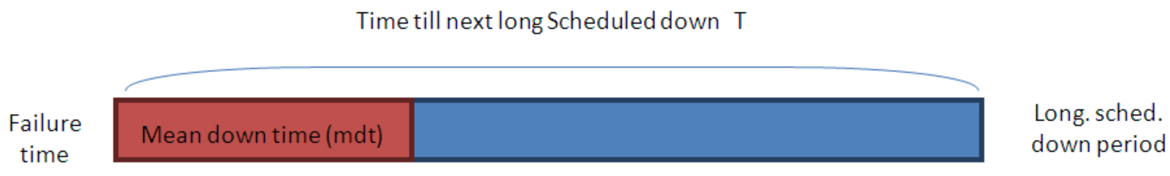


Fig.8 Distribution of time until long scheduled down time

$$\bar{I} \cdot x < \frac{(T - mdt)}{T} \cdot I_{max}$$

(Eq. 6)

This methodology does not take into consideration that after the repair to bring the intensity to its design value, the facility can suffer another failure caused by another event that brings it down. Thus increasing the down time and rendering the previous repairs useless. To take this factor into consideration a parameter x is introduced. It defines the proportion in which the average intensity has to be increased to perform these repairs. In this simulation it was used the following value:

$$x = 1{,}1$$

So this maintenance would only happen when the average intensity after repairs is 10% bigger than the average intensity obtained without performing repairs. This way a balance is reached between the hardware availability and the beam effectiveness.

If $\bar{I} > \; \mathrm{x} \cdot I'$ then I means that we must stop the accelerator in order to bring the intensity to the maximum and increase the beam effectiveness of the operation.

5.2.2.3 **Tuning**

Tuning is the state a facility gets into once all repairs needed have been scheduled. The facility will maintain this state until it finishes its recovery and resumes operation. While tuning a facility ignores all new failures.

5.2.3 Maintenance

A failure will at some point require maintenance time. This maintenance could be performed immediately after the failure, during a scheduled maintenance period or during another shut down provoked by another event.

Availsim treats differently the maintenance during a scheduled down period from a non-scheduled down one.

5.2.3.1 **Scheduled Maintenance**

A scheduled maintenance as its name say is a period of time when the facility is intentionally brought down to perform routine maintenance. The extension and start date of this maintenance is initially fixed. However there are certain factors that can affect both the duration and the date of the maintenance.

5.2.3.1.1 Duration

The extension of a scheduled maintenance is fixed by the routine maintenance that must be performed. This means a scheduled maintenance will never be shorter than the duration fixed[3].

On the other hand it can be extended. At the same time the routine maintenance is performed, failed components can be repaired. The amount of component that can be repaired depends on the component's mean down time and the duration of the maintenance period. It was decided by the RAMI team that a down time could be extended up to 150% original mean down time (MDT). This extension allows to repair components that otherwise could provoke a future longer shut down.

Recovery time after a scheduled down time is not affected by the proportional law that extends the down time depending on how long the down has lasted.

5.2.3.1.2 Date

There are 2 scheduled maintenance periods. A short one in the middle of the year and a long one at the end of it. The dates when they begin are introduced by the user. However it can substantially change during the course of a simulation.

The user can introduce a margin time that allows Availsim to advance the date of the maintenance period in order to save operation time and increase the availability. The margin used in this study was of 1 month. This means that a scheduled maintenance could be advanced up to 1 month from its original date.

There are different situations when it is decided to advance a maintenance period.

The facility is already down. If at the date of the maintenance period the facility is already down the start of the maintenance period is taken as the time when the facility went down. If the down time is shorter than the maintenance period then the down time is extended. If it is longer than the scheduled period then maintenance is performed during the down time without further effect on it.

Previous shut down. If the facility suffered a shutdown during the 1 month margin before the scheduled date, and the duration of that down time was longer than the scheduled maintenance period, the maintenance is cancelled. The necessary repairs were performed during the shutdown.

If scheduled maintenance repairs are performed during a down time provoked by an event, the time used is subtracted from the amount of down hours provoked by that event and counted as normal scheduled maintenance time. This means an event that causes a long shut down can have its down hours substantially reduced because most of those hours count as scheduled maintenance time.

5.2.3.1.3 Repairs

Repairs performed during a scheduled maintenance have fewer restrictions than those performed in a shutdown. There is no manpower limitation. It is a scheduled date and enough personnel is supposed to be available to perform all the required repairs. Access to all the facility is allowed including the vault[3].

However if the maintenance repairs were performed during a previous shutdown, the repairs are submitted to restrictions of a non-scheduled shut down.

5.2.3.2 **Non scheduled maintenance**

A non-scheduled maintenance is provoked when an event brings down the facility and requires to be repaired in order to resume operation.

5.2.3.2.1 Repairs

The procedure to choose which events are to be repaired during a nonscheduled shut down differ depending on the event that caused it. If the event that caused the shutdown brings down the facility by itself it must be repaired. An event that brings down the accelerator by itself is an event that will always bring the accelerator down, no matter the degradation it already has. For that reason these kinds of events must always be the ones repaired first and all the manpower required will be destined to its repair.

The second case is when the facility goes down due to accumulated degradation. It means that too many components have failed and the degraded operation is no longer sustainable. These situations are handed differently. Availsim sorts all the broken events from the one with the longer mean down time to the one with the shortest one. Then checks for every event the degradation it provokes and saves it. It continues to check events until one brings the parameter below the allowed degradation. That event is sent for repair and continues checking the rest of events. Using this methodology, the events with longer mean down times are checked first and they are less prone to be repaired. As it moves through the events accumulating degradation the mean down times of the events keeps getting shorter. This way the events sent to be repaired are the ones with lowest mean down time thus minimizing the down time of the non-scheduled maintenance.

Once enough repairs to bring up the facility are scheduled secondary repairs can be performed. Secondary repairs are those events that don't force the facility down but get repaired because there is enough time and manpower to do it. The events to be repaired are chosen the same way as the accumulated degradation case. It is desirable to repair the events with longer mean down times instead of the ones with shorter ones. The reason is that events with shorter mean down times can be repaired on future shorter down periods. A secondary repair can extend the maintenance time to a certain factor specified. In the case of IFMIF it was decided that the maintenance time could be extended up to 150% the original down time.

Secondary repairs are subjected to 2 more restrictions. One is the manpower available. Once the manpower has been assigned to the primary repairs the rest of personnel can perform secondary repairs. If secondary repairs are finished within the down time and there is enough time to perform more repairs the recently free manpower will be reassigned to new repairs.

The other restriction is the vault access. It takes a fixed period of time for the vault to be accessible by the repair personnel. If the primary repairs don't require access to the vault, and the duration of the down time is superior to the vault access time, access will be granted to the maintenance team to perform repairs in the vault. In case the primary repairs require access to the vault, access will be granted automatically to perform secondary repairs in it.

5.2.3.2.2 Duration

The duration of a nonscheduled down time is set by the crucial repairs needed to bring up again the facility. The mean down time provoked by an event that needs to be repaired is calculated by the following expression[3].

$$mdt = access\ time + mttr + recovery\ time$$

(Eq. 7)

Access time includes the cooling time of the system (if necessary), the physical time needed to access the location of the repairs and the time spent on the detection of the failure.

The mean time to repair (MTTR) as its name says is the mean time required to repair the failure by the number of worker specified. Availsim makes no distinction between the component being repaired or replaced.

The recovery time includes all the actions performed after doing the repairs. There is a recovery time for the system followed by a tuning up time. The duration of a nonscheduled maintenance can be extended up to a certain specified factor the same way a scheduled maintenance could. In this case it was allowed to extend the downtime up to the 150% of the initial down time.

5.2.3.2.3 Recovery

Originally Availsim calculated the recovery time in a different way. The ILC included dumpers that allowed continuing operation for the upstream regions of the accelerator while the ones downstream were shut down. For this reason the recovery was dependent on the regions that were down. Every region had a recovery time of its own, Availsim calculated the recovery time as a chain of recovery time by region. This is of course correct but the IFMIF RAMI team had already calculated the accumulated recovery time for each event. For this reason this calculation was not needed because the value was an input data so this part of the code was scratched. Now Availsim instead of calculation the recovery time by bringing up region after region and accumulating the recovery time of each one just takes the value of the input data.

At the same time, the original Availsim developers had included a factor that increased the recovery time the longer the shutdown was. This factor was obtained through empiric observation. This factor was maintained because it represents unforeseen difficulties in the repairs, and mitigates the effect that during repairs failures are ignored. The value used in Availsim 2.0 was directly extracted from the one used in the original Availsim and is 0.2. However is not used as an absolute value but randomized with a normal distribution in order to represent more realistically the variable recovery time[9].

5.2.3.2.4 Hot repairs

Hot repairs are provoked by those events that can be repaired while the facility continues operating. These repairs are treated separately. They don't share manpower with the rest of repairs. The duration of the hot repairs is calculated the same way as a normal repair.

6 Availsim 2.0 software description

6.1 Availsim 2.0

Originally Availsim was run from an excel file. The excel files contained all the elements of the ILC accelerator. Using a macro allowed to switch different configurations of the ILC in order to simulate them and compare them. After selecting a configuration using another macro would transform the data to CSV format and start the simulation. Unfortunately this excel file was very specific for the ILC accelerator. So the macros were dismissed. Now the input files are saved as CSV files and the availsim routine is initiated from Matlab®. After performing the simulation the results are automatically stored in an XLS file named by the user.

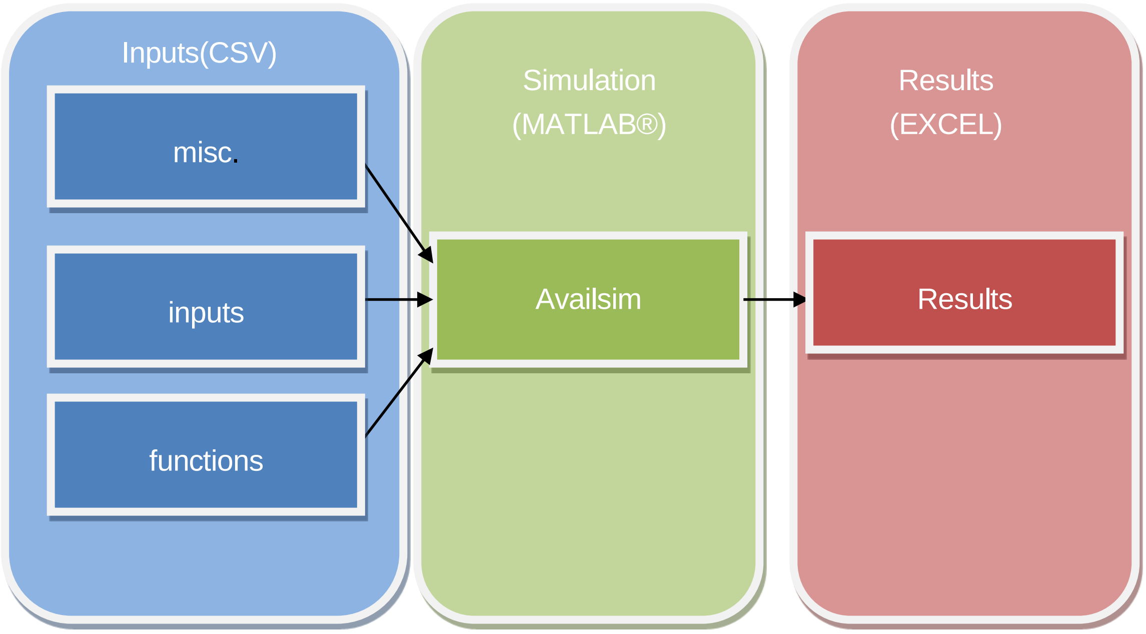


Fig.9 Availsim operation stages

6.2 Data input

The input data sheet was substantially modified to the needs of the RAMI team. The input data sheet is introduced in Availsim in .csv format.

6.2.1 Events

Instead of using components, the sheet now stores events. An event is a specific failure mode for a component; this means that a component will have as many events as failure modes. This allows Availsim to study different failure modes for a component instead of only the most probable one.

6.2.1.1 **Name**

The first main information the sheet provides is the name of the event or the gate. This holds no purpose other than to be easily identified by the user. For an event it usually contains the component involved and its failure

6.2.1.2 **Quantity**

Availsim 2.0 treats events individually. However same type events can still be treated as a group. By default quantity foe each event will be 1 unless is specified by the user otherwise.

6.2.1.3 **Facility**

Facilities are described by a number. For example if the system to be simulated has 2 accelerators one would be Facility 1 and the other Facility 2.

6.2.1.4 **Location**

Initially this value was intended to point Availsim an “Access Time” value from another data sheet, but it was more efficient to introduce all the time values on one single sheet. Now this value is used to count how many times the Vault has been accessed. The Vault location is coded as “V”. Any other location serves just informative valor.

6.2.1.5 **The ID (identification)**

As mentioned before, the RAMI team has been working with the RiskSpectrum® software. An identification code was created in order to classify every single event. This code contains letters and numbers and defines the facility, the part, the location, the recovery time, the component and the number of every event in this order. The code was maintained in Availsim as a way to compare easily results with RiskSpectrum®[5].

6.2.1.6 **Mean time between failures**

Availsim uses mean time between failures which is the inverse of the FR while Risk Spectrum uses the FR itself.

6.2.1.7 **Access time**

This is the time needed in order to start the repairs.

6.2.1.8 **Mean time to repair**

The mean time to repair is the time needed to repair an event.

6.2.1.9 **Recovery**

This is the time that takes for a repaired event to achieve normal performance again after a down time.

6.2.1.10 **Manpower**

The Manpower is the amount of workers needed to repair an event.

6.2.1.11 **Function affected**

This is the function affected by the event. If this field is empty the event won't have any kind of effect on the accelerator performance.

6.2.1.12 **Degradation calculation**

This field describes in which way the degradation will be applied on the target function. It can be Multiplicative or additive (negative. This is indicated by entering Mult or Add in this field.

6.2.1.13 **Degradation**

The degradation is the effect taken by the target function. Its value and the measurement unit depend exclusively of function affected.

6.2.1.14 **Group of comp**

This parameter prevents events that events that apply the same degradadion further degrade the operation. If this field is empty it means that the event's degradation will be applied in all cases.

6.2.2 Functions

The functions sheet replaces the parameter sheet in the original Availsim. Originally the parameters in Availsim represented real operation parameters. Now the functions can represent redundancies, binary values and operation values.

6.2.2.1 **Facility**

The facility attribute defines which facility does a function belong to and it is affected by.

6.2.2.2 **Type**

The normal functions are coded with a 1, the special ones are coded with a 2 and the critical ones are coded with a 3.

6.2.2.3 **Name of the function**

The name of the function has to be the same one used in the events sheet on the "function affected" field because is the one used by Availsim to link Functions and Events.

6.2.2.4 **Design value**

The design value of a function is the one a function has when is no degraded. Its the starting point for every function and the one that is going to be degraded in case of a failure.

6.2.2.5 **Minimum value**

The minimum value is the last value the function can reach until it degrades another function. While the value is between the design value and the minimum value the function is considered degraded. For critical functions being below the minimum value forces the facility to shut down and make repairs.

6.2.2.6 **Function affected**

This field contains the name of the function affected by the current function. The name has to be the same as the targeted function because is the one Availsim uses to link functions.

6.2.2.7 **Degradation calculation**

This field describes in which way the degradation will be applied on the target function. It can be Multiplicative, additive (negative) or set to a fixed value. This is indicated by entering Mult, Add or Set in this field.

6.2.2.8 **Degradation**

The amount of degradation applied on the target function. If the degradation calculation is Set then this field must contain the value the target function will be set to.

6.2.2.9 **Level**

The parameter level is only used if the current function sets another function to a certain value instead of applying additive or multiplicative degradation. If the target function has already been set to a value by another function, the current function will only be able to set the target to a new value if its level is more important or equal than the last one. The level is sorted in an descending order, meaning that functions altered by another function can only be set to a new value by a function of the same or lower level.

6.2.3 Miscellanea

The miscellanea sheet contains parameters of the simulation as well as the names of the input files.

6.2.3.1 **Events input**

This field contains the name of the events data sheet. The name has to include the extension (.csv) in order for Availssim to find it.

6.2.3.2 **Functions input**

This field contains the name of the functions input file.

6.2.3.3 **Max people in repairs**

This is the maximum people available for performing repairs at the same time. Note that this value must be at least equal to the highest manpower required to repair an event.

6.2.3.4 **Simulation hours**

The total duration of the simulation

6.2.3.5 **Allow access**

The amount of hours a shutdown has to last in order to allow access to the vault to perform secondary repairs.

6.2.3.6 Extra repair factor

The proportion a shutdown can be extended in order to repair secondary failures.

6.2.3.7 First short scheduled down/duration/ frequency

The hours until the first scheduled maintenance period happen, the duration of it and the hours between these short schedules.

6.2.3.8 First long scheduled down/duration/ frequency

The same data as 6.2.3.7 but with the long scheduled maintenance.

6.2.3.9 Schedule down cancelation

The margin of hours before a scheduled maintenance period in which if a shutdown happens (and is long enough) can replace the scheduled maintenance.

6.2.3.10 Trace

If this value is 1 the history matrix will be created. If no history is needed the value must be 0

6.2.3.11 Seed

The seed that will be used in order to generate the random values.

6.3 Routines description

6.3.1 AVAILSIM

Routines called: INITMISC, INITFUNCTIONS, INITEVENT, INITFACILITY, MAINLOOP, SAVERESULTS and PRINTRESULTS.

Input:

This is the main program that simulates the availability of the acceleration. It can be structured in 3 parts. The first one is the inicialization of the main variables (events, function, facility and miscelanean variables). Once all the variables are filled the program is ready to begin the simulation. The second block is the simulation itself, it provides the loop in order to perform as iterations as established. Inside the loop one simulation is performed and its results stored. Once the simulation is done the main variables return to their original values and the random seed is altered in order to provide a brand new iteration. The third block saves the results of all the iterations combines into an .xls file.

This routine was barely altered from original Availsim, the main change that was introduced was the loop to perform multiple iterations.

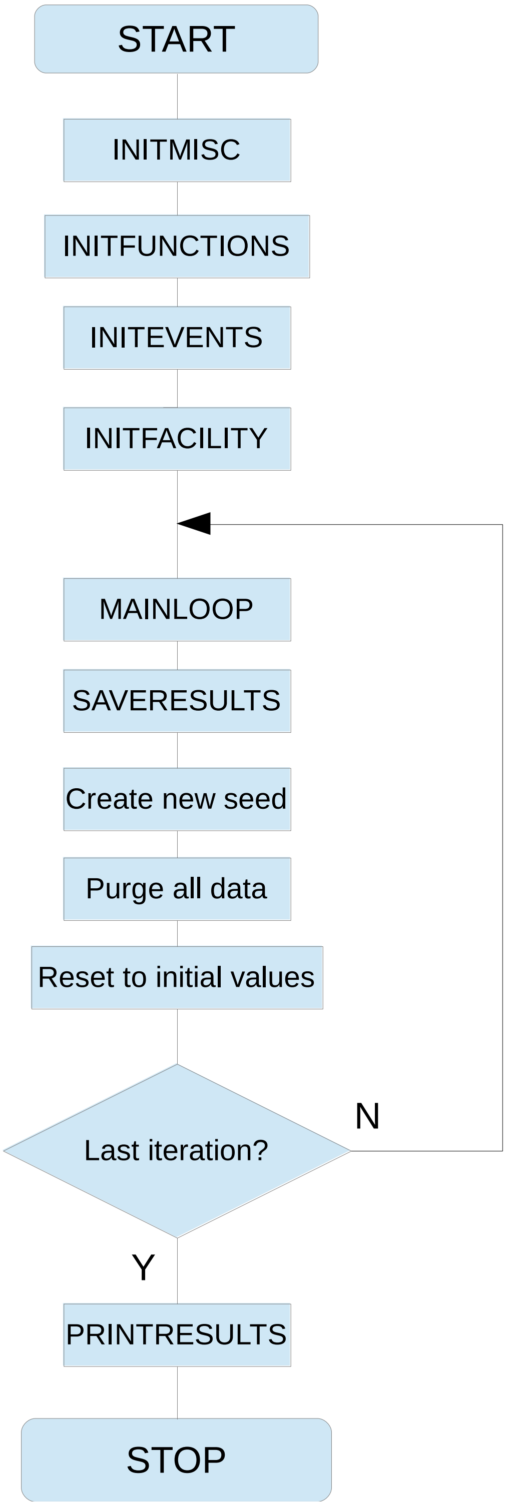


Fig.10 Availsim routine flow diagram

6.3.2 Initiation routines

6.3.2.1 **INITMISC**

The miscellanea initiation routine reads the misc.csv file which contains the fixed parameter for the simulation as well as the name of the files from wich Availsim must extract the input data and where to save the results to. It creates the facilityresults, eventresults and functionresults structures in order to be filled with all iteration results. The history matrix is also created by this routines. This matrix is a log of every event or action in the simulation sorted by time. Due to the number of events and the length of the simulated period this matrix ends up having several thousand lines which drains too many resources and thus is not recommended for more than 1 iteration. Its main use is to facilitate bug hunting and serve as an initial view of how the simulation operates.

The rest of parameters are explained in the input data subchapter.

6.3.2.2 **INITFUNCTIONS**

Routine called: INITFACILITY,

This routine read the function data from the file specified in the miscellanea sheet and creates the functions structure filling it with the data used in the simulation.

The routine also creates a matrix named critfunctions that contains the pointers to the critical function to easier accessibility during the simulation. It also copies the structure function into initialfunctions that will be used at the end of each iteration to initialize the function structure to start a brand new iteration.

6.3.2.3 **INITEVENT**

The INITEVENT routine initializes the event structure and fills it with the events input file specified in the miscellanea file. The event identification code defines the facility, part, location, recovery, redundancy and type of component of the event. Also a number at the end is based in order to separate it from equal events. It is different for every event.

Another important task performed by the INITEVENT routine is the creation of the IBANG matrix. Originally in Availsim every component had a direct degradation on the beam. Dividing the degradation between the mean time to repair the component , one could obtain the cost of every repair hour of the component. Sorting these costs from higher to lower one could obtain which components offered the most “bang for the buck”. In other word which component would be more rentable to repair first. However, due to the introduction of functions, this methodology was not possible anymore. So in this version the matrix IBANG sorts the events by mttr + recovery. Although it cannot be assured the repairs will be the most rentable, the downtime required for the repairs is minimized.

6.3.2.4 **INITFACILITY**

This routine is called from INITFUNCTIONS in order to create the facility structure.

6.3.3 Simulation routines

6.3.3.1 **MAINLOOP**

Called by : AVAILSIM

Routines called: RANDEXP, FACILITYSTATE, SAVEYEAR, SETUPSCHEDDOWN, RECOVERFROMSHUTDDOWN, SCHEDULEHOTREPAIRS, SETUPREPAIRS, RECOVERFROMREPAIR.

MAINLOOP is without any doubt the most important routine on Availsim. It is the core of the simulation and calls many important routines. The first part of the routine initializes the local variables that will be used later. The most important of these variables is the NEXTBREAK matrix. This matrix stores the next failure time of each event, which is calculated by the RANDEXP routine and gets refreshed everytime the event is repaired (or ignored). Once we have at our disposal all the variables needed MAINLOOP proceeds to select the next time event that is going to happen. These time events can be:

- MINFIXTIME: A hot fixable component has been repaired.
- NEXTSHORTSCHEDDOWN: Is the time for short scheduled maintenance period.
- NEXTLONGSCHEDDOWN: Is the time for long scheduled maintenance period.
- MINBREAKTIME: An event involving the failure of a component in one of its failure modes has happened.
- ENDTUNETIME: The recovery and tune up after a shutdown (scheduled or non-scheduled) has finished.
- ENDYEAR: Reached the end of a year.
- ENDSIMULATION: Reached or surpassed the simulation hours.

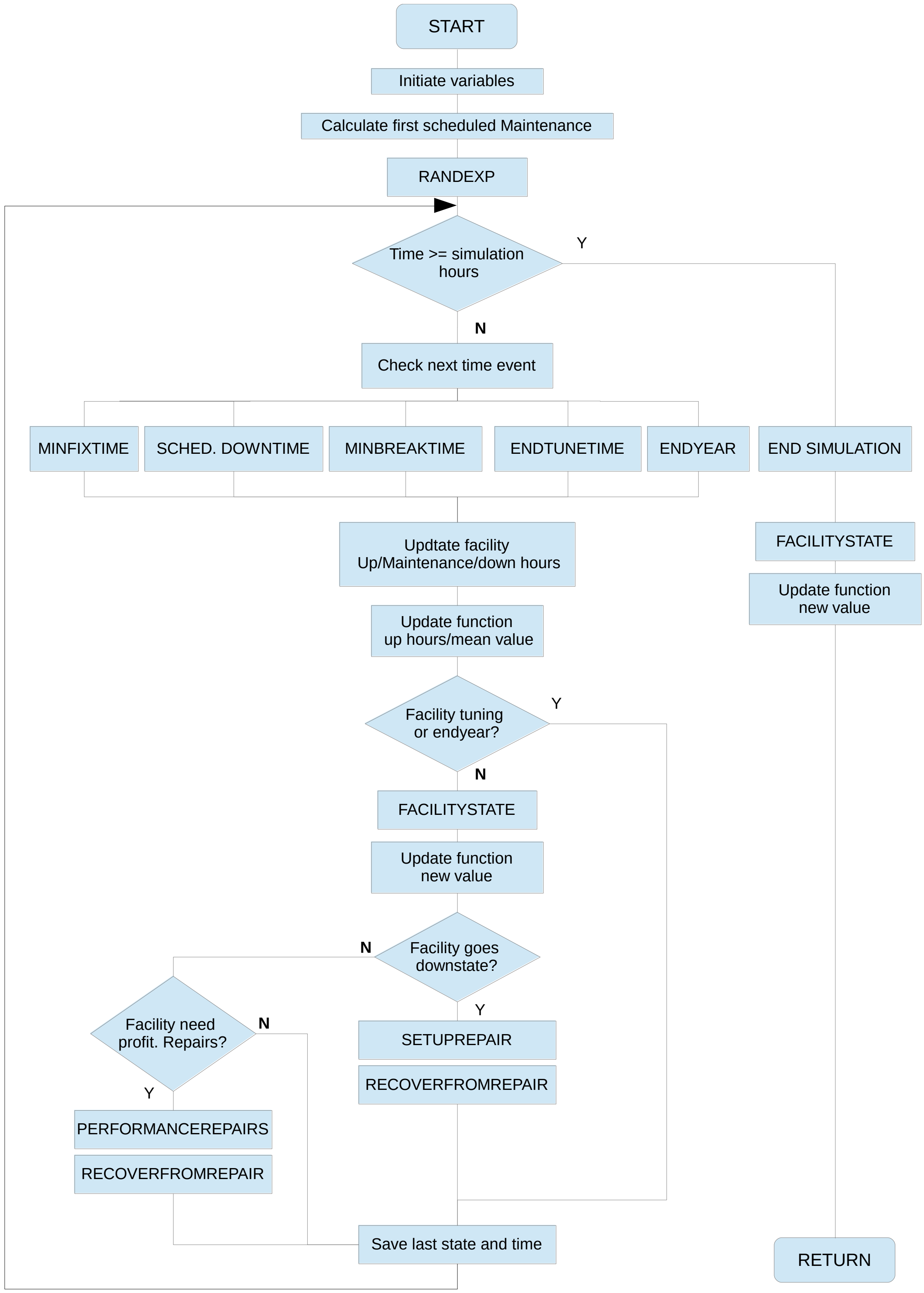


Fig.11 MAINLOOP routine flow diagram

6.3.3.1.1 MINFIXTIME

A pretty straight forward event, a hot repairable event has been repaired. A new failure time is given and it is taken out from the list of hot repairable events under repair.

6.3.3.1.2 NEXTSHORTSCHEDDOWN & NEXTLONGSCHEDDOWN

Once the time for our scheduled maintenance period is reached it is required to check if the facility is already down or if it is up. The procedure is the same for both the short a long maintenance periods and it is explained in the chapter [5.2.3]

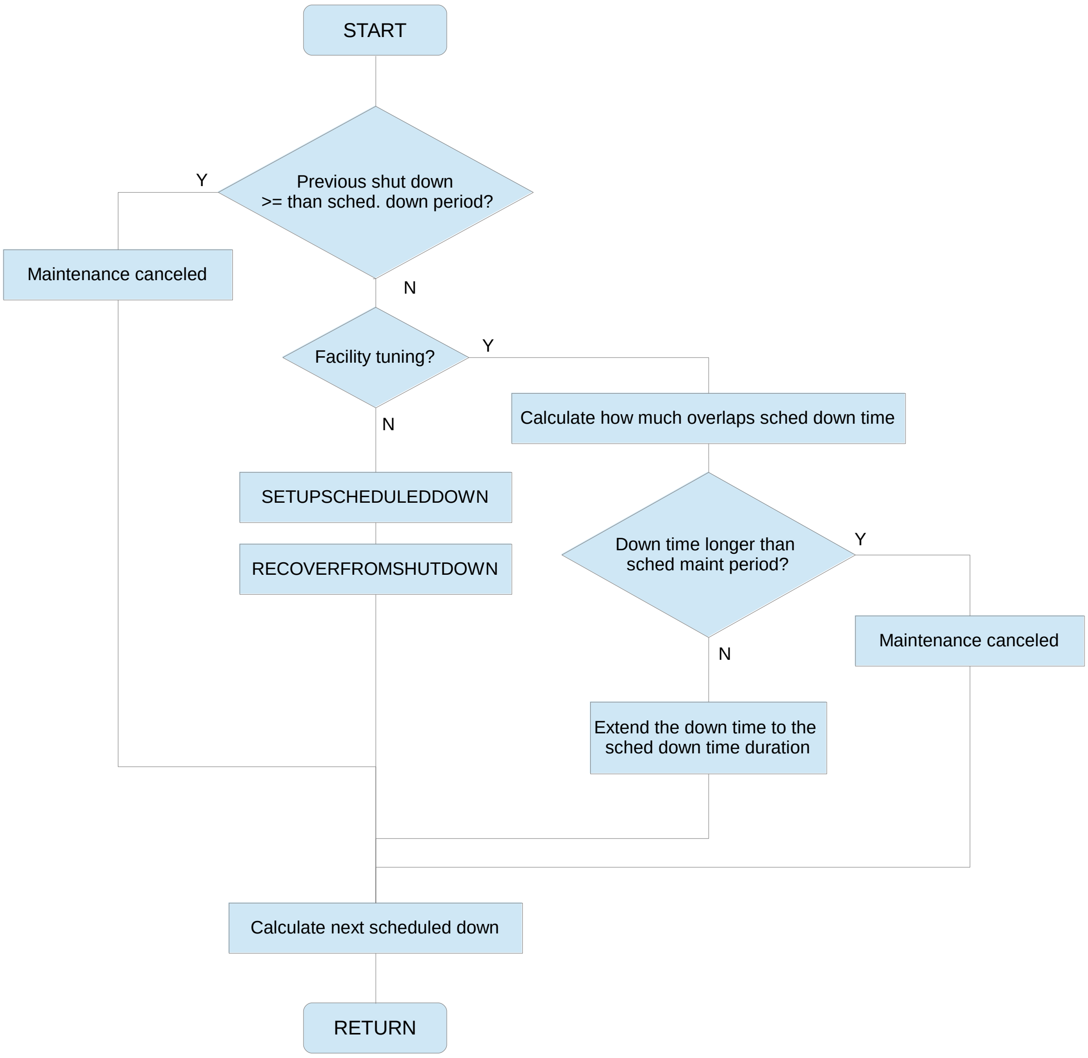


Fig.12 NEXTSHORTSCHEDDOWN & NEXTLONGSCHEDDOWN flow diagram

If the maintenance is not canceled the routine s SETUPSCHEDULEDDOWN and RECOVERFROMSCHEDDOWN are called.

6.3.3.1.3 MINBREAKTIME

When a component breaks into one of its possible failure modes we call it an event. When an event happens we have to take some considerations before checking what effect it has on the facility operation.

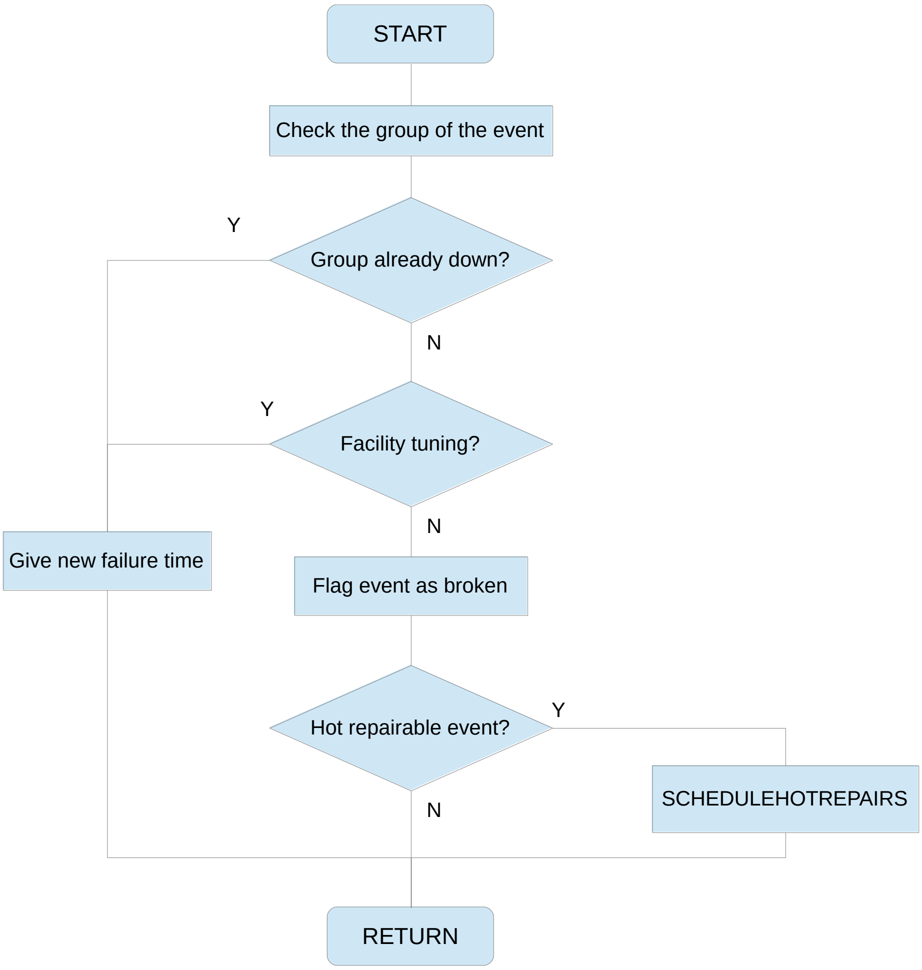


Fig.13 MINBREAKTIME flow diagram

If the facility is tuning, it means it is down and it already has an end tune time assigned. If a component fails when we are down, we ignore this failure. This is done because it would require a new end tune time and reorganizing the manpower, that could mean leaving repairs half done. This would exponentially increase the complexity of the process. In order to reflect the effect the failure of components have on the tuning time we adopted the randomized proportional law that adds extra tuning time in base of how long the facility has been down.

There are group of components that are part of the same system. When a component from that group fails, it brings the full system down and applies its corresponding degradation to the facility. If another component from the same group fails, it would have no effect because the system is already down. To represent this we have the group attribute. If a component from an already failed group breaks, this failure is ignored.

We must assign new failure times to the ignored components and store the times every component has been ignored. If a component with a high mean down time is ignored several times it would take veracity from the simulation and we need to know it.

If we are neither tuning, nor the event's group is down, we have to update the component as broken. If the component is hot repairable we schedule its repair straight away calling the routine SCHEDULEHOTREPAIRS.

6.3.3.1.4 ENDTUNETIME

This event means that repairs, recovery and tune up of the facility have finalized and it is ready to resume operation. The facility's state is set to up and the repaired events are given new failure times.

6.3.3.1.5 ENDYEAR

This event has no effect in the simulation and serves only the purpose to obtain independent annual results in order to observe how the availability progresses during the simulation time.

6.3.3.1.6 Checking the state and scheduling repairs

Before checking the consequences the current event has on the facility operation one must gather the data of the facility and functions performance from the last event to the current time. If the last state was down the corresponding down hours to the facility must be added, if it was up then up hours will be added, etc

The time has come to check the effect the current time event has on the facility's performance. This is done by calling the routine FACILITYSTATE which determines the new function's values, and ultimately, the facility state. If the facility state is up, it means that the facility operation is somehow degraded but we are able to continue operation. If its state is down we are no longer able to continue operation and the facility must be shut down and perform repairs. If the facility is tuning there is no need to check its state or to schedule further repairs.

To decide which component should be repaired the routine SETUPREPAIRS is called. Once repairs have been scheduled the time the facility will be able to resume operation is calculated by the RECOVERFROMREPAIR routine.

The loop is now completed and the program proceeds to choose the next time event.

6.3.3.2 **SETUPREPAIRS**

Called by : MAINLOOP

Routines called: SCHEDULEREPAIRS

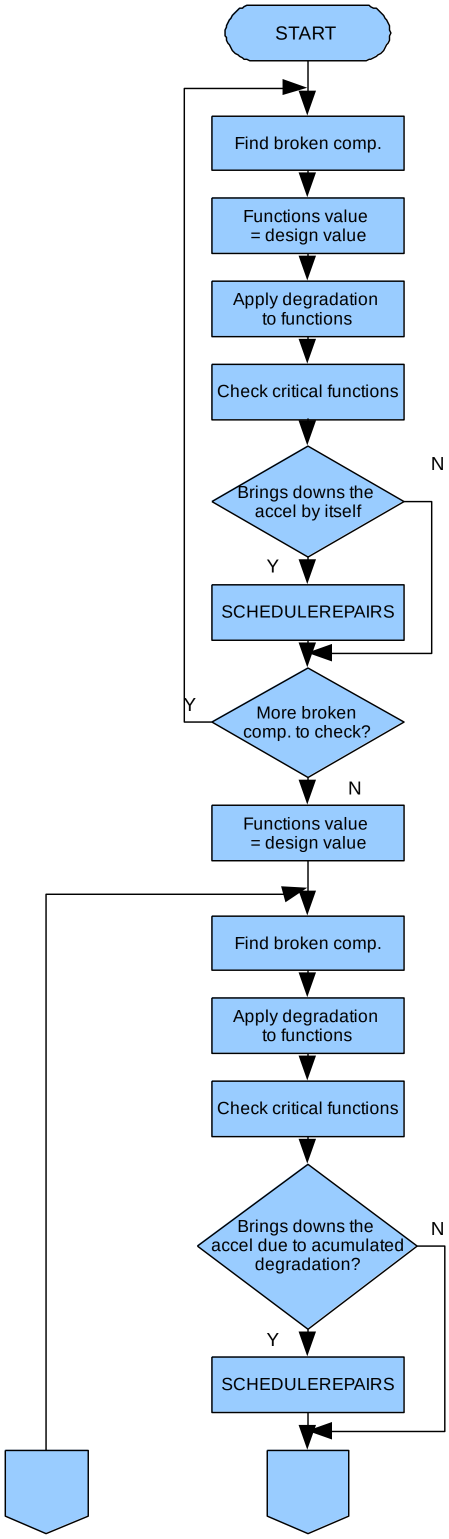
If one facility must be shut down to perform repairs, this routine decides which events are to be repaired. It starts by looping through the broken components to see if one by himself brings down the facility. If it finds one then SCHEDULEREPAIRS is called.

If no event is found able to shut down the facility by itself SETUPREPAIRS starts looking for broken component that bring down the facility due to accumulated degradation. For this procedure the IBANG matrix was created. As explained before the IBANG matrix sorts the events by mean down time. Thanks to his matrix the broken components are sorted from the highest mean down time to the lowest. The program starts looping from the events with highest mean down time and adding the corresponding degradation. At some point an event will make the facility go from up state to down state, so this event is where we will start the repairs. The utility of this method is that allows the program to discard the broken components with high mean down time first, thus reducing the amount of time required to resume operation. When the program reaches the event that makes the facility go down it calls SCHEDULEREPAIRS. Once repairs for this component have been issued it continues to check the following broken ones and repeats the process.

At this point all the critical repairs needed to resume operation have been scheduled and how long those repairs will take is known. The amount of time needed to perform the repairs (including the recovery and tuning) is the allowed repair time. This is the time used to perform non critical repairs. If stated so in the miscellanea input file the repair time can be extended in a proportional factor in order to take advantage of the shut down and repair some event that takes longer than our initial allowed time. This means that extending a few hours the down time (usually the factor used is 1.5 the allowed repair time) events that in the future could provoke another down time are repaired. In the miscellanea file is specified to allow access to the accelerator vault if the mean down time surpasses a certain amount of time (if the critical repairs don't require access to the vault, it cannot be accessed to perform secondary repairs).

Once the programs checks if the repair time is extended or the vault can be accessed it uses the IBANG matrix again to decide the events that are to be repaired. It prioritizes the ones with longer mean down time to take the maximum advantage of the current shut down. The repairs are scheduled calling the SCHEDULEREPAIRS routine as before.

When all repairs are scheduled it saves the list of the repaired components. It also saves how much the repair time was extended and which component's fault was it. The secondary repairs are also stored in order to estimate the down time we have saved performing those repairs.


START
Find broken comp.
Functions value
= design value
Apply degradation
to functions
Check critical functions
N
Brings downs the
accel by itself
Y
SCHEDULEREPAIRS
Y
More broken
comp. to check?
N
Functions value
= design value
Find broken comp.
Apply degradation
to functions
Check critical functions
N
Brings downs the
accel due to acumulated
degradation?
Y
SCHEDULEREPAIRS

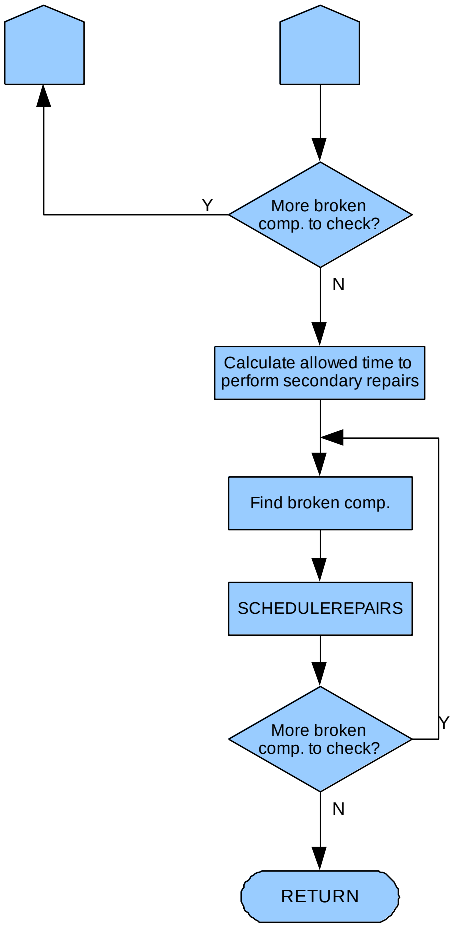


Fig.14 SETUPREPAIRS flow diagram

6.3.3.3 SCHEDULEREPAIRS

Called by: SETUPREPAIRS

SCHEDULEREPAIRS calculates when an event repair can be started based on the amount of manpower it requires and how many people are free to perform this repair. If it is a non-critical repair it also checks if the repair can be done in the allowed repair time. If the time required to do the secondary repair is longer than the allowed one then the component won't get repaired.

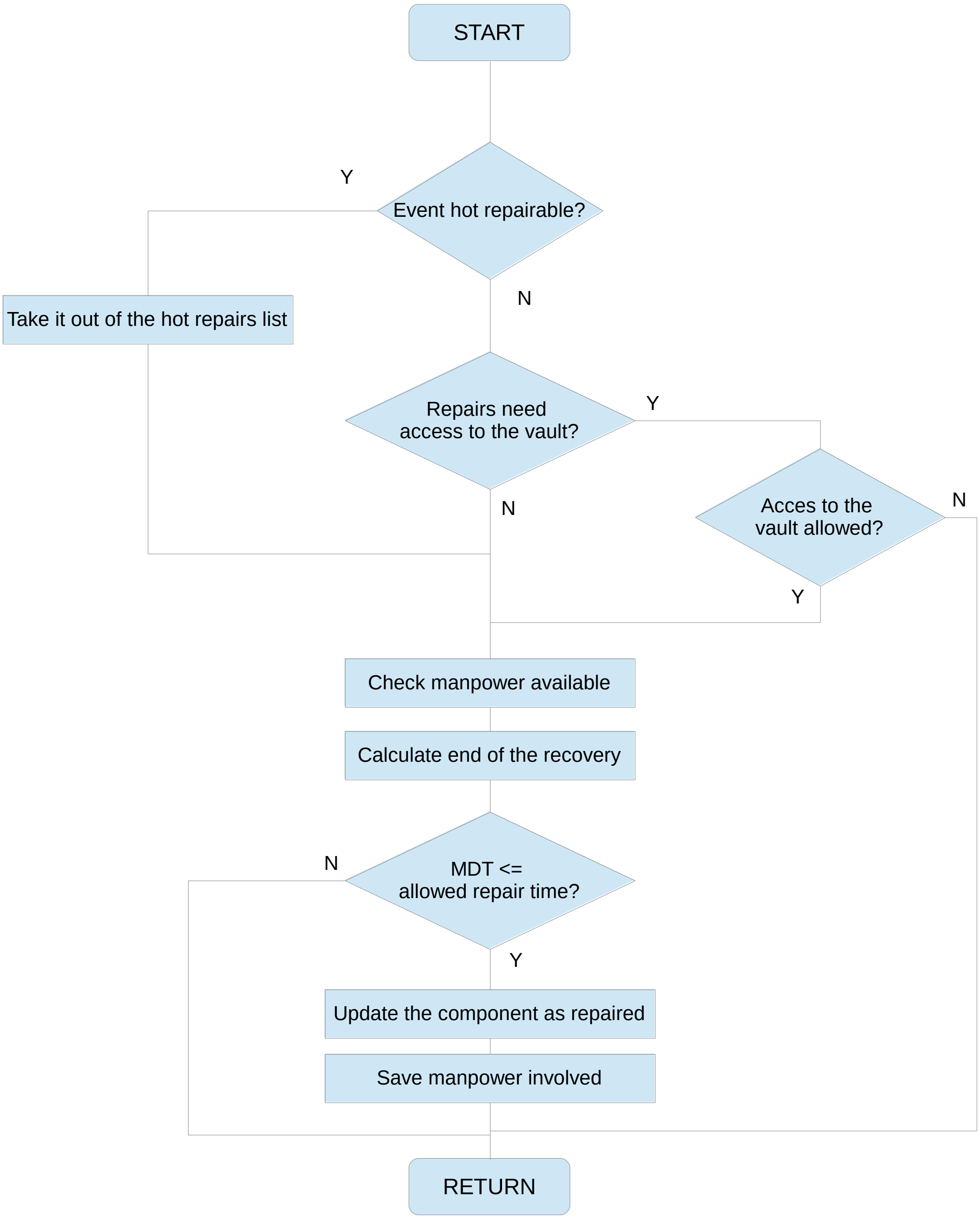


Fig.15 SCHEDULEREPAIRS flow diagram

Note that for critical repairs the allowed repair time is set to a huge number because those are repairs that need to be performed.

Once it has checked the component can be repaired it calculates the finishing time of the repair as the time when the event will finish the recovery. The end repair time that defines when the facility will finish its recovery is the one from the component with the latest end recovery time (if it does not surpass allowed repair time).

6.3.3.4 **RECOVERFROMREPAIR**

Called by: MAINLOOP

Routines called: RANDRECOVER

This routine takes the facility end recovery time and it adds and random extra tuning time based on the amount of time the facility has been down. It gives the definitive end tune time when the facility will resume operation.

6.3.3.5 **SETUPSCHEDDOWN**

Called by: MAINLOOP

Routines called: SCHEDULEREPAIRS

This routine checks the repairs are to be performed during a scheduled maintenance period. It sorts the events by the matrix IBANG prioritizing the ones with highest mean down time. The allowed repair time is the length of the scheduled maintenance period but it can also be extended to perform additional repairs. The maximum extended time is defined by the same factor used in SETUPREPAIRSS and is extracted from the miscellanea file.

Note that the full extent of the maintenance period will always be used. No restrictions about the manpower are included because it is assumed that during scheduled maintenance there is always enough manpower. Access to the vault is allowed. Despite this variation this routines performs as SETUPREPAIRSS.

6.3.3.6 **RECOVERFROMSHUTDDOWN**

Called by: MAINLOOP

This routine sets the time when the facility will be able to resume normal operation after an scheduled maintenance.

6.3.3.7 **SCHEDULEHOTREPAIRS**

Called by: MAINLOOP

The SCHEDULEHOTREPAIRS routine checks the time a hot fixable event will be able to be repaired and when the repair will end. It calculates when the required manpower to perform the repair will be available and sets it as the repair starting time. Then it calculates when the repair will be done.

6.3.3.8 **FACILITYSTATE**

Called by : MAINLOOP

The FACILITYSTATE routine decides if a facility is too degraded to continue operation after the failure of one or more components. To do this it loops through the broken components and applies their defined degradation to their associated functions. Next it proceeds to check the effect those degraded functions have onto other functions until it reaches the critical functions. These functions are the ones that determine the state of the facility.

If there are no broken components then the state is automatically up.

Then the software calculates the difference in intensity output between stopping and repairing every event or continue degraded operation. If the output is superior with repairs then the facility will be shut down and repaired.

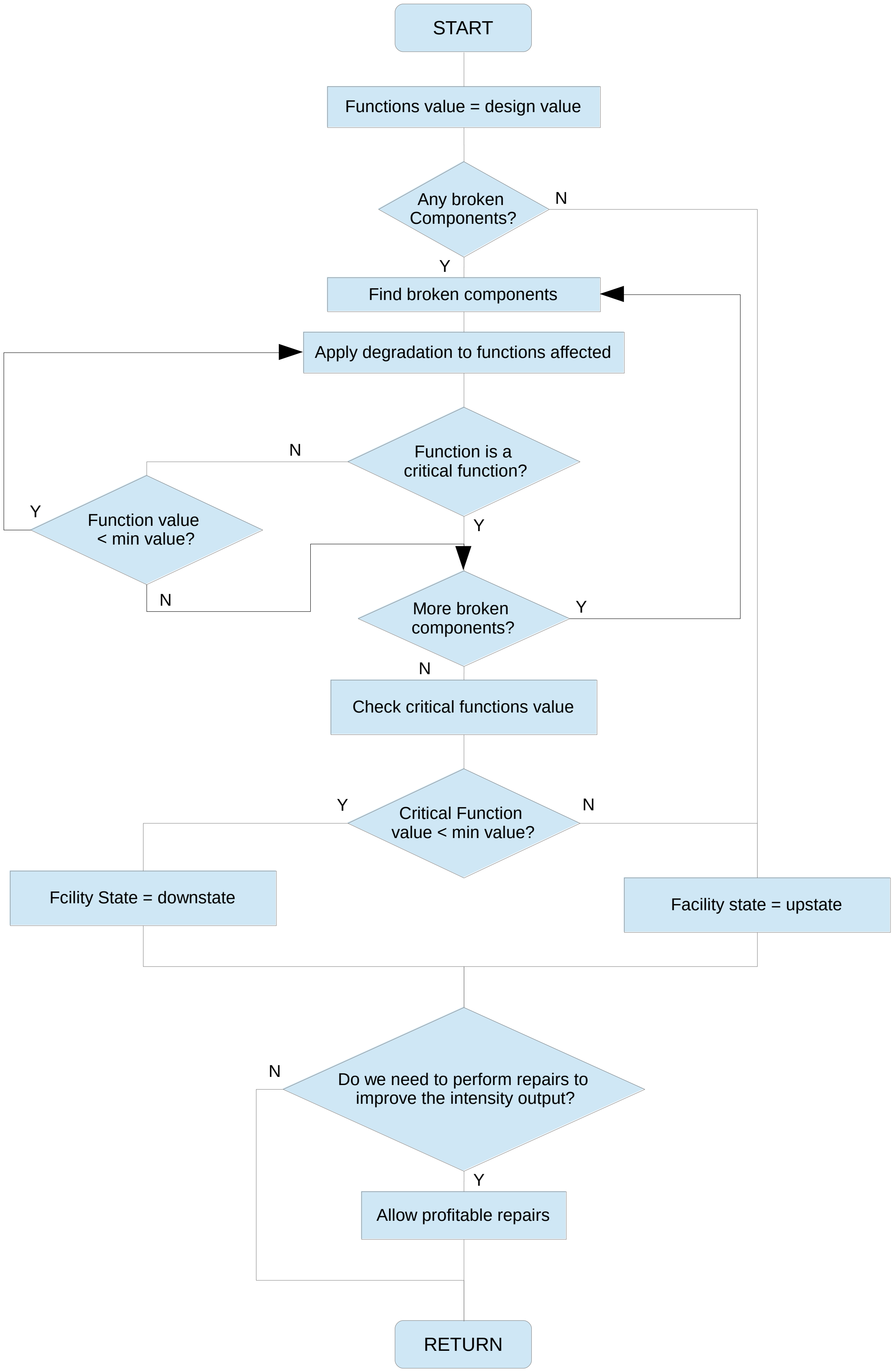


Fig.16 FACILITYSTATE flow diagram

6.3.4 Result routines

6.3.4.1 SAVEYEAR

Called by: MAINLOOP

This routine doesn't have any effect on the simulation development. It serves the purpose of being able observe the evolution of the availability through the simulation period. It saves for each facility the up hours, down hours and maintenance hours every year individually.

6.3.4.2 SAVERESULTS

Save results store the data needed from every iteration.

6.3.4.3 PRINTRESULTS

This routine prints the results in the XLS results file.

6.3.5 Utility routines

6.3.5.1 RANDEXP

Called by: MAINLOOP

Although it is a very simple routine it serves a very important role in the simulation. RANDEXP gives repaired events a new failure time. To do this it creates a random flat number using the event own seed. Then using the inverse of the integrated probability density function of an exponential and taking the mean time between failures as the average time returns the time when this component will fail.

6.3.5.2 RANDRECOVER

Called by: MAINLOOP

This routine generates a random number with a distribution intended to represent the distribution of recovery time from repairs. It also uses the inverse of the integrated probability density function of an exponential. The mean used to center the distribution is the amount of time the facility has been down multiplied for the factor introduced in the miscellanea file. This value represents that the more a facility is down the more it takes to bring it up again.

6.4 Data output

The results of the simulation are automatically exported at the end of the operation to an .xls file with the name specified in the miscellanea file. This file contains 5 sheets (6 if the history was saved). Every sheet is independent.

6.4.1 Facility

The facility sheet contains the general results of the simulation. Each line belongs to a facility. Keep in mind that every result is obtained performing the mean of all the iteration 'results. The results shown in the different columns are:

- Facility: The number of the facility the results belong to.
- Uphours: The total amount of hours the facility has been up and working. Regardless if the operation was degraded or at full power.
- Downhours: The total amount of hours the facility has been down due to an emergency shutdown. This time doesn't include the hours dedicated to scheduled maintenance.
- Scheduled maintenance hours: As it names says this value is the amount of time the facility has spent performing scheduled maintenance. It includes both the long scheduled period and the short scheduled period.
- Operational availability: This is the total availability of the facility calculated as the up hours divided by the total time simulated. It includes the scheduled maintenance time as time the facility has been down.
- Hardware availability: This is the facility's availability without taking into consideration the time the facility has spent in scheduled maintenance. I t will be always superior to the operational availability.

- Accesshours: The amount of time the vault of the facility has been in access.
- Extended hours: The total hours we have extended the repairs to perform non critical repairs. This time is already contained in the downhours field and it is purely informative.
- Used down hours for scheduled maintenance: The total amount of hours that we have used during non-scheduled down times to perform scheduled maintenance. This time is already included in the scheduled maintenance hours field. It allows us to calculate the amount of time we have saving by advancing a scheduled maintenance to take advantage of a non-scheduled shut down.
- Standard deviation of the availability

6.4.2 Events

The events sheet contains every single one of the events involved in the simulation and its results. It also shows the fixed parameters of each event. This values have been obtained by calculating the mean of all iterations results for each field.

It contains the following columns.

- Name: the real name of the event
- Facility: The facility the event belongs to.
- ID: The event's identification code.
- Nfailuresnotignored: How many times this event has occurred and it hasn't been ignored.
- Nfailurescausingdown: The amount of times this event has fiorced a shut down.
- Down hours caused: The amount of hours this event has provoked to the facility. It only counts the time spent in shut downs caused by this event.
- Repair hours incremented: The hours this component has extended a down time in order to be repaired. This field and the down hours caused contain the full amount of down hours caused by the event.

- Opportunity repair hours: The hours this event has been subjected to repairs while the facility was shut down by a different event.

- Nfailuresignored: The amount of times this event has been ignored due to a failure during a down or maintenance period.

The rest of fields contain the initial information extracted from the event's input file. It allows us to be able to classify the events by many field and look for correlations between events with similar parameters.

6.4.3 Functions

This is the functions results sheet and it provides the mean results of all iterations for each function the same way the events sheet does. It contains the following columns:

- Name: the name of the function.

- Meanvalue: The value the function has taken during the operation of the facility. Note that it only takes into consideration the function's value when the facility is up.

- Timesdown: Hown many times this function has been down, meaning that it acquired a value below its minimum one.

  We include the design value and the minimum value tolerable to compare with the mean value obtained.

6.4.4 Year

This result sheet is pretty straight forward. It contains each facility's mean hardware availability value for each year. It is not an accumulated value meaning that each availability is calculated in that year's period.

Each line is a facility and each column a year.

6.4.5 Iteration

This sheet provides the operational availability results of each iteration for each facility. The columns represent the different facilities and the rows the iteration.

6.4.6 History

The history is a complete log of every action performed by the simulation. It is great for understanding its operative and checking for errors. But it has a big downside, a single iteration can fill several thousand rows, which means that the memory charge becomes huge for Matlab® to handle and easily surpasses the maximum matrix dimension tolerated by excel. For this reason is almost exclusively used for one iteration simulations. Nevertheless it is still a great tool.

7 Assumptions, hypothesis and parameters of the simulation

There are some considerations to take when performing a simulation with Availsim that one must know to fully understand its results:

7.1 Availsim considerations

- When the accelerator is shutdown, failing components are not taken into consideration. Instead, its failure is ignored (but registered) and given a new failure time. This will continue until the accelerator is up again.
- Long non-scheduled maintenance can pose difficulties in order to bring up the accelerator. In order to account for this and reflect that the longer the down period the harder is to bring up the accelerator a factor of extra 20% time of the MDT is applied to the non-scheduled maintenance periods.
- The component charged with the down hours is the one that has caused the down. Independently of how much degradation it applies to the beam.
- When a scheduled maintenance period is performed during a non-scheduled shutdown, the amount of time spent on the scheduled maintenance is subtracted from the down hours caused by the component that caused the down.
- With the current input data, the two facilities only interfere with each other with the manpower. In future analyses, common auxiliaries and other facilities could be included.
- The value a function takes when the accelerator is down is not taken into consideration when calculating its average value.
- The MTTR is the same a repair team would need in the repair shop[3].
- Availsim doesn't make a distinction whether a component is repaired or changed[3].

- Remote handling has not been modeled in Availsim, for that reason MTTR are constant and does not include randomization[3].
- The access time to begin repairs already includes detection, cooling and acces time[3].
- The recovery time after a repair includes the recovery time itself, the tuning time and the 20% factor that increment the duration of the shutdown depending on the downtime[9].
- Only one restricted region is modeled, the vault. The rest of the facility does not require to be allowed access.

7.2 Simulation parameters

- The span of the simulation is 30 years. This elevated value is to allow components with low failure rates to fail and observe its consequences.
- The simulation assumes the accelerator has reached steady state after years of operation. For this reason infant mortality and fatigue failures are not modeled[5].
- 2 scheduled maintenance periods are included. A short one with duration of 3 days that will happen at the middle of the year. And a Long one with one moth of duration that will take place at the end of the year. This maintenance period duration cannot be decreased because it is the exact time needed to perform crucial actions like changing the test materials[4].
- The manpower available has been set to a high number in order to not be a limiting factor in this simulation.
- If a repair takes more than 12 hours access to the Vault will be granted[3].
- Down times can be extended up to 150% its original duration in order to perform additional repairs.

8 Error and number of iterations

As explained before Availsim originally performed a single iteration. This served its original purpose but the IFMIF Rami team required more precision and the capacity to perform multiple iterations was implemented. The number of iteration required to attain a certain error is a function of the simulation itself, the input data and the error desired.

8.1 Distribution

In order calculate the number of iterations required the distribution of the result data must be studied. For that reason a 400 iteration simulation was performed. The output parameters studied were the hardware availability and operational availability.

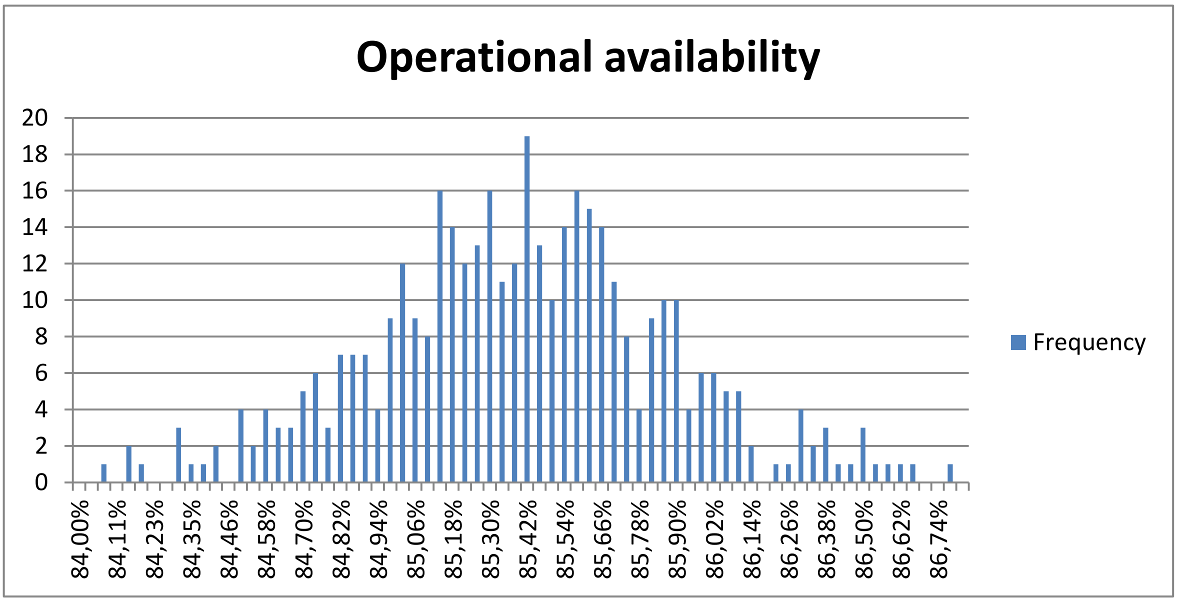


Fig.17 Operational availability distribution

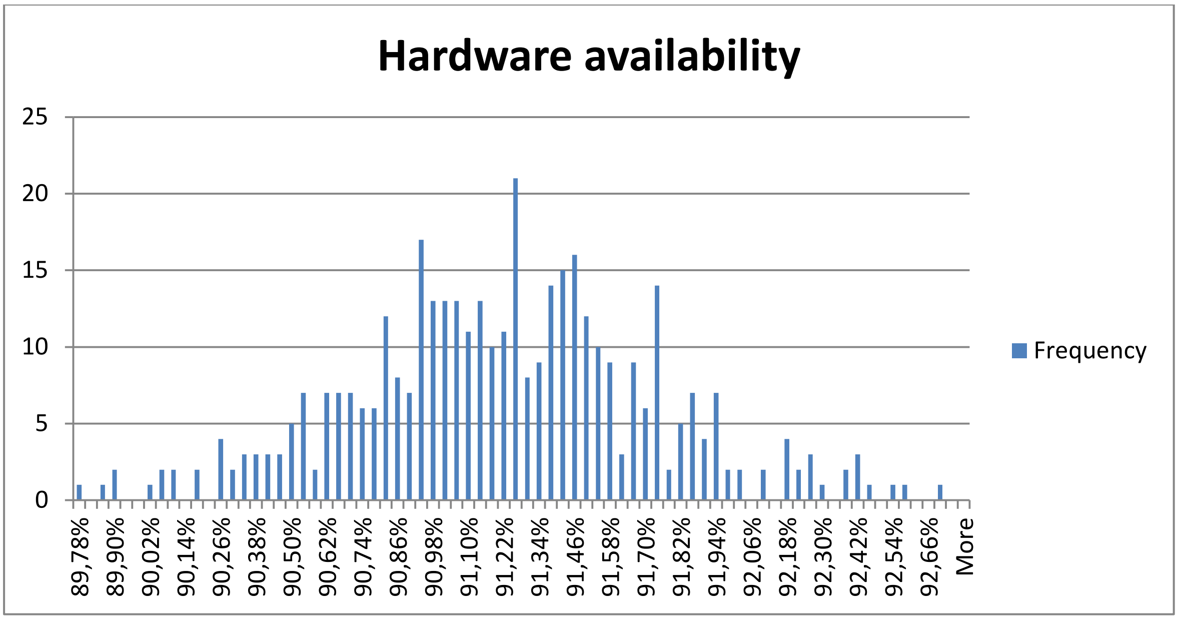


Fig.18 Hardware availability distribution

In histograms Fig.17 and Fig.18 the tendency is easily noticeable as a normal distribution. The validation of the distribution is needed in order to be able to rely on the data output. If the results do not follow a distribution the conclusion extracted from them will not be reliable. The results outputs could follow another distribution rather than a normal one dependent on the data input.

The operational availability distribution is centered on its mean 85,38% and the data output has a standard deviation of 4,77E-03.

The hardware availability is centered around its mean 91,19% and the results obtained had a standard deviation of 5,10E-03.

The point of performing an elevated number of iterations is to achieve a stable value for the standard deviation. The standard deviation is the parameter that will ultimately provide the error of the results. The lower this parameter gets the lower the error will be. The way to achieve this is to perform more iterations. But it will reach a point when the standard deviation no longer decreases and remains stable.

8.2 Error

All the data obtained from a Montecarlo simulation contains certain error. This error must be bounded in order to reflect the random factor the simulation has in it. A level of confidence must be decided in order to provide this error. The level chosen is 90%. This level was decided because it is the same level of confidence used in the RiskSpectrum® calculations and it was considered sufficient by the RAMI team[3].

The error that these results entail is calculated using the following formula obtained from[11] :

$$\varepsilon = z\alpha/_2 \cdot S_0 \cdot \frac{1}{\sqrt{n}}$$

(Eq. 8)

Where:

- ε is the standard error.
- $z\alpha/_2$ is the z-value of the (1-(α/2)) percentile of the standard normal distribution for the level of confidence(LOC) chosen looked up in the normal distribution tables.
- S_0 is the standard deviation of the samples.
- n is the number of iterations performed.

The data used to calculate the error is the one obtained from the previous 400 iteration simulation [chapter 8.1].

	n	LOC	Mean	$z\alpha/_2$	S_0	$\pm\varepsilon$
Oper. Avail.	400	90%	85,38E-2	16,45E-1	4,77E-03	3,93E-4
Hard. Avail.	400	90%	91,19E-2	16,45E-1	5,10E-03	4,19E-4

Table-2 Availability error

Table-2 shows that the operational availability obtained from the simulation is 85,38%±0,04%. So we can assure with a 90% level of confidence that the operational availability obtained will be comprised into the interval from 85,42% to 85,34%.

The same way we can assure with a 90% of confidence that the hardware availability will be between 91,24% and 91,15%.

8.3 Iterations

The error has been calculated for a number of iterations. But what is interesting is to define an accepted error and calculate how many iterations would be needed to achieve that error given a certain level of confidence. The procedure to calculate the number of iterations is the reverse of calculating the error. The error is set and given a standard deviation and level of confidence the minimum number of iterations are obtained. The standard deviation used is the one obtained from a simulation with an elevated number of iterations.

The number of iterations is calculated with this formula which a transformation from the previous formula used to calculate the error [11].

$$n = \left(\frac{z\alpha/_2 \cdot S_0}{\varepsilon} \right)^2$$

(Eq. 9)

Where:

- ε is the maximum error accepted.
- $z\alpha/_2$ is the z-value of the (1-(α/2)) percentile of the standard normal distribution for the level of confidence(LOC) chosen looked up in the normal distribution tables.
- S_0 is the standard deviation of the samples.
- n is the minimum number of iterations obtained.

Using the S_0 obtained from the 400 iteration simulation and the previous formula the number of iterations required for a specified maximum error can be estimated. The error accepted as maximum was $\pm$1E-3.

	LOC	ε	$z\alpha/_2$	S_0	n
Oper. Avail.	90%	$\pm$0,001	16,45E-1	4,77E-03	62
Hard. Avail.	90%	$\pm$0,001	16,45E-1	5,10E-03	71

Table-3 Number of iterations required

Therefore at least 71 iterations would be needed to achieve an error no larger than 1E-3. In the end this result is more of an indicative value than an absolute one. The reason is that the standard deviation used to calculate the number of iterations comes from a 400 iteration simulation. So the standard deviation could be higher for 71 iterations than for 400 iterations. For that reason 80 iterations are chosen as enough repetitions to obtain reliable data.

8.4 Validation of the simulation

After choosing a number of iterations there is the need to validate that the simulation meets the required precision. For this reason 2 checks must be performed. Firstly the results must meet the level of confidence established. Secondly the error obtained must be equal or inferior to the one selected.

8.4.1 Confidence interval

The 90% level of confidence was chosen previously in order to obtain the same precision obtained with RiskSpectrum® results. After performing the simulation with 80 iterations the results must be checked in order to validate they meet the level of confidence. To do so the confidence interval must be calculated. The interval is calculated using the following formula extracted from [11].

$$\bar{X} \pm \theta = \bar{X} \pm z\alpha/_2 \cdot S_0 \cdot \sqrt{1 + \frac{1}{n}}$$

(Eq. 10)

Where:

- $\bar{X}$ is the average of the availability
- θ is the confidence.
- $z\alpha/_2$ is the z-value of the (1-(α/2)) percentile of the standard normal distribution for a 90% level of confidence.
- S_0 is the standard deviation of the samples.
- n is the number of iterations performed in this case 80.

These are the confidence intervals obtained:

	n	LOC	Mean	$z\alpha/_2$	S_0	$\pm\theta$
Oper. Avail.	80	90%	0,86	16,45E-1	4,76E-03	7,88E-3
Hard. Avail.	80	90%	91,86E-2	16,45E-1	5,08E-03	8,41E-3

Table-4 Confidence intervals

So taking the results into consideration one has to be able to assure with a 90% of confidence that all the values obtained from the iterations are contained into the confidence interval.

The confidence interval for the operation availability is [85,21% 86,78%]. While the confidence interval for the hardware availability is [91,02% 92,70%]. It must be checked that from the 80 samples obtained from the simulation for each parameter, at least 90% of them are included into the calculated interval.

For both the operation and hardware availability there are 7 samples outside the confidence interval. For an 80 iteration simulation that means 91,25% of the samples belong into the interval. This meets the requirement of 90% confidence so the results of the simulation are accepted.

8.4.2 Error

Once the confidence interval has been checked the error must be calculated for each value following the same procedure explained in [chapter 8.2].

	n	LOC	Mean	$z\alpha/_2$	S_0	$\pm\varepsilon$
Oper. Avail.	80	90%	0,86	16,45E-1	4,76E-03	8,76E-4
Hard. Avail.	80	90%	91,86E-2	16,45E-1	5,08E-03	9,35E-4

Table-4 Error for 80 iterations

In both cases can be observe that the error doesn't surpass the maximum specified in [chapter 8.3], that was of 1E-3. Therefore 80 simulations have been proven enough to satisfy both the confidence level desired and the maximum error accepted.

9 Results of the simulation

Once the results of the previous 80 iterations simulation have been validated as reliable they must be analyzed. These results were obtained using the last set of data for the IFMIF design. The simulation at hand has been validated in [chapter 8] and 80 iterations have been performed.

9.1 Availability

The first result to be analyzed is the availability. Availsim provides the availability for every facility involved. In IFMIF case it simulates 2 parallel accelerators called facility 1 and facility 2. Since each accelerator provides 50% of the beam availability the global availability of the accelerator is calculated using [Eq.3]:

$$\frac{Availability_{facility1} + Availability_{facility2}}{2} = Availability_{total}$$

Table-5 Availability results

facility	Up hours	Down hours	scheduled maitenance hours	operational availability(%)	hardware availability(%)
1	224655,62	21394,74	16750,63	85,49	91,30
2	227351,64	18663,02	16786,33	86,51	92,41
Global				86,00±8,76E-4	91,86±9,35E-4

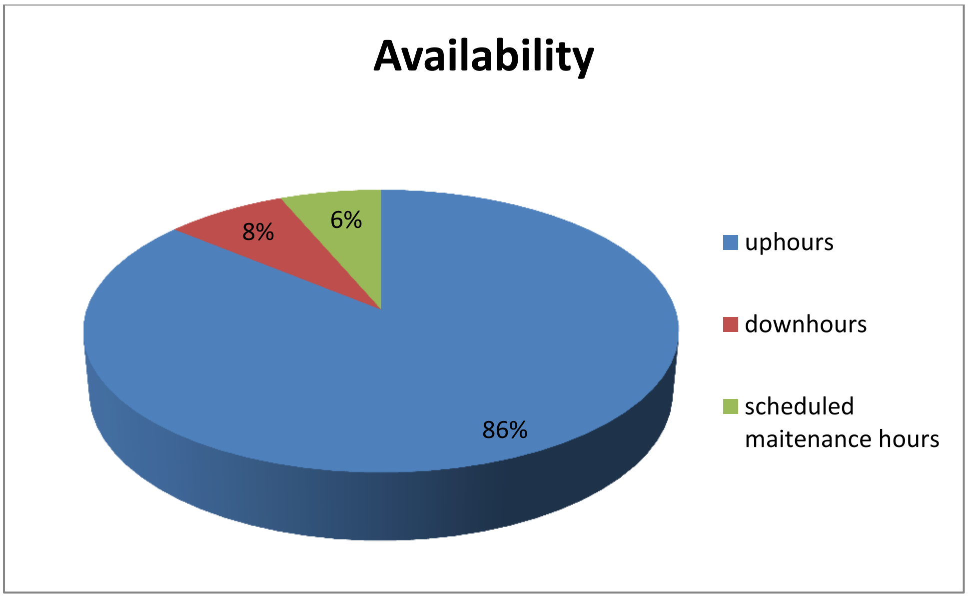


Fig.19 Availability results

Other data for the facilities is displayed.

facility	Access hours	Extended down hours	Down hours used for sched. maint.	Vault accesses
1	10939,32	564,52	547,27	126,50
2	8328,96	652,86	425,12	119,23

Table-6 additional results

Access hours are the average amount of time the vault has been in access due to repairs. The extended repair hours are amount of time the maintenance periods (scheduled and nonscheduled) have been extended in order to perform more repairs during the accelerator operation time.

The down hours used for scheduled maintenance is an interesting informative value. It's the time that has been saved due performing scheduled maintenance during a nonscheduled one. So effectively 547 h and 425 h of down time have been saved for facility 1 and 2 respectively. That equals to a net hardware availability profit of 0,22% for facility 1 and 0,17% for facility 2.

In table-6 is observed that during the 30 years of operation the facility 1 vault has been accessed 126 times while the vault in facility 2 has been accessed 119 times. It can be observed that there is a relation between the access to the vault and the availability. Repairs performed in the vault require longer time than most of the repairs on the accelerator. So the facility 2 has been accessed an average of 8 times less than facility 1. This fact has influenced the higher availability of facility 2 respect facility 1

9.1.1 Hardware availability per system

Availsim provides the average amount of hours provoked by every event. If the user has classified the events by different systems the unavailability for each system can be obtained. The following table shows the results for both facilities.

System	Average down hours provoked (h)	Contribution to unavailability (h)	Unavailability (%)	Total availability (%)	Availability probability (%)
Diagnostics	524,27	1,31%	0,11%	99,89%	99,85%
HEBT	4607,01	11,50%	0,94%	99,06%	99,02%
Injector (& LEBT)	3447,60	8,61%	0,70%	99,30%	99,26%
MEBT	4960,86	12,38%	1,01%	98,99%	98,95%
RF System	10054,30	25,10%	2,04%	97,96%	97,92%
RFQ system	3638,94	9,08%	0,74%	99,26%	99,22%
SRF Linac	12824,73	32,02%	2,61%	97,39%	97,35%
Total	40057,76	100%	8,14%	91,86%	91,86%

Table-7 Hardware Availability per system

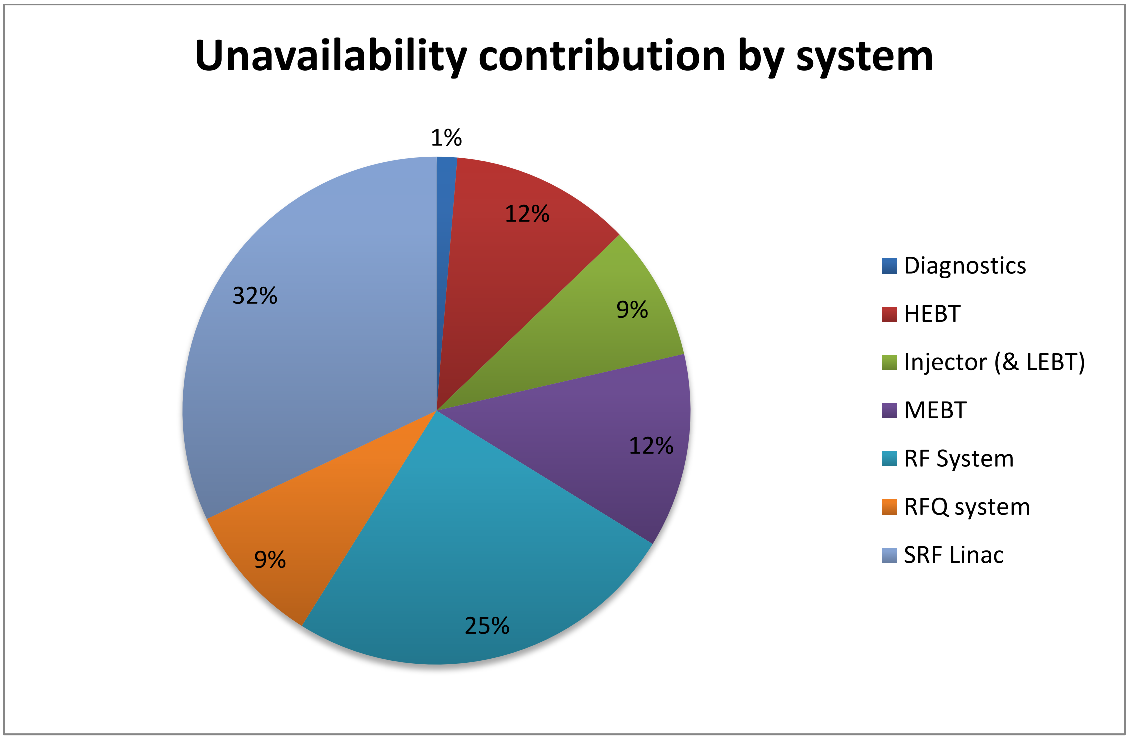


Fig. 20 Availability results by system

It can be observed in Fig. 20 that the systems that have a deeper impact on the availability by far are the RF system and the SRF Linac. Availsim calculates total availability for each system not its probability to fail. For this reason to be able to compare availability results for each system with other software like RiskSpectrum® first the availability results must be converted to the probability of availability. While the Availability probability for each system is different from the actual availability, the global availability of the accelerator will be the same in both cases.

Taken it one step further it can be checked which events are the main cause of the unavailability.

9.1.2 Events down hours provoked

Taking a look into the events sheet one can obtain result about how an event and by extend a component affects the availability of the accelerator.

Events (failure modes)	Average down hours provoked	Average failures per run
Loops board	1696,33	706,79
Solid State RF Pre-driver	1691,25	469,83
Beam vacuum valves (assumed normally closed)	1579,55	3,69
Solid state RF amplifier Common Cause Failure	1472,43	25,58
Feedthroughs (vacuum leak)	1401,92	11,59
Power supply	1370,57	1149,28
Signal module	1259,35	117,23
Step motor	1146,37	6,13
Electrical wire (Step motor power)	1139,95	55,29
Hoses and their fittings	1125,26	40,06
RF vacuum window (ceramic)	1091,22	1,90
Flexible membrane (Niobium-Titanium alloy)	1028,84	1,76
Solenoid valve	1012,76	61,49
Power Cables 30m	999,26	25,34
PLC	997,58	332,79
Turbomolecular pump	976,04	180,80
RF window	953,87	17,33
Control cable connector	943,00	185,60
Step motor (detune cavity)	933,12	12,10
Acquisition modules	860,93	41,93
Power Cables 5m	815,51	142,61
Low voltage power wires and conectors	791,62	78,06
Power Cables	769,05	194,98
PS	708,67	529,01
Spliter	701,92	22,99
Electrical connection (Step motor power)	665,10	39,89
Pipes (water)	656,74	18,82
Welds HWR structure	622,85	1,13

Table-8 Events down hours

This is just a fraction of the events listed but they are the ones affecting the most the accelerator. If the one stat focused are the down hours provoked it can be observed that events like a loops board failure happens constantly. On the other hand events like beam vacuum valves failures which are shown to happen less than 4 times per run add almost the same amount of down hours due to their elevated mean down time to recover.

This table results can point the design team in the correct direction to improve the availability of the accelerator by adding redundancies or directly improving the reliability of a component.

9.1.3 Events opportunity repairs

Another interesting data that can be obtained is how many hours a component has been repaired during down times provoked by other components.

Events(failure modes)	Average down hours provoked	Average failures per run	Average opportunity repair hours
Electrical wire (Step motor power)	1139,95	55,29	5272,58
Electrostatic sensor	186,81	69,17	4599,61
Turbomolecular pump	976,04	180,80	3897,03
Electrical connection (Step motor power)	665,10	39,89	3666,30
Control cable connector	943,00	185,60	2936,56
Step motor (no reponse)	461,25	24,63	2259,68
Titanium sublimation pump	251,37	72,65	1573,93
Solenoid valve	1012,76	61,49	1067,14
Electronic Front End	111,92	88,96	984,69
Step motor (detune cavity)	933,12	12,10	980,10
Access traps and doors	16,44	6,20	617,10
Hoses and their fittings	1125,26	40,06	566,66

Table-9 Events opportunity repairs

In this table it can be observed which components benefited the most of repairs during down times not provoked by them. Especially interesting are the cases of events with relatively low down hours provoked but high opportunity repair hours such as the case of the electrostatic sensors. It is important to take these events into consideration. As improvements are applied on the events that cause most of the down hours, these event won’t be able to be repaired during the down time thus eventually provoking down hours on their own.

The total opportunity repair hours used in this simulation are:

- Facility 1: 14463 h
- Facility 2: 16009 h

These values mean that 67,60% of the downtime in Facility 1 has been used to perfrom additional repairs. For facility 2 it has been used 85,78% to perform additional repairs.

9.1.4 Events ignored

Due to the methodology of the simulation, during down times failures are ignored. So the amount of failures ignored must be checked.

Events(failure modes)	Average failures ignored
Power supply	187,425
Loops board	113,65
PS	86,0875
Solid State RF Pre-driver	77,15
PLC	53,55
Power Cables	31,7875
Control cable connector	30,1875
Turbomolecular pump	29,975
Power Cables 5m	23,9125
Low voltage power wires	20,675
Diagnostics board	20,0125
Power Cables 10m	19,9
Signal module	18,05
Electronic Front End	14,5875
Low voltage power wires and conectors	12,45
Thermocouple	12,225
Titanium sublimation pump	11,7125
Electrostatic sensor	11,575
Power supply	11,4125

Table-10 Events ignored failures

It can be observed how there are events that have a lot of failures ignored. Due the limitations of the software it is impossible to know the effect those failure would have on the availability. However it is important to check if the event with more ignored failure have high mean down time. In this case the most ignored events had all relatively low MDT so it assumed that they wouldn’t have a great impact on the availability.

9.2 Beam effectiveness

Availsim provides the average value of functions during the operation. And since operation parameters are set as functions it allows extracting the average beam effectiveness of the accelerator,

9.2.1 Intensity

As has been explained before the raw availability of the accelerator is not the only parameter to maximize. The quality of the beam provided is as important as the hardware availability. In the IFMIF case the BEAM effectiveness is defined by the amount of dpa the beam provokes. While this would be the ideal way to calculate it, it was too complicated to obtain an exact relation on how each operation parameter affected the dpa output. For this reason Intensity was chosen as an indicator the beam effectiveness as it is a parameter that has a direct relation on the dpa produced.

The table-11 displays the values obtained for the operation parameters during a 30 year run.

Facility	function	Average value	Up hours	Times down	design value	min value
1	Energy	40,59	224668,57	0,30	41,00	38,00
1	Intensity	119,88	224668,57	0,00	125,00	65,00
1	Eoverh	0,64	230925,30	768,56	1,00	0,00
2	Energy	40,48	227365,33	0,56	41,00	38,00
2	Intensity	116,62	227367,30	0,00	125,00	65,00
2	Eoverh	0,62	215971,13	1125,93	1,00	0,00

Table-11 Intensity and energy values

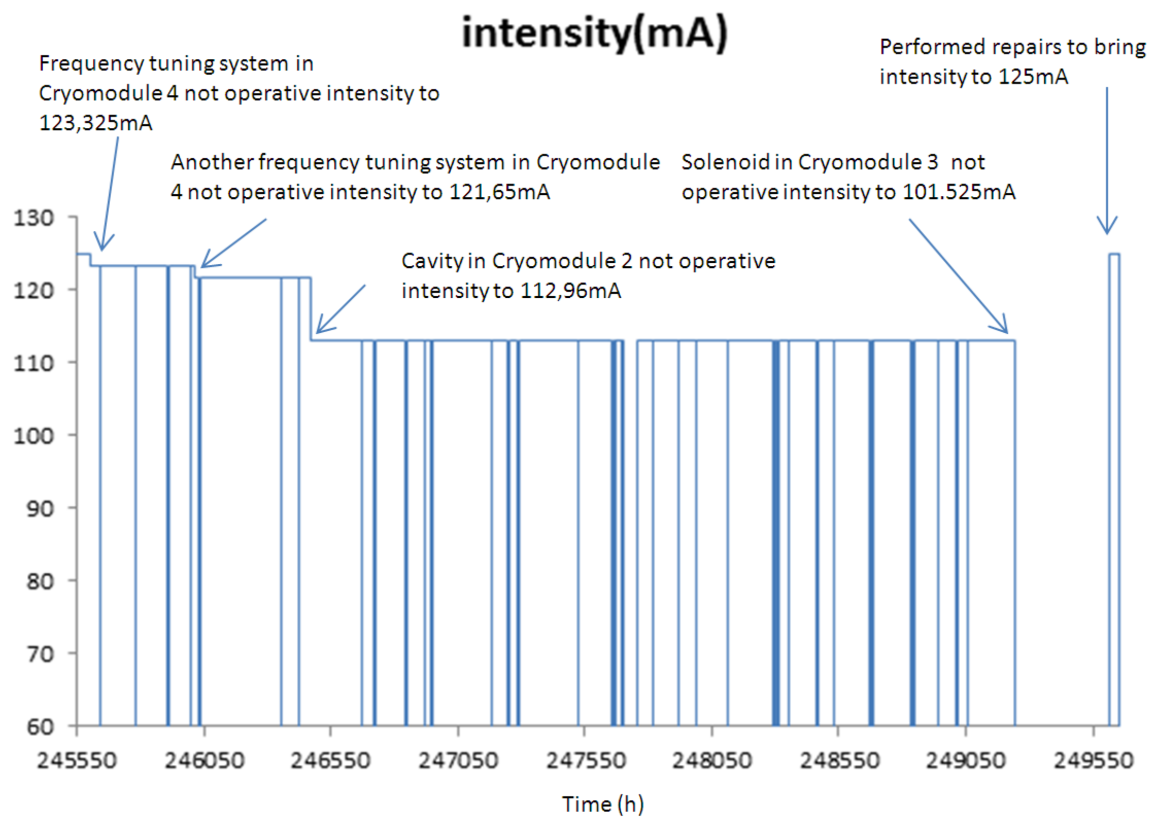


Fig.18 Intensity progression

In fig. 18, the progression and treatment of the intensity by AvailSim can be observed. Initially the intensity has its design value of 125mA. There is a failure in the frequency tuning system of a cavity in the cryomodule 4, which would bring the intensity down to 123.325mA. But this is not enough degradation to bring down the accelerator. A second frequency tuning system in cryomodule 4 fails and applies further degradation bringing the intensity down to 121.65mA. It is still good enough to continue operation.

In the next event, there is a failure in a cavity from cryomodule 2 which degrades the intensity down to 112.96mA. AvailSim has to check if it will be more profitable to stop the accelerator and perform the needed repairs in order to bring the intensity back to 125mA instead of maintaining operation with this amount of degradation. The average intensity without performing repairs is the actual intensity value 112.96mA. The average intensity that includes stopping the accelerator and performing repairs is 118.25mA. But if the 'x' parameter is applied the average intensity required to perform the repairs has to be above 124,86mA. So degraded operation continues.

In the following event, a solenoid from cryomodule 1 fails and intensity is further degraded down to 101,52mA. AvailSim checks again if repairs are to be performed. The average intensity without repairs including the extra 10% would be 111,7mA. The average intensity obtained by performing repairs would be 114mA. So, repairs are scheduled and the intensity is brought back to its design value 125mA.

During the operation the accelerator has suffered numerous shut downs. Although, as can be seen in the figure, the longest down period of the facility (in the time lapse exposed) is the one needed to repair the intensity output. That is why too many stops to improve the intensity output take a heavy toll on the hardware availability.

The average value for each parameter is calculated only when the facility is operating. Since the Intensity is the parameter that defines the beam effectiveness for each facility the results are exposed in table-12.

Facility	Beam effectiveness(%)
1	95,90±0,12
2	93,30±0,17
total	94,68±0,15

Table-12 Beam effectiveness

9.2.2 Energy

It is also interesting to observe the value of the energy. Although the design energy output is 40 MeV, the accelerator is able to provide an extra 1 MeV of energy output. Since this extra energy will be used only if the energy value is below 40 MeV a new functions was added. Energy overhead is the function that reflects the extra energy the accelerator can provide. So to obtain the average operation value of the energy the value of the energy overhead must be deducted from the value of the energy function.

Facility	Average operational Energy (MeV)	Average Energy overhead (MeV)	Average real energy (MeV)
1	40,59	0,64	39,95
2	40,48	0,62	39,85

Table-13 energy value

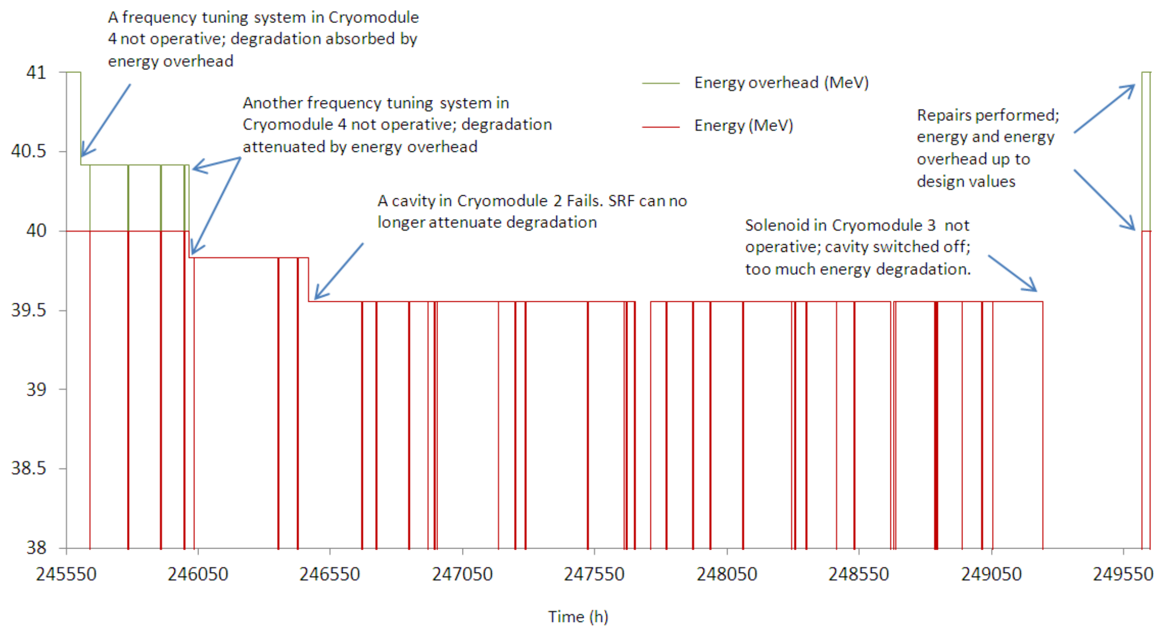


Fig.19 energy progression

The figure above shows the energy output progression during the same time of operation as figure 19. When the frequency tuning in cryomodule 4 fails the consequent degradation is absorbed by the energy overhead. The energy stays at 40 MeV and the accelerator continues operation. There is another failure in a frequency tuning system from the same cryomodule and further degradation is applied. This degradation cannot be fully absorbed by the energy overhead but it is attenuated and brings down the energy value to 39.83 MeV. A new failure in cryomodule 2 increases the amount of degradation. However, the full energy overhead is being used and cannot attenuate the new degradation. The energy output is brought down to 39.55 MeV, which is more than the limit of 38 MeV. The accelerator continues to operate degraded until it is shut down to perform the repairs to improve the intensity output. After the repairs, the energy and energy overhead are both brought to their design values of 40 MeV and 1 MeV respectively.

9.3 Beam availability

The beam availability is the value this study was designed to calculate. It is the parameter that relates the availability of the accelerator with its performance. It is calculated by the following expression Eq. 4.

$$HA_f \cdot BE_f = BA_f$$

Where HA_f is the hardware availability, BE_f is the Beam effectiveness and BA_f is the beam availability. The results obtained are:

Facility	HA(%)	BE(%)	BA(%)
1	91,30±0,14	95,90±0,12	87,57±0,23
2	92,41±0,10	93,30±0,17	86,22±0,25

Table-14 Beam availability

Since each facility contributes 50% to the beam output of the accelerator the global beam effectiveness for IFMIF is:

$$\frac{BA_{f1} + BA_{f2}}{2} = 87{,}13\% \pm 0{,}24\%$$

(Eq. 11)

Trips were not included in this calculation. The RAMI team performed an estimation of the availability loss for each accelerator due the trips. The result obtained was a loss of 2,5% beam availability[12]. The final result of the beam availability is:

$$87{,}13\% \cdot 97{,}50\% = 84{,}95\% \pm 0{,}23\%$$

BA (%)	Availability loss due trips (%)	BAfinal (%)
$\mathbf{87,13 \pm 0,24}$	97,5	$84{,}95 \pm 0{,}23$

Table-15 Final beam availability

It is interesting to compare these results performing the same simulation without the restriction of maximizing the energy output. The results are the following:

HA(%)	BE(%)	BA(%)	BAfinal (%)
92,50	86,25	79,78	77,79

Table-16 Final beam availability without maximizing energy output

As can be observed despite the hardware availability being superior the beam effectiveness decreases drastically. For this reason the beam effectiveness without maximizing the energy input is inferior. From this can be deduced that while trying to reach a higher intensity output may lead to inferior hardware availability it will improve the beam effectiveness in the end.

9.4 Other results

Besides the data required to perform the calculation of the beam availability Availsim provides other information with more or less utility depending on the precision used in the inputs.

9.4.1 Manpower

If the data input on the number of manpower required for each repair is accurate, Availsim provides the maximum number of workers ever needed for one nonscheduled down time. This allows the planning of personnel in order to be able deal with the needed repairs and not extending down times due to the lack of manpower.

In this simulation case the maximum number of workers ever needed for a repair was 9. It can be interesting the check the distribution of the manpower during the operation time. Due to the heavy need of resources that would be needed to check the manpower required in every downtime for all iteration at once this can only be done with a single iteration as a time. For this reason the analysis lacks the statistical reliability to obtain reliable results. Nevertheless it is still interesting to observe the manpower used evolution.

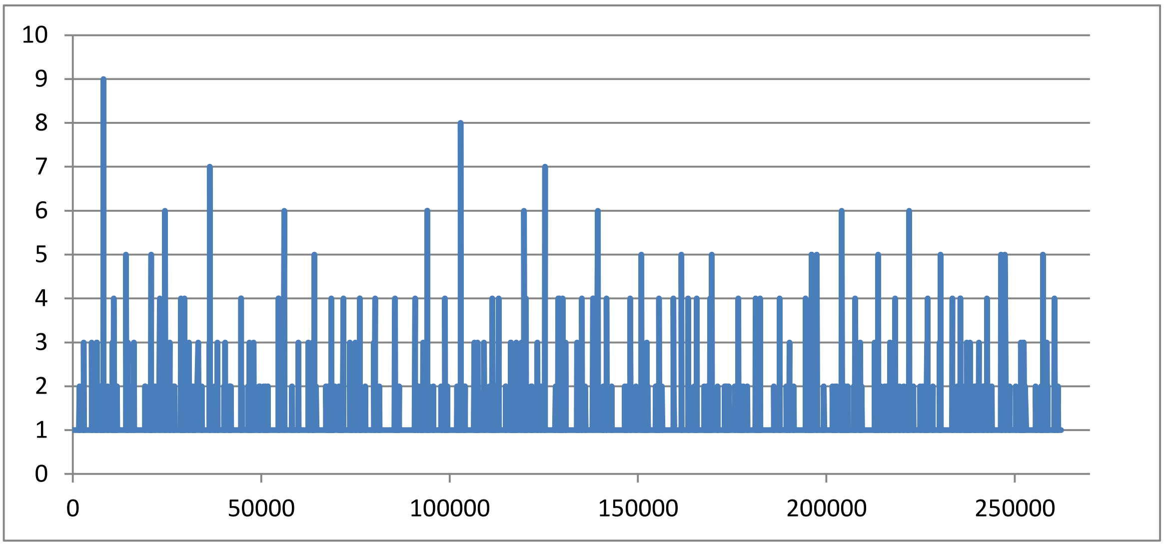


Fig.20 Manpower progression

Another simulation was done this time setting the number of personnel available to 4. It was chosen 4 because this is the manpower used for the event with the highest personnel requirement in the model. Any number inferior to 4 would mean that in the event of a failure of this component there would never be enough people to repair it thus remaining broken for the entire simulation.

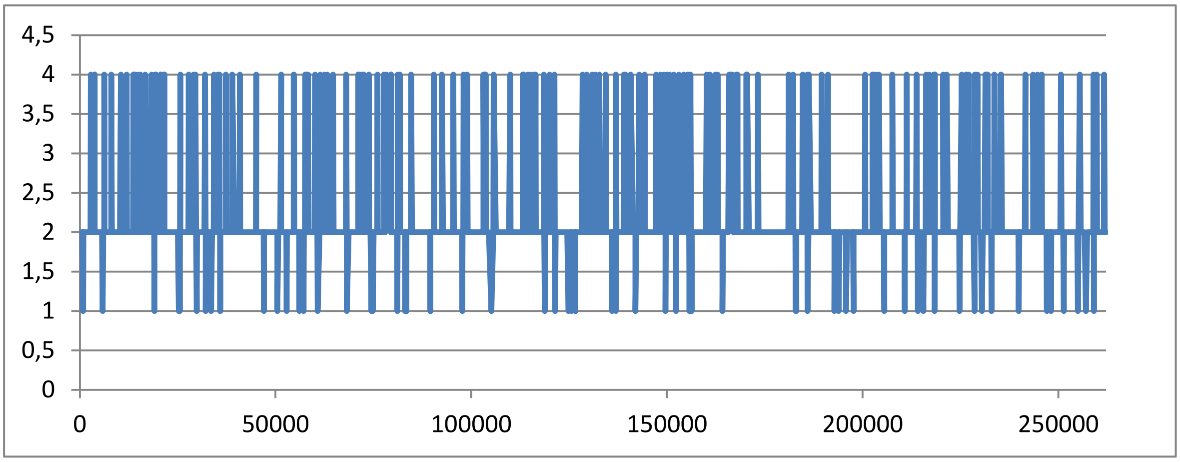


Fig.21 Manpower progression with limited personnel

As can be observed Availsim respects the limitation and never are used more workers than 4. Looking at the availability results we can observe that the manpower limitation has effectively decreased the availability of the accelerator.

Simulation	Hardware availability(facility 1)
Without manpower restriction	91,30
Minimum manpower available	90,80

Table-17 Effect of limited manpower on the hardware availability

Although it hasn't decreased in a great measure is still noticeable and interesting to see. It is important to remember that the scheduled maintenance periods are not affected by manpower restrictions.

9.4.2 Vault access

Availsim counts the number of time the vault has been accessed and how much time did the repairs in the vault last.

facility	hours spent in access	proportion of downtime used in access	times the vault has been accessed	average hour spent per access
1	10939,32	51,13%	126,50	86,48
2	8328,96	44,63%	119,24	69,85

Table-18 Vault accesses

It can be observed that even though the facilities' vaults have been accessed a similar number of times, the repairs of facility 2 required 24% more time. Both facilities have used almost half or more than half of their downtime performing repairs in the vault. That means the vault will be accessed not only frequently but during long periods.

10 Software validation

In order to offer credibility on the results obtained from software the software itself must be tested first. Due to the amount of changes introduced to Availsim a new benchmarking test is needed. Due to the unique nature of Availsim is not possible to compare its entire features with other software. For this reason only the basic features were tested by these means. The program used to compare the results was RiskSpectrum®. It was chosen because it is the professional software used by the RAMI team.

10.1 Basic simulation test

The first stage of the benchmarking process consisted in performing an availability analysis of a system model using RiskSpectrum® and Availsim. The reason one system is chosen instead the whole accelerator facility is dues to the differences previously explained in maintenance, degraded operation and failure management between the software tools. For this reason the following test is devised to check if the basic core of the simulation works properly after being altered in order to introduce the new features.

The chosen system analyze was the water cooling system belonging to the RFQ. The Availsim simulation was performed under the following parameters:

- The operation time of the simulation was set to 30 years in order to provide enough room to allow the most reliable components to fail.
- No degraded operation is allowed. Every event provokes the shutdown of the system and must be repaired.
- Two scheduled maintenance periods programmed per year. A short one scheduled at 6 moths with duration of 3 days. A long one scheduled at the end of the year with duration of 20 days.
- No possibility of using nonscheduled down time to perform repairs or maintenance programmed for scheduled maintenance periods.
- The 0.2 factor to extend down times in order to reflect events that are ignored during that down time is canceled. The extension factor is used on systems with a large number of events (approximately 17300 events in the case of IFMIF). However the water cooling system contains only 33 events and the probability of an event happening when another one is being repaired is extremely low.

– The simulation is composed of 1000 iterations.

The results of the Availsim simulation and the RiskSpectrum® analysis are the following.

Software	Mean (%)	5% (%)	95% (%)
RiskSpectrum®	998,26E-01	999,71E-01	994,97E-01
Availsim	998,27E-01	998,29E-01	998,26E-01

Table-19 Availsim and RiskSpectrum® results comparison

The difference between the two means is 1,40E-3 %. The entire range of the Availsim availability results fits into the error range of the RiskSpectrum® results. Once the Availability results have been checked the event unavailability output must me checked. The following table displays the events of the system sorted in a descending unavailability contribution order for both Availsim and RiskSpectrum®.

Availsim	**RiskSpectrum®**
1RGWWSKG	1RGWWSKG
1RGWCWRG	1RGWCWRG
1RGWCFMG	1RGWCFMG
1RGWWMVG	1RGWWMVG
1RGWWW3G	1RGWWW3G
1RGWWTWG	1RGWWTWG
1RGWWHXG	1RGWWPWG
1RGWWPWG	1RGWWHXG
1RGWWRBG	1RGWWRBG
1RGWWVPC	1RGWWVPC
1RGWCTHG	1RGWCTHG

Table-20 Availsim and RiskSpectrum® events comparison

As can be observed the order is the same for both Availsim and RiskSpectrum® except in one case. In Availsim the events 1RGWWHXG appear before 1RGWWPWG. In RiskSpectrum® it is the opposite case. The reason is from 1RGWWHXG to the end none of the events causes any amount of down hours in Availsim. This is due to RiskSpectrum® dealing with probabilities while Availsim dealing with real down hours. If an event has an extremely low chance to fail, it will be reflected on RiskSpectrum®. But Availsim may never see that component fail, and for that reason it does not provoke any down hours.

10.2 Enhanced features verification

The very reason of this project was to adapt an already unique software and enhance some of its features. For this reason in order to test the new features introduced there was none other software to compare to. For this reason the verification that Availsim performed correctly was to be done by other means.

This was done thought the log file of Availsim called history [appendix 1.3.6]. The log contains every action performed during one iteration. So by checking the events and observing how the software responds to them the performance of the simulation can be evaluated. This procedure has limitations though. Firstly the log can only be obtained from one iteration at a time. Meaning that if there is a bug and does not appear in that iteration it will not be detected. Another evident downfall of this methodology is that it takes time because is done by visual observation.

10.3 Comparison of IFMIF availability results

It is interesting to compare the results of the Availasim simulation with the results obtained from the RiskSpectrum® calculation using the same set of input data[12].

Software	Hardware availability	Beam effectiveness	Beam availability
AvailSim	91,86%	94.68%	84,95%
RiskSpectrum®	91.57%	88.73%	81.25%
Requirement	91.10%	95.55%	87.00%

Table-20 Availsim and RiskSpectrum® beam availability comparison

As can be observed the Beam availability for Availsim is significantly superior than the one obtained from RiskSpectrum®. It is a logical outcome taking into consideration that Availsim allows maintenance management, multiple simultaneous repairs and degraded operation which are aimed to improve the availability and beam effectiveness.

11 Conclusions

Once the results have been studied and verified a series of conclusion can be obtained from the adapted Availsim based on its utilization and response from the IFMIF RAMI team:

11.1 Conclusions on the adaptation of Availsim

Availsim has not been only adapted to accept the IFMIF model. Some features have been added in order to bring the simulation closer to the IFMIF operation.

11.1.1 Multiple iterations

The addition to perform multiple iterations in each simulation allows the user to extract more realistic data. This is a vital for a Montecarlo simulation. Enough iterations have to be performed in order to obtain a reliable result within a desired confidence interval and error.

11.1.2 Inclusion of Functions

The inclusion of functions instead of parameters allows observing the effect on the accelerator's availability for every failure mode of a component instead of only the most common one. This can be useful for designers because they can focus on that failure mode instead of trying to improve the whole component's reliability.

Functions also allow Availsim to add more complexity to the models. Initially every component would have an effect on the accelerator availability. The functions allow introducing components whose failure does not have any effect on the availability until a minimum of operating components is reached. Redundancies can easily be modeled with functions.

Output parameters like the intensity can be flagged as special functions. This means Availsim will calculate if it is more profitable to stop and repair the failures to bring it to is maximum value or instead is more profitable to wait for the next scheduled maintenance period. Thanks to this feature the beam effectiveness can be maximized. Although it will mean adding some degree of hardware unavailability in the end the beam availability will improve and that is the final goal as was explained in chapter [9.3].

11.1.3 Simulation efficiency

On the downside due to the new elements this modification has introduced (events instead of components and functions instead of parameters) the processing time of the simulation has been increased exponentially. The number of events to simulate has been increased from hundreds to tens of thousands. This huge increase in events takes a heavy toll on the efficiency. The original Availsim could take a few minutes to perform a simulation.

It has to be taken into consideration that originally Availsim performed a single iteration of 1 year of operation. The adaptation of Availsim performs multiple iterations and accepts higher simulation times (in this project 30 years were used as simulation time).

For that reason simulation take several hours and even days depending on the number of iterations. A backup routine was introduced in order to save all the results after every iteration to prevent loosing data if the simulation was stopped accidentally.

11.2 Conclusions on Availsim use by the RAMI team

Availsim has been used by the RAMI team and its results have been studied and added to their reports. This has been due the features that Availsim had that could not be found in other RAMI software.

11.2.1 Maintenance

Availsim allows introducing more realistic approaches to scheduled maintenance periods trying to always minimize the downtime. This brings the simulation one step closer to reality which is always desirable. The user can change the frequency and duration of maintenance periods in order to improve the overall availability. This feature offers the user the possibility to experiment with different maintenance strategies in order to find the optimal one. Also Scheduled maintenance can be executed during nonscheduled down times. This method allow to save down time.

11.2.2 Manpower

The manpower restriction introduces the possibility of studying the optimal personnel required for the correct operation of the facility. It can also be observed how manpower restriction affects the availability. However Availsim does not make distinction on the specialization of the workers and shifts.

11.2.3 Results

The results obtained with Availsim are closer to the IFMIF availability requirements than RiskSpectrum®[12]. The Beam availabily calculated with Availsim is 84,95% while the availability obtained with RiskSpectrum® is 81,25%. The availability requirement of 87% has not been reached though.

As explained before RiskSpectrum® is a probability calculation while Availsim is an availability simulation. The great advantage of a simulation is that can be programmed to follow the same protocols the operation team would perform to a certain degree.

11.2.4 Input data

A downside of Availsim is the inputs require significant more data than RiskSpectrum®. This provokes a larger time to create them and introduces more probability to make mistakes introducing the data.

11.3 Final conclusion

The results obtained with Availsim were considered valid for the RAMI Team and allowed them to make a comparison with the ones obtained with RiskSpectrum®. Both softwares provide different views on the same goal, the beam availability. Its aim is to provide the RAMI team with the possibility of introducing new variables to the accelerator model and observe the effect on tis availability. The new data provided can be useful in pointing the design team into directions that previously hadn't been observed.

This version of Availsim is not mean to be considered in any case superior to the original Availsim software. While the original was created in order to compare different accelerator distribution through the availability this version aim is to simulate one model trying to maximize its beam availability. For this reason this version is not an improved Availsim software but only one version of it.

12 Future work

Availsim is a simple software but with great potential. Being Availsim open source software the only limitation to its development is imagination and resources. These are some ideas of how Availsim could be developed to:

- Improve Availsim in order to be able to simulate all the facilities in a system like IFMIF allowing interaction between facilities.
- Introducing a friendly user interface. Although the original availsim had its own interface based on excel macros it was discarded in this version because was considered to be too much specific for the ILC accelerator and similar[9].
- Improving the performance of the simulation in order to reduce the time spent.
- Adapting it in order to be performed within a cluster would increase drastically the performance.
- Adding more complexity to the manpower restrictions such as shifts or the specializations of the technicians.
- It would be very interesting to introduce variability in the repair time. Thanks to the efforts of the RAMI team there is complete data concerning the reliability of the remote handling systems and human error on the repairs performed on IFMIF. Adapting IFMIF to accept this data would bring it one step closer to reality thus improving the quality of the simulation[13].

This version of Availsim will be uploaded to internet and be available for anyone to download and make modifications.

13 Acknowledgements

This project could not have been possible without the help of Enric Bargalló whose constant support, help and patience allowed me to bring this project to term.

I want to thank also the directors of the project Javier Dies and Carlos Tapia for offering me the possibility to participate on a project of this magnitude. Also I want to thank the entire FEEL department for their fantastic work that provided me with the data needed for the analysis.

Of course I'm very thankful of Tom Himel, creator of Availsim. His extensive annotations in the code allowed for an easy understanding of the software.

Finally I want to thank my family and friends for their support and patience.

FEEL
FUSION ENERGY ENGINEERING LABORATORY

ETSEIB

14 Bibliography

[1] R. Brucker, P. Fernández, C. Tapia, J. Dies, J. Abal, and J. M. Arroyo, "IFMIF RAMI Guidelines," 2009.

[2] "IFMIF Conceptual Design Activity (CDA)," 1996.

[3] and J. D. E. Bargalló, J. Abal, J. M. Arroyo, C. Tapia, "Accelerator Facility RAMI report DDD-II," 2012.

[4] J. M. Arroyo, "Maintenance strategy for IFMIF,Internal Report," 2012.

[5] G.Martínez, "Contribution to RAMI assessment of the IFMIF injector," 2012.

[6] Internal Report, "Design Definition Document I of the IFMIF Test Cell, Access Cell, and Test Module Handling Cell," 2011.

[7] A. Mosnier, "IFMIF-EVEDA Status of the Project," 2010.

[8] R. Brucker, P. Fernández, C. Tapia, J. Dies, J. Abal, and J. M. Arroyo, "IFMIF RAMI Guidelines," 2009.

[9] T. Himel, J. Nelson, N. Phinney, M. Park, and M. Ross, "Availability and reliability issues for ILC," *IEEE Particle Accelerator Conference (PAC)*, 2007.

[10] P. A. P. Bargalló, E., NGHIEM, "Accelerator facility failure acceptance and beam degradation," 2012.

[11] E. Baudais, K. Breit, T. Canty, A. Custer, R. Dassen, J. Goldberg, A. J. Guelzow, J. K. Hellan, M. de Icaza, J.-P. Iivonen, A. Kirillov, S. Klost, G. Leblanc, L. Luangkesorn, T. Miesbauer, W. Schuller, A. S. Tigelaar, C. Twardy, A. Weber, and M. Welinder, *The Gnumeric Manual, version 1.12*. 2012.

[12] E. Bargalló, J. Abal, J. M. Arroyo, P. J. Sureda, C. Tapia, and J. Dies, "Accelerator F RAMI report DDD-III," 2013.

[13] E. Baeza, "Human reliability assessment in remote handling tasks as a contribution to RAMI reviews of IFMIF," 2013.