

Institutionen för systemteknik

Department of Electrical Engineering

Examensarbete

Model-based Diagnosis of a Satellite Electrical Power System with RODON

Examensarbete utfört i Fordonssystem
vid Tekniska högskolan i Linköping
av

Olle Isaksson

LITH-ISY-EX--09/4236--SE

Linköping 2009



Linköpings universitet
TEKNISKA HÖGSKOLAN

Model-based Diagnosis of a Satellite Electrical Power System with RODON

Examensarbete utfört i Fordonssystem
vid Tekniska högskolan i Linköping
av

Olle Isaksson

LITH-ISY-EX--09/4236--SE

Handledare: **Emil Larsson Ph.D. Student**
ISY, Linköpings universitet

Peter Bunus Ph.D.
Uptime Solutions AB

Examinator: **Associate Professor Erik Frisk**
ISY, Linköpings universitet

Linköping, 15 February, 2009

	Avdelning, Institution Division, Department Division of Vehicular Systems Department of Electrical Engineering Linköpings universitet SE-581 83 Linköping, Sweden	Datum Date 2009-02-15
Språk Language <input type="checkbox"/> Svenska/Swedish <input checked="" type="checkbox"/> Engelska/English <input type="checkbox"/> _____	Rapporttyp Report category <input type="checkbox"/> Licentiatavhandling <input checked="" type="checkbox"/> Examensarbete <input type="checkbox"/> C-uppsats <input type="checkbox"/> D-uppsats <input type="checkbox"/> Övrig rapport <input type="checkbox"/> _____	ISBN _____ ISRN LITH-ISY-EX--09/4236--SE Serietitel och serienummer ISSN Title of series, numbering _____
URL för elektronisk version http://www.vehicular.isy.liu.se http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-16763		
Titel Diagnosis av en satellits elsystem Title Model-based Diagnosis of a Satellite Electrical Power System with RODON Författare Olle Isaksson Author		
Sammanfattning Abstract <p>As space exploration vehicles travel deeper into space, their distance to earth increases. The increased communication delays and ground personnel costs motivate a migration of the vehicle health management into space. A way to achieve this is to use a diagnosis system. A diagnosis system uses sensor readings to automatically detect faults and possibly locate the cause of it. The diagnosis system used in this thesis is a model-based reasoning tool called RODON developed by Uptime Solutions AB. RODON uses information of both nominal and faulty behavior of the target system mathematically formulated in a model.</p> <p>The advanced diagnostics and prognostics testbed (ADAPT) developed at the NASA Ames Research Center provides a stepping stone between pure research and deployment of diagnosis and prognosis systems in aerospace systems. The hardware of the testbed is an electrical power system (EPS) that represents the EPS of a space exploration vehicle. ADAPT consists of a controlled and monitored environment where faults can be injected into a system in a controlled manner and the performance of the diagnosis system carefully monitored. The main goal of the thesis project was to build a model of the ADAPT EPS that was used to diagnose the testbed and to generate decision trees (or trouble-shooting trees).</p> <p>The results from the diagnostic analysis were good and all injected faults that affected the actual function of the EPS were detected. All sensor faults were detected except faults in temperature sensors. A less detailed model would have isolated the correct faulty component(s) in the experiments. However, the goal was to create a detailed model that can detect more than the faults currently injected into ADAPT. The created model is stationary but a dynamic model would have been able to detect faults in temperature sensors.</p> <p>Based on the presented results, RODON is very well suited for stationary analysis of large systems with a mixture of continuous and discrete signals. It is possible to get very good results using RODON but in turn it requires an equally good model. A full analysis of the dynamic capabilities of RODON was never conducted in the thesis which is why no conclusions can be drawn for that case.</p>		
Nyckelord Keywords Model-based diagnosis, satellite electrical power system, RODON, conflict-directed search, advanced diagnostics and prognostics testbed, ADAPT		

Abstract

As space exploration vehicles travel deeper into space, their distance to earth increases. The increased communication delays and ground personnel costs motivate a migration of the vehicle health management into space. A way to achieve this is to use a diagnosis system. A diagnosis system uses sensor readings to automatically detect faults and possibly locate the cause of it. The diagnosis system used in this thesis is a model-based reasoning tool called RODON developed by Uptime Solutions AB. RODON uses information of both nominal and faulty behavior of the target system mathematically formulated in a model.

The advanced diagnostics and prognostics testbed (ADAPT) developed at the NASA Ames Research Center provides a stepping stone between pure research and deployment of diagnosis and prognosis systems in aerospace systems. The hardware of the testbed is an electrical power system (EPS) that represents the EPS of a space exploration vehicle. ADAPT consists of a controlled and monitored environment where faults can be injected into a system in a controlled manner and the performance of the diagnosis system carefully monitored. The main goal of the thesis project was to build a model of the ADAPT EPS that was used to diagnose the testbed and to generate decision trees (or trouble-shooting trees).

The results from the diagnostic analysis were good and all injected faults that affected the actual function of the EPS were detected. All sensor faults were detected except faults in temperature sensors. A less detailed model would have isolated the correct faulty component(s) in the experiments. However, the goal was to create a detailed model that can detect more than the faults currently injected into ADAPT. The created model is stationary but a dynamic model would have been able to detect faults in temperature sensors.

Based on the presented results, RODON is very well suited for stationary analysis of large systems with a mixture of continuous and discrete signals. It is possible to get very good results using RODON but in turn it requires an equally good model. A full analysis of the dynamic capabilities of RODON was never conducted in the thesis which is why no conclusions can be drawn for that case.

Acknowledgments

I would like to thank my industrial supervisor Peter Bunus for supporting me and giving me the great opportunity to work with a NASA related project. I would also like to thank my university supervisor Emil Larsson and examiner Erik Frisk for their support and the interesting discussions. I would like to thank the people at Uptime Solutions: application engineers Beate Frey and Burkhard Munker who always took time to answer my questions about modeling with RODON and developers Henrik Johansson and Martin Schmid who answered my questions about RODON. I would also like to thank my family and fiance for withstanding my endless talk of satellites and diagnosis.

Contents

1	Introduction to the thesis	1
1.1	Problem description	1
1.2	Objectives	1
1.3	Limitations	2
1.4	Existing work	2
1.5	Contributions	2
2	Theory of model-based diagnosis and RODON	3
2.1	Diagnosis	3
2.1.1	Model-based diagnosis	3
2.1.2	Conflict-directed search	4
2.2	RODON	6
2.2.1	The composer	7
2.2.2	The analyzer	11
3	The NASA Advanced Diagnostics and Prognostics Testbed (ADAPT)	15
3.1	Objectives	15
3.2	Concept of operations	16
3.3	Functional description	16
3.4	Systems description	17
3.4.1	Power generation unit	18
3.4.2	Power storage unit	29
3.4.3	Power distribution unit	32
3.4.4	Control and monitoring	33
3.5	The Advanced Caution And Warning System (ACAWS) scenarios	35
3.5.1	Scenario loads	35
3.5.2	Load monitoring	36
3.5.3	Scenario descriptions	36
4	Model of the Advanced Diagnostics and Prognostics Testbed (ADAPT)	51
4.1	Physical models of the components	51
4.1.1	Wire	51
4.1.2	Resistor	52
4.1.3	Relays	52
4.1.4	Circuit breaker	53

4.1.5	Battery charger	53
4.1.6	Charge controller	53
4.1.7	Sensors	53
4.1.8	Accumulator	55
4.1.9	Solar panel	58
4.1.10	Inverter	71
4.1.11	Loads	71
4.2	RODON implementation	72
4.2.1	NASA Library	72
4.2.2	Model structure	74
4.2.3	Model-based diagnosis of the testbed	74
4.2.4	Generation of decision trees	77
5	Results and discussion	81
5.1	Model-based diagnosis	81
5.1.1	Discussion	81
5.2	Decision trees	85
5.2.1	Scenario 1a	85
5.2.2	Scenario 1b	87
5.2.3	Scenario 1c fault 1	87
5.2.4	Scenario 1c fault 2	88
5.2.5	Scenario 1d fault 1	88
5.2.6	Scenario 1d fault 2	88
5.2.7	Scenario 2a	88
5.2.8	Scenario 2b	91
5.2.9	Scenario 3a	91
5.2.10	Scenario 3b	91
5.2.11	Scenarios 4a, 5b fault 1 and 6a fault 1	91
5.2.12	Scenarios 4b and 5a fault 2	91
5.2.13	Scenario 6a fault 2	92
5.2.14	Scenario 6b fault 1	92
5.2.15	Scenario 6b fault 2	93
5.2.16	Discussion	93
5.3	Interactive model-based diagnosis in RODON	93
5.4	Related work	96
5.4.1	Testability Engineering and Maintenance System - Real Time (TEAMS-RT)	96
5.4.2	Hybrid diagnostic engine (HyDE)	98
5.4.3	Fault Adaptive Control Technology (FACT)	100
5.4.4	ADAPT bayesian networks (BN) Model	104
5.5	Conclusions	107
5.6	Future work	107
	Bibliography	109
	A Abbreviations	113

Chapter 1

Introduction to the thesis

1.1 Problem description

As space exploration vehicles travel deeper into space, their distance to earth increases. The increased communication delays motivate a migration of the vehicle health management into space. In addition to faster reaction times there is also a financial aspect. An automation of the vehicle health management would decrease ground personnel costs.

The advanced diagnostics and prognostics testbed (ADAPT) developed at the NASA Ames Research Center provides a stepping stone between pure research and deployment in aerospace systems. ADAPT is developed to test, evaluate and mature diagnostic and prognostic systems. It also provides a standardized platform where different vehicle health management systems can be compared. The hardware of the testbed is an electrical power system (EPS) that represents the EPS of a space exploration vehicle. The main goal of the thesis project is to build a model of the ADAPT EPS that should be used for evaluation of various diagnostics analysis.

1.2 Objectives

The main objective of the thesis is to build a complete model of the ADAPT EPS. The model will be built in a model-based reasoning tool called RODON. RODON is a commercial, industry proven tool developed by Uptime Solutions AB. Once the model is created, it will be used to fulfill the objectives listed below:

- Perform model-based diagnosis on ADAPT.
- Create decision trees (or diagnostic trouble-shooting trees) based on various fault symptoms.
- Compare the results from RODON with other diagnostic methods.

1.3 Limitations

The project described in this thesis has the following limitations:

- The model was created from a system description and sampled data from July 2007.
- The stationary case has been modeled and a dynamic model has not been implemented within the frame of the thesis project.
- An interface between the ADAPT API and RODON has not been developed.

The missing functionality described above is intended to be addressed by future work.

1.4 Existing work

NASA provides the satellite electrical power system, documentation of it and sampled experiment data from a set of predefined simulation and diagnostics scenarios performed on the testbed. The interested user can find the data at the website: <http://dx-competition.org/>.

The decision trees will be created based on the symptoms from these scenarios. RODON provides a model-based reasoning tool capable of performing model-based diagnosis and automatically generate decision trees from a model. RODON can compute top events based on root causes or find the root cause based on a top event. No changes or contributions had to be made in the RODON software.

1.5 Contributions

A detailed model of a satellite electrical power system has been created comprised of 884 components with both nominal, faulty, continuous and discrete behavior. The temperature effects on a satellite electrical power system have been investigated and are included in the model. A library called "NASA" has been created in RODON. The library components are fully reusable. Diagnostic trouble shooting trees for a satellite electrical power system has been created.

Chapter 2

Theory of model-based diagnosis and RODON

This chapter introduces a few basic concepts about diagnosis, model-based diagnosis and how model-based diagnosis can be done with RODON.

2.1 Diagnosis

The term diagnosis is often associated with medical science. A doctor checks a patient's symptoms and formulates a diagnosis. Diagnosis can also be done on a technical system, i.e. a car or a satellite electrical power system. In this case, the technical system is the patient and the engineer/algorithm is the "doctor". The diagnosis problem is to detect a fault in a system and to locate the cause of it [16]. Fault detection can be done in several ways. A common method is to compare sensor readings with a threshold. If the readings exceed the threshold the system is considered faulty. In safety critical systems it is also common to have redundant functions, i.e. having two sensors measuring the same quantity. The system becomes more robust and a sensor failure can be separated from a failure of the monitored system. A more traditional diagnosis method is to use a set of diagnostic rules created from experience. These diagnostic rules could look something like this: "If the lamp is not lit when the switch is on, the switch is stuck open or the lamp is defect or both are defect". One advantage of the rule-based diagnostic method is that it is very efficient with respect to memory and computing time [3]. This makes the method suitable for on-board diagnostics where limited resources are available. The diagnostic method used in this thesis is called model-based diagnosis and is explained in the following section.

2.1.1 Model-based diagnosis

The simplest form of model-based diagnosis is to use a model of the nominal system. Observations from the diagnosed system are inserted into the model. If

the observations are inconsistent with the model the system is considered faulty. A basic way to do this is to insert system input u into the model and compare its output \hat{y} with the real system's output y . If the size of the difference r between the model and system output is larger than a threshold J , the system is considered faulty. The difference r is called a residual. Figure 2.1 shows how this could look like.

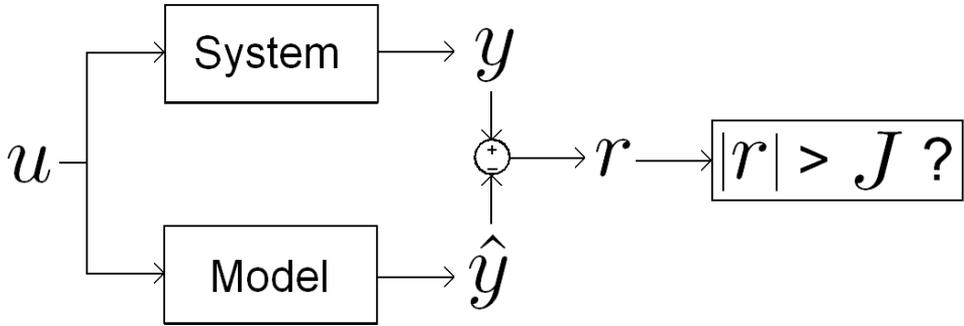


Figure 2.1. A basic way of model-based diagnosis performed on a system. If the size of the residual r is larger than a threshold J , the system is considered to be faulty.

Consider the system depicted in Figure 2.2. It contains two types of components: adder and multiplier. Nominally, the output of the adder is the sum of its two inputs and the output of the multiplier is the product of its two inputs. The inputs to the system are A, B, C, D and E. The outputs from the system are F and G. X, Y and Z are internal variables not known outside the model. If the inputs are A=3, B=2, C=2, D=3, E=3, the outputs should be F=12 and G=12 if the system is working correctly. However, if F=10 is observed from the real system given these input, the nominal behavior modes of the components are inconsistent with the observation. A single-fault explanation is adder A1 failed or multiplier M1 failed depicted in Figure 2.3. If multiplier M2 has failed, incorrect input is sent to adders A1 and A2. G=12 holds if also adder A2 has failed causing it to be computed to 12 incorrectly. The two faulty components in the double fault case is depicted in Figure 2.4.

2.1.2 Conflict-directed search

When information about faulty system behavior is available, it is interesting to include it in the model. A way to do this, in a component-based modeling approach, is to define behavior modes for components in the system. The nominal mode is included in addition to failure modes. The behavior modes for a wire component could be: "okay", "disconnected" and "shorted to ground". A conflict is detected when a set of observations is inconsistent with the current behavior modes in the model, like in the example from the previous section with the observation F=10. Conflict-directed search is a method used to find a set of behavior modes that explains a given set of observations. Conflict-directed search uses conflicts to guide

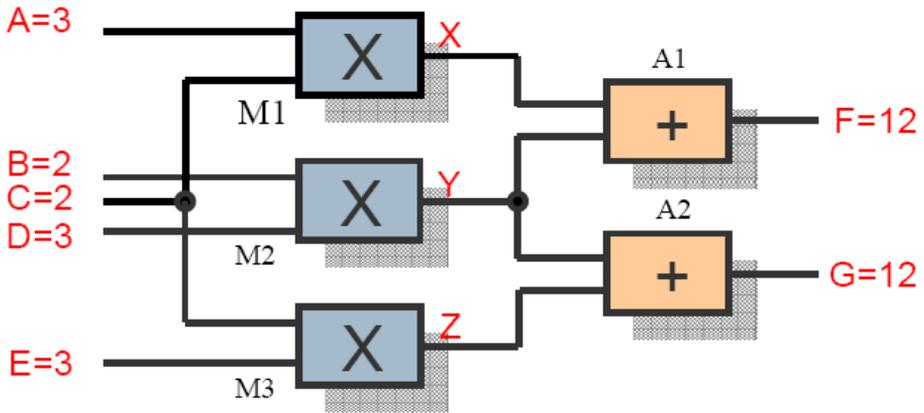


Figure 2.2. A simple system consisting of adders and multipliers. The figure is taken from [3].

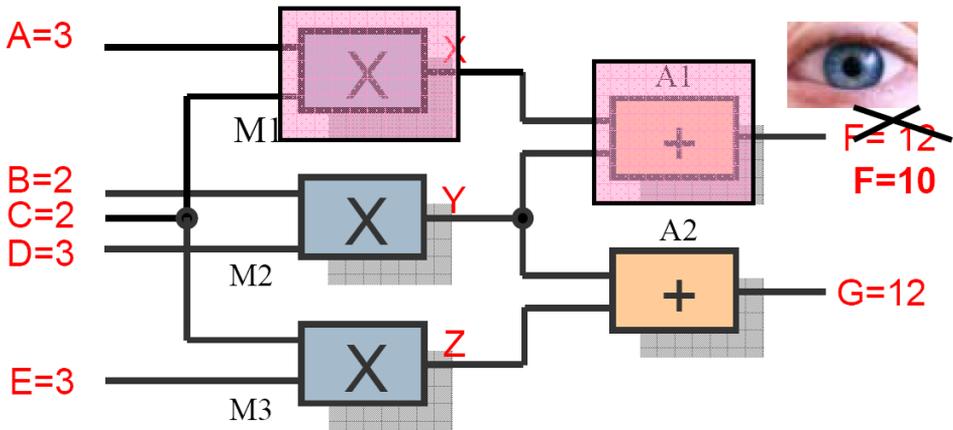


Figure 2.3. A simple system consisting of adders and multipliers. The two single-faults explaining the observation are marked in the figure. The figure is taken from [3].

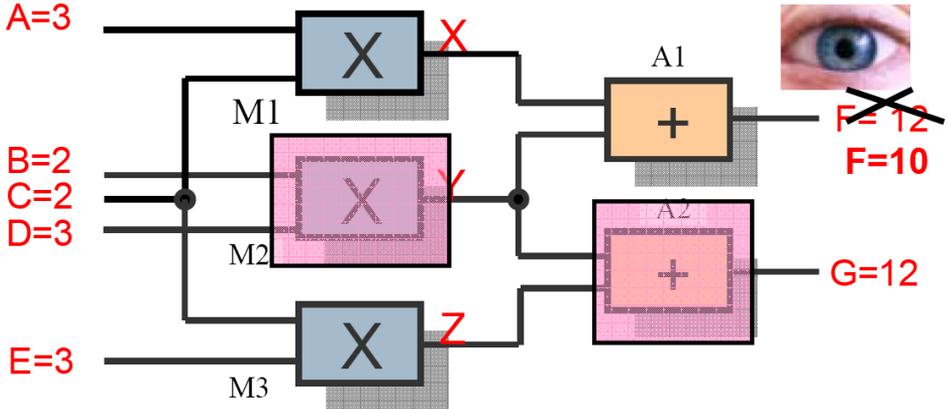


Figure 2.4. A simple system consisting of adders and multipliers. A double-fault explaining the observation is marked in the figure. The figure is taken from [3].

the search among the failure modes.

When conflicts have been detected, the diagnosis system changes the current behavior modes of the components until the conflicts are gone. There can of course be several explanations for the observations and several sets of behavior modes can be created. The generated sets of behavior modes are the candidates of the diagnosis.

2.2 RODON

RODON is a commercial model-based reasoning tool developed by Uptime Solutions AB. It provides an equation-based object oriented language called Rodelica. Rodelica is strongly related to Modelica [1] but it has additional features which makes it suitable for diagnostic problems. A few properties of Rodelica are listed below.

- It supports models with interval data types instead of sharp values.
- It supports bi-directional signals, enabling i.e. sneak currents.
- It supports failure modes and uses conflict-directed search.
- It is object oriented enabling reusable components.
- It is equation-based.

The following sections show how to build a simple model in the RODON composer and then simulate and diagnose it in the RODON analyzer.

2.2.1 The composer

The modeling in RODON is done in the composer environment. Classes are created in libraries and the models in RODON are made up of instances of these classes. The electrical library with the bulb package expanded is depicted in Figure 2.5. Classes can be created by direct coding in Rodelica or by using the graphical user interface (GUI) with functionality such as drag and drop from the library into the model. To get a better view of how to work with RODON, a simple electrical circuit as the one depicted in Figure 2.6 is modeled as an example. The circuit contains a 10 W bulb connected to a 12 V battery and ground through two wires. The wires and the bulb can be disconnected, the other components have no failure modes in this model. The bulb shines bright if enough power is consumed by it, otherwise it shines dimmed or is off if no power is consumed. The Rodelica code for the wire model is depicted in Figure 2.7. The "wirePin" variables p1 and p2 are the component's interface to the outside. If the failure mode variable "fm" is zero (okay), there is no voltage drop across the wire and the current through the interfacing pins p1 and p2 is equal. If the failure mode variable "fm" is one (disconnected), there is no relation between the voltages of the pins p1 and p2 and the current through them is zero. When the bulb, battery and ground classes have been modeled as well, they can be dragged into the top level of the model in the GUI. The top level of the finished model is depicted in Figure 2.6. Once the model has been created, RODON supports several diagnostic methods [3]:

1. Model-Based Diagnosis (MBD), including interactive MBD which means that additional measurements can be provided by the user to narrow down the number of diagnostic candidates.
2. The automatic generation of decision trees (or diagnostic trouble-shooting trees), which can serve as a model documentation or to assist the mechanic in a workshop in a guided diagnosis.
3. The automatic generation of diagnostic rules for on-board diagnostics.

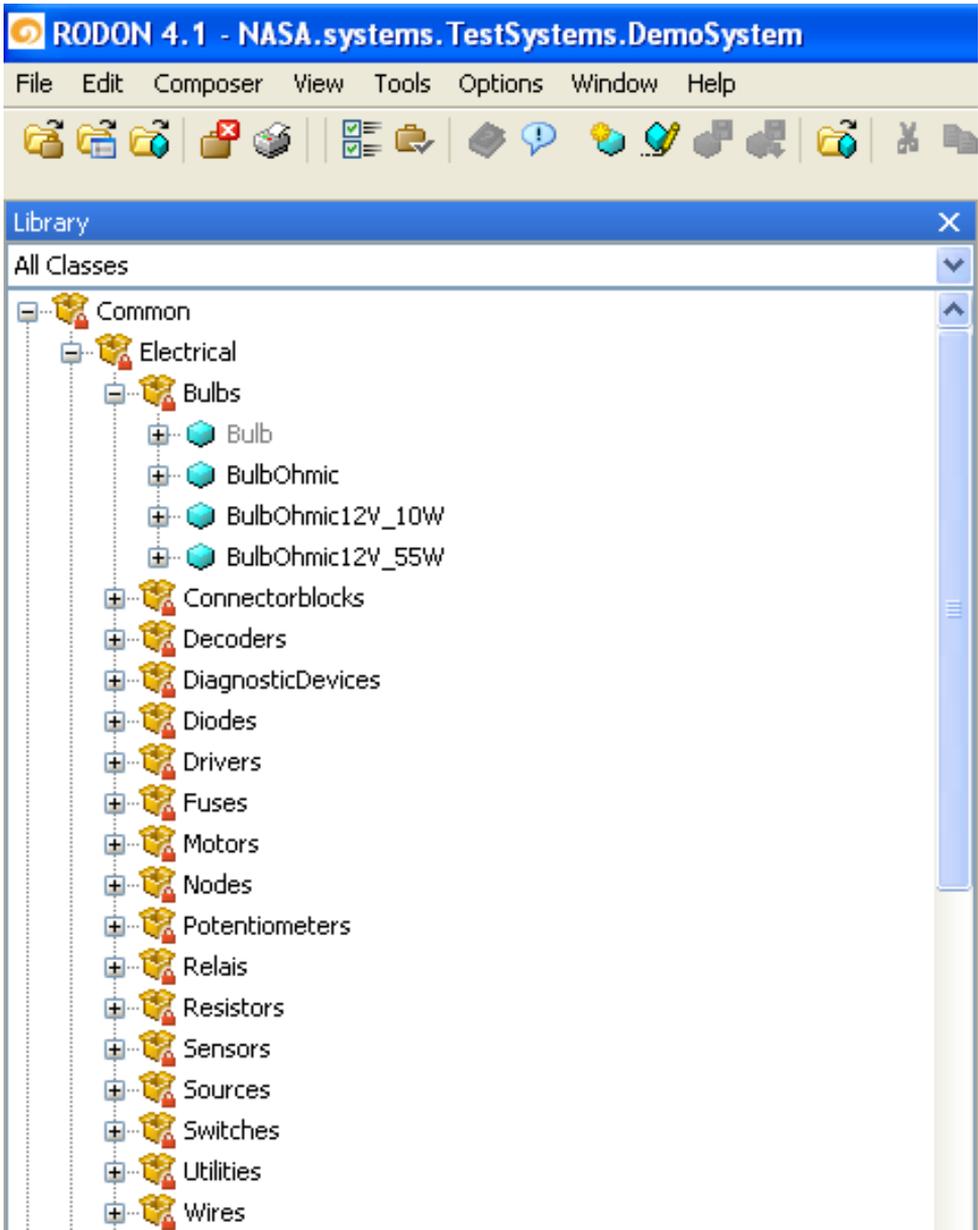


Figure 2.5. The electrical library with the bulb package expanded. A screenshot of the RODON composer.

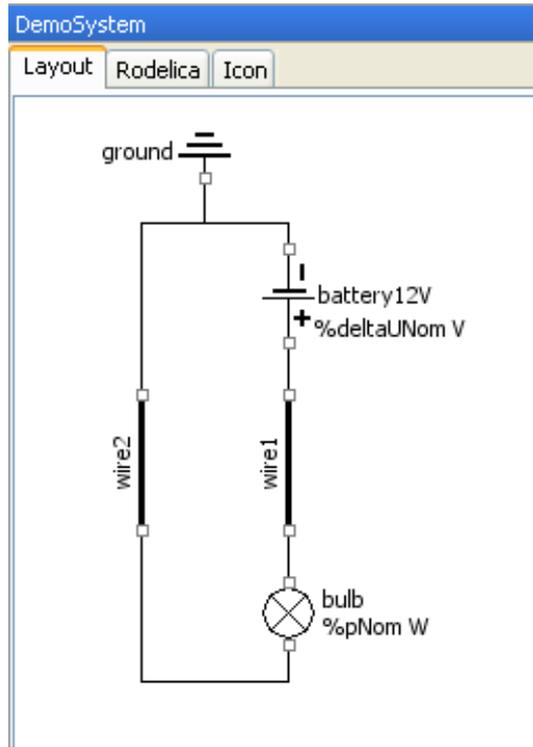


Figure 2.6. The top level of the finished model. A screenshot of the RODON composer.

```

WireBasic
Layout Rodelica Icon
modes can be added in class extensions. <DIP> <DIP>
@author Soerman Information & Media AB */
// =====
model WireBasic
// =====
public
  /** Electrical pin 1 of component. */
  WirePin p1;

  /** Electrical pin 2 of component. */
  WirePin p2;

  /** Failure mode variable. */
  FailureMode fm (max = 1, mapping = "ok, disconnected");

protected
  /** General pin type of wire to be replaced by individual connector
  types, so that wires transferring different quantities basically
  can be described by the same model.<br>
  Default connector type is <quot> Pin <quot>;, transferring voltage
  and current.*/
  replaceable connector WirePin = Pin;

// -----
behavior
// -----

  /** Constraints for health state "okay":
  if (fm == 0)
  {
    /** Current balance:
    p1.i + p2.i = 0;

    /** No voltage drop between pins 1 and 2:
    p1.u - p2.u = 0;
  }

  /** Constraints for failure mode "disconnected":
  if (fm == 1)
  {
    /** There is no current:
    p1.i = 0;
    p2.i = 0;
  }

// =====
end WireBasic;
// =====

```

Figure 2.7. The Rodelica code for the wire model. A screenshot of the RODON composer.

2.2.2 The analyzer

Simulation and diagnosis of models in RODON is done in the analyzer environment. The following section describes how the simple model created earlier can be simulated and diagnosed in the analyzer environment.

Simulation

The created model is loaded into the analyzer and simulated by clicking on the "simulate" button. The bulb shines bright and the model has no conflicts. The result from the simulation is seen in the analyzer view depicted in Figure 2.8. The model can be simulated again, but now with wire2 disconnected. The bulb does not shine this time since there is no current flowing through the circuit. The result from the simulation is seen in the analyzer view depicted in Figure 2.9.

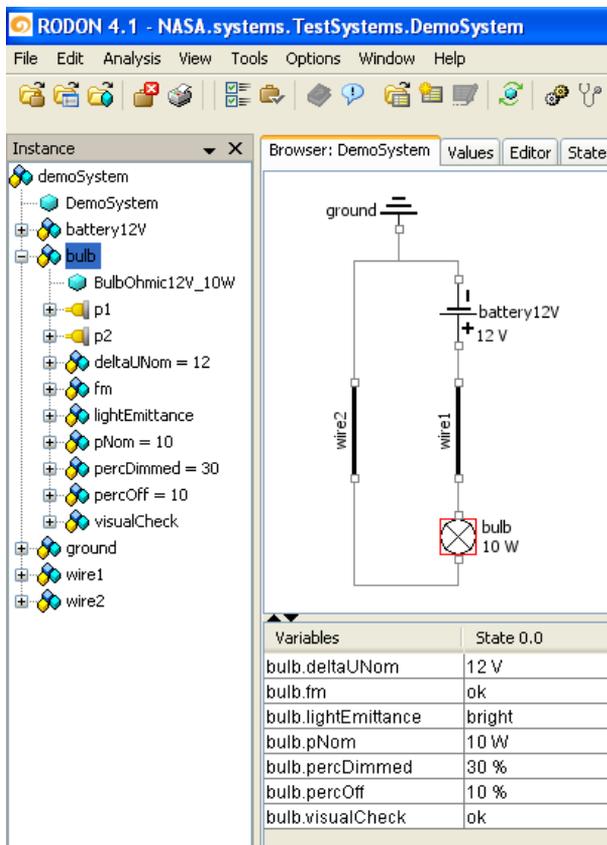


Figure 2.8. Simulation results without observations inserted. The bulb is shining bright (denoted by the "bright" value associated to the bulb.lightemittance variable). A screenshot of the RODON analyzer.

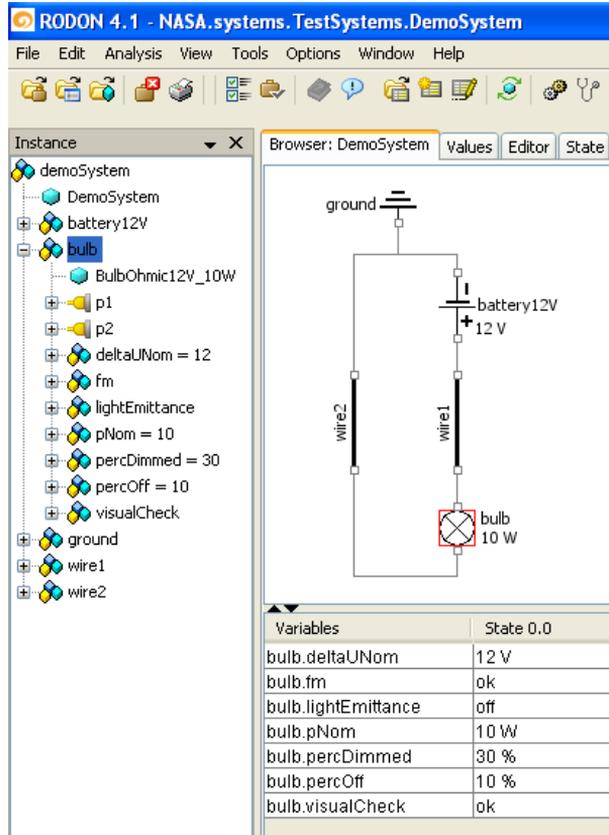


Figure 2.9. Simulation results with wire2 disconnected. No observations are inserted. The bulb is not shining at all (denoted by the "off" value associated to the bulb.lightemittance variable). A screenshot of the RODON analyzer.

Diagnosis

The created model is loaded into the analyzer and an observation "bulb off" is inserted which means the the bulb is not shining at all. The nominal behavior modes of the components can not describe this observation and a conflict is created. The diagnosis engine works with the failure modes and finds combinations of them that are consistent with the observation. In this case there are no faults injected manually and the diagnosis engine generates candidates based on the observations. The generated candidates are depicted in Figure 2.10.

RODON 4.1 - NASA.systems.TestSystems.DemoSystem

File Edit Analysis View Tools Options Window Help

Instance

- demoSystem
 - DemoSystem
 - battery12V
 - bulb
 - BulbOhmic12V_10W
 - p1
 - p2
 - deltaUNom = 12
 - fm
 - lightEmittance = off
 - pNom = 10
 - percDimmed = 30
 - percOff = 10
 - visualCheck
 - ground
 - wire1
 - wire2

Browser: DemoSystem Values Editor Sta

Variables State 0.0

Variables	State 0.0
bulb.deltaUNom	12 V
bulb.fm	disconnected
bulb.lightEmittance	off
bulb.pNom	10 W
bulb.percDimmed	30 %
bulb.percOff	10 %
bulb.visualCheck	defective

Diagnosis:

```
bulb disconnected; 1,
wire1 disconnected; 1,
wire2 disconnected; 1
```

Figure 2.10. Diagnosis results with observation "bulb off" inserted. The observation is explained by one of the wires or the bulb being disconnected. A screenshot of the RODON analyzer.

Chapter 3

The NASA Advanced Diagnostics and Prognostics Testbed (ADAPT)

Assessment and comparison of different vehicle health management technologies can be difficult. To facilitate this task the researchers at NASA Ames Research Center have developed the advanced diagnostics and prognostics testbed (ADAPT). The testbed acts as a common platform where different vehicle health management technologies, so called test articles, can compete against each other on equal conditions. To achieve this, ADAPT consists of a controlled and monitored environment where faults can be injected into a system in a controlled manner and the performance of the test article carefully monitored. The hardware of the testbed is an electrical power system (EPS). The testbed functionally represents the EPS of a space exploration vehicle. The testbed is located in a laboratory at the NASA Ames Research Center.

3.1 Objectives

When automated diagnostic methods are used in aerospace vehicles challenges arise. Some of them are listed below [18].

- Low failure probability of components making it complicated to repeat failures.
- The cost of failures, especially in human crewed vehicles.
- The difficulty to select an appropriate diagnostic technology.
- The cost of verification and validation.
- The lack of large-scale diagnostic technology demonstrations.

To meet these challenges ADAPT was developed with the following goals in mind [18]:

1. "Provide a technology-neutral basis for testing and evaluating diagnostic systems, both software and hardware."
2. "Provide the capability to perform accelerated testing of diagnostic algorithms by manually or algorithmically inserting faults."
3. "Provide a real-world physical system such that issues that might be disregarded in smaller-scale experiments and simulations are exposed - "the devil is in the details"."
4. "Provide a stepping stone between pure research and deployment in aerospace systems, thus create a concrete path to maturing diagnostic technologies."
5. "Develop analytical methods and software architectures in support of the above goals."

3.2 Concept of operations

When diagnosis is performed on the testbed, the focus is on the vehicle health management system performing the diagnosis instead of the testbed which is being diagnosed. The testbed is controlled by a number of relays and monitored by a large set of sensors. Consequently it is possible to detect an injected fault and recover from it if the correct action is taken. To facilitate the execution of the experiments performed with the testbed, three operating roles have been defined [18] : user, antagonist and observer.

The user simulates an actual crew member or pilot who operates and maintains the EPS with the help of a health management application. The antagonist injects faults into the system, either manually by physically acting on the system, or remotely by spoofing sensor values through a computer connected to the system. The malicious actions of the antagonist are not known to the user who is responsible of choosing a suitable recovery action. The observer logs all data in the experiment and monitors how the user responds to the faults injected by the antagonist and therefore measures the effectiveness of the test article. The observer also acts as a safety officer of the experiment and can issue an emergency stop. In Figure 3.1 the layout of the lab is depicted. For more information on the execution of the experiments see the ADAPT operations and safety manual [17].

3.3 Functional description

The testbed functionally represents the electrical power system (EPS) of a space exploration vehicle. The EPS has one simple task: to provide the connected loads with power. The EPS has two different sources of power: light and a connection to the electrical power grid through a wall socket. These sources are then used

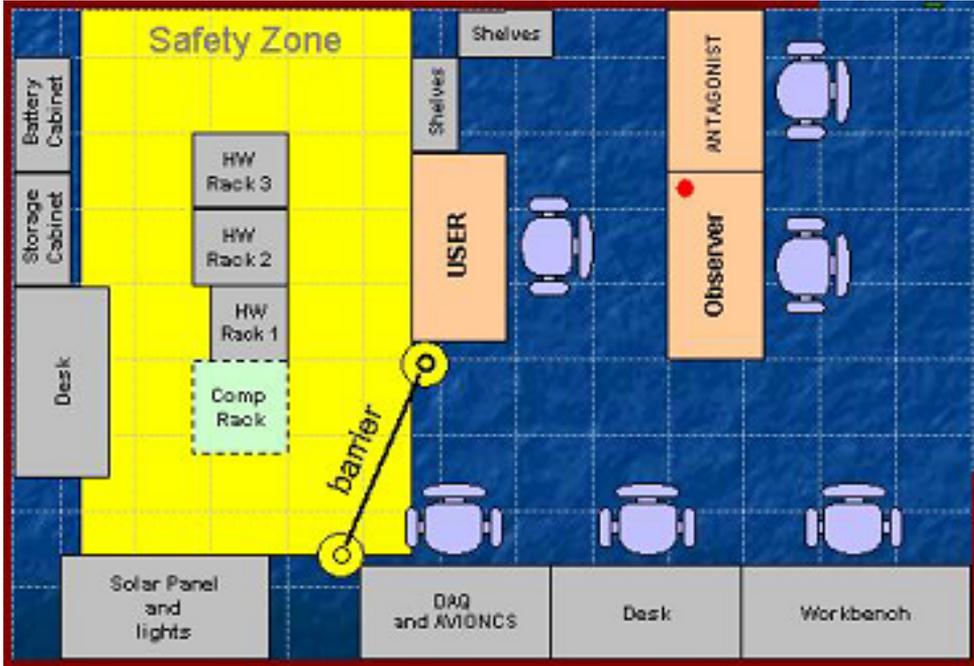


Figure 3.1. ADAPT lab layout. The picture is taken from [17].

by the EPS to store and distribute power to the connected loads which may represent subsystems such as propulsion, life support, thermal management systems, avionics, etc. To achieve this the testbed has been divided into three units: the power generation, storage and distribution units. The power generation unit contains three sources of charging power which can function independently of each other, the power storage unit contains three battery packs and the power distribution unit contains two load banks which provides two sets of loads with both AC and DC power [18]. The system contains several relays which make it possible to charge an arbitrary battery with an arbitrary charging source and connect an arbitrary battery to an arbitrary load bank. This EPS contains a lot of redundant functions which are of utmost importance to the successful outcome of missions in space. An overview of the testbed is depicted in Figure 3.2.

3.4 Systems description

To get a deeper understanding of how the functionality described in Section 3.3 is achieved and to get a deeper understanding of how the top level of the system is affected by its components we need to look deeper into the system. The hardware is located in three equipment racks, a battery cabinet and a solar panel unit. The three equipment racks can be seen in Figure 3.3. The power generation

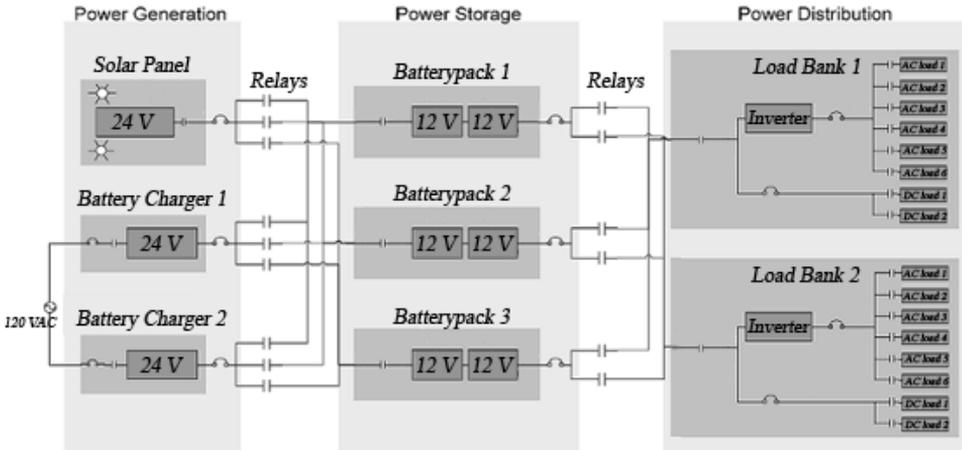


Figure 3.2. Overview of the electrical power system. The picture is taken from [18] and slightly modified.

functionality is located in the first rack and the solar panel unit. Since the solar panel is located indoors without access to sunlight the light energy comes from two controlled 1 kW metal halide lamps. The setup of the two lamps and the solar panel can be seen in Figure 3.4. The power storage functionality is located in the second rack and the battery cabinet, which is depicted in Figure 3.5. Finally, the power distribution functionality is located inside the third rack. All pictures of the hardware is taken from the ADAPT safety and operations manual [17].

3.4.1 Power generation unit

The power generation unit can charge batteries using two sources of energy: light energy and a connection to the power grid through a wall socket. These two types of energy input are then used by three charging sources: two battery chargers connected to the power grid through a wall socket and a solar panel connected to a charge controller that controls the charging current. The power generation unit can be divided into six subsystems: the solar panel unit, the battery charger panel, the protection and enable panel and three battery-charge selection panels. These six subsystems are described in the following sections.

Solar panel unit

The solar panel unit contains a 100 W solar panel, a light transducer and a temperature sensor. The light transducer measures the incoming light to the solar panel and the temperature sensor measure the temperature of the solar panel. The solar panel consists of 72 polycrystalline solar cells where each cell has an area of 120 cm^2 [10] which makes up the total area of 0.864 m^2 . The electrical parameters of the solar panel found in Figure 3.6 are measured at STC (standard



Figure 3.3. ADAPT EPS equipment racks 1, 2, and 3. The picture is taken from [17].

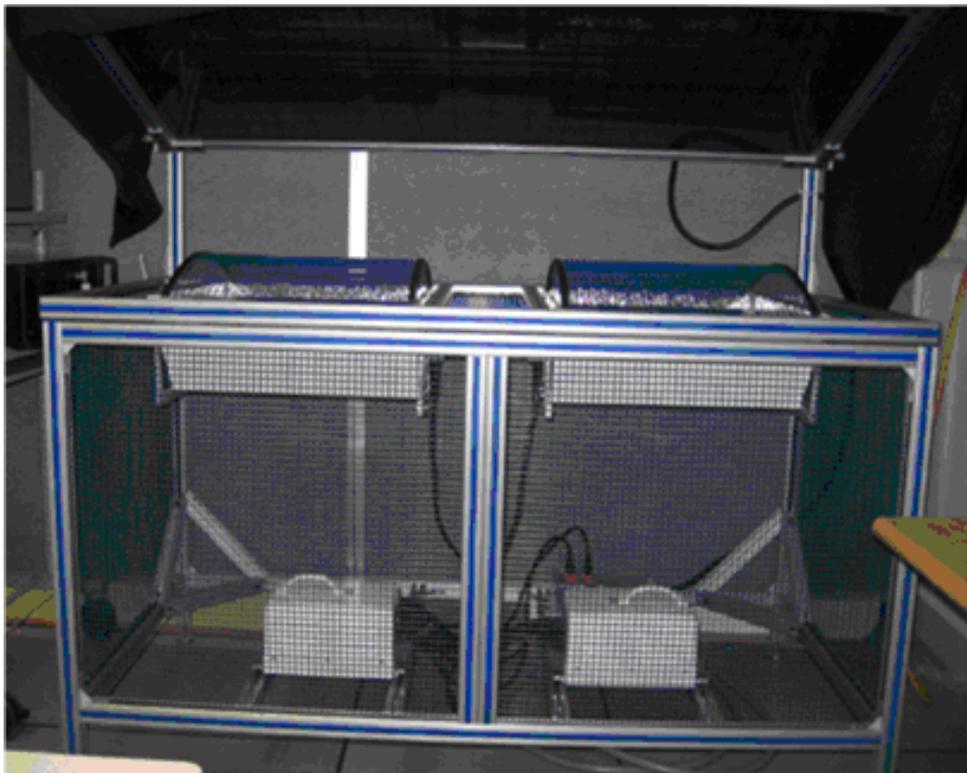


Figure 3.4. The setup of the solar panel unit and the lamps. The solar panel is found in the top and the two lamps just below. The picture is taken from [17].



Figure 3.5. Picture of the battery cabinet with the three battery packs inside taken from [17].

test conditions): 1000 W/m², 25 °C and air mass 1.5 spectrum.

Electrical Ratings

		EC-47	EC-51	EC-55	EC-94		EC-102		EC-110		EC-115	
P _p	W	47	51	55	94		102		110		115	
V _p	V	16.0	16.2	16.5	15.9	31.8	16.2	32.4	16.4	32.7	16.5	33.0
I _p	A	2.94	3.15	3.33	5.91	2.96	6.30	3.15	6.72	3.36	6.97	3.48
V _{oc}	V	20.0	20.0	20.0	20.0	40.0	20.0	40.0	20.0	40.0	20.0	40.0
I _{sc}	A	3.58	3.70	3.79	7.29	3.65	7.49	3.75	7.67	3.84	7.71	3.86
Cells		36	36	36	72		72		72		72	
Max. Series Fuse		15 A										
<small>Note: EC-94, EC-102, EC-110 and EC-115 have field-selectable voltage. Minimum Specified Power Rating is 90% of above power rating. All modules are factory-inspected to produce at least 96% of above power rating</small>												

Figure 3.6. Electrical ratings of the EC100 series. The EC102 solar panel has been modeled. The table is taken from the installation and operation manual [11] for the solar panel.

The EC102 solar panel has been modeled. The open circuit voltage of this model is 40 V which is the voltage output when the connection between the positive and negative terminals of the solar cell is broken. The short circuit current is 3.75 A when the positive and negative terminals are directly connected to each other without any load. The maximum power point is found at a voltage of 32.4 V and a current of 3.15 A [11]. The temperature of the solar panel is monitored which makes it interesting to see how it effects these electrical parameters. The list below shows those effects, specified by the solar panel manufacturer [9]:

- Maximum power: -0.49%/°C.
- Short circuit current: 0.09%/°C.
- Open circuit voltage: -0.41%/°C.

Battery charger panel

The battery charger panel consists of two multistep battery chargers with 24 V, 20 A output. Note that this does not necessary mean that the charging voltage and current is fixed at these values, on the contrary, 24 V battery chargers have a charging voltage that can vary between 24 and up to 40 or more volts. The charging voltage and current depends on which charging stage the charger is operating in. The battery chargers found in this system operates in five different modes: initialization, bulk charge, absorption charge, equalization charge and float charge [4]. A short explanation of the different stages of the charger can be found in the list below and in Figure 3.7. A picture of the battery charger panel can be found in Figure 3.8.

1. Initialization: The monitor circuit verifies appropriate battery voltage levels and good electrical continuity between the battery and the charger DC output.

2. Bulk charge: Charging with constant current at full power. The charger switches to absorption charge at 75% - 80% of full recharge.
3. Absorption charge: Charging with constant voltage at absorption level. This conditions the battery for maximum performance. Adaptive timing transition to equalization.
4. Equalization charge: Charging with constant voltage at equalization level. This minimizes battery cell voltage variation. Adaptive timing transition to float maintenance.
5. Float charge: Charging with constant voltage at float / maintenance level. This keeps the battery fully charged and maintains optimum specific gravity. A charge reset monitor protects the battery against deep discharge from excessive appliance current draw.

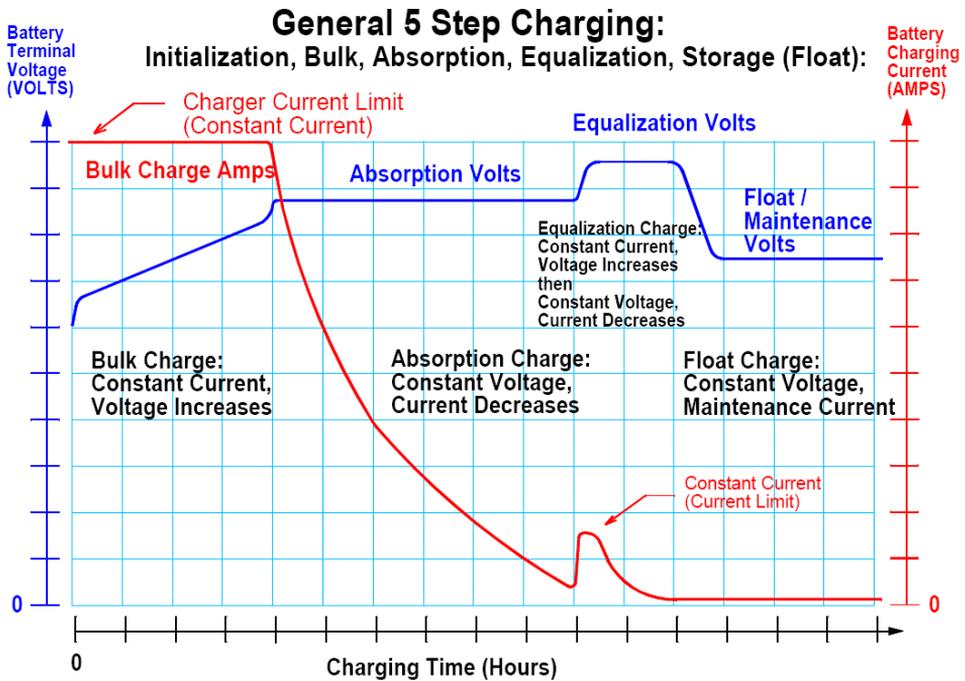


Figure 3.7. Plot showing the charging voltage at the different charging stages of the battery chargers. The picture is taken from [4].

Protection and enable panel

The protection and enable panel protects the power generation unit from dangerously high currents and provides the possibility to enable or disable the chargers.



Figure 3.8. Picture of the battery charger panel containing the two battery chargers taken from [17].

It also measures the voltage before and after the charge controller. To protect the chargers and the batteries from high currents, circuit breakers have been installed before and after the chargers. The on/off control of the chargers are done by relays situated between the AC supply or solar panel and the charger. A picture of the protection and enable panel can be seen in Figure 3.11 and a circuit diagram of the protection and enable panel together with the solar panel unit and the battery charger panel can be found in Figure 3.10. A description of the symbols used in the circuit diagrams of this report is found in Figure 3.9.

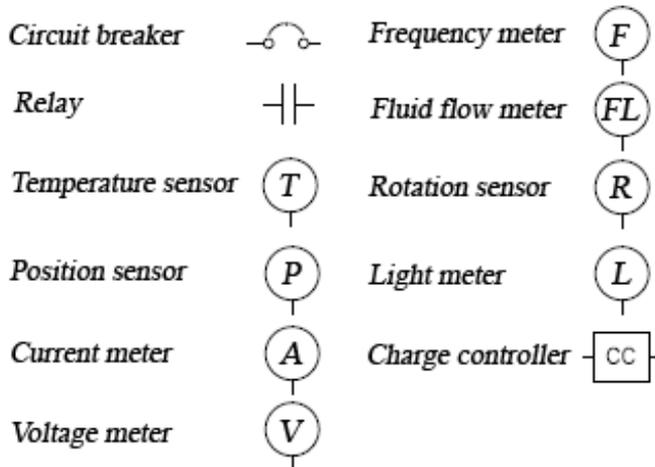


Figure 3.9. Description of the symbols used the in circuit diagrams of this report.

Battery-charge selection panels

The power generation unit contains three battery-charge selection panels, which makes it possible to select which charger that should be connected to which battery and select if a certain battery should be charged at all. Each panel interfaces with one battery, one charger and the other two battery-charge selection panels.

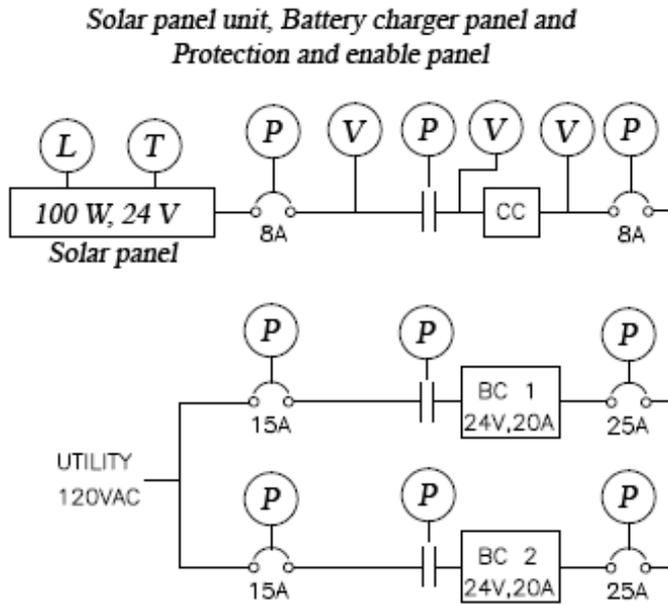


Figure 3.10. Circuit diagram of the protection and enable panel together with the solar panel unit and the battery charger panel. A description of the symbols used is found in Figure 3.9.

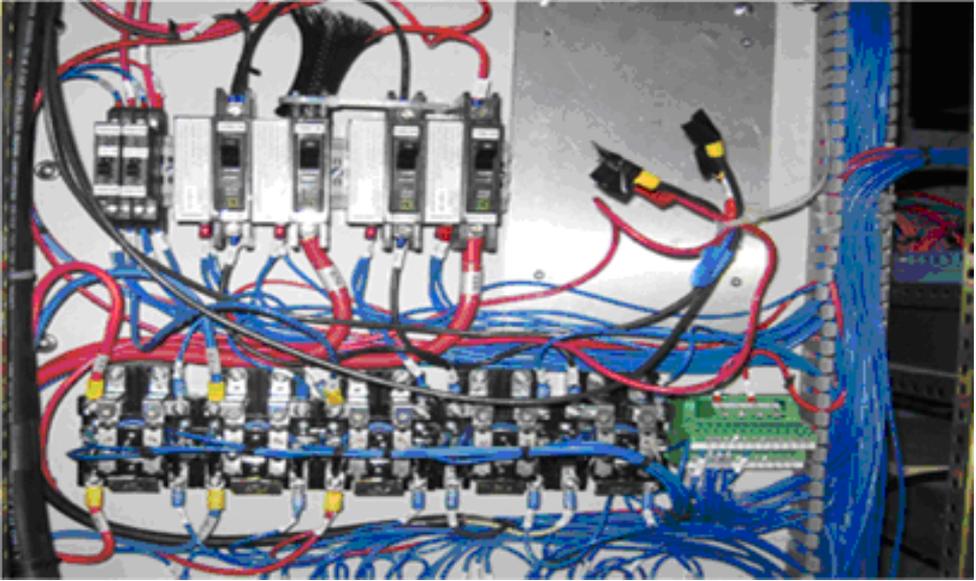


Figure 3.11. Picture of the protection and enable panel taken from [17].

Each panel consists of four relays, a voltage meter and a current meter. Panel i contains three relays connected to charger i and controls which battery pack that is connected to charger i . Panel i also contains a relay connected to battery pack i and controls if that battery pack should be charged at all. Nominally, each charger is only connected to at most one battery pack and each battery pack is only connected to one charger at most. Additionally, under nominal conditions, each charger is prevented from charging another charger. The voltage and current meter inside panel i measures the charging voltage and current into battery pack i . A circuit diagram showing how the three panels interfaces with the batteries, chargers and each other is found in Figure 3.12 and a picture of a battery-charge selection panel is found in the Figure 3.13.

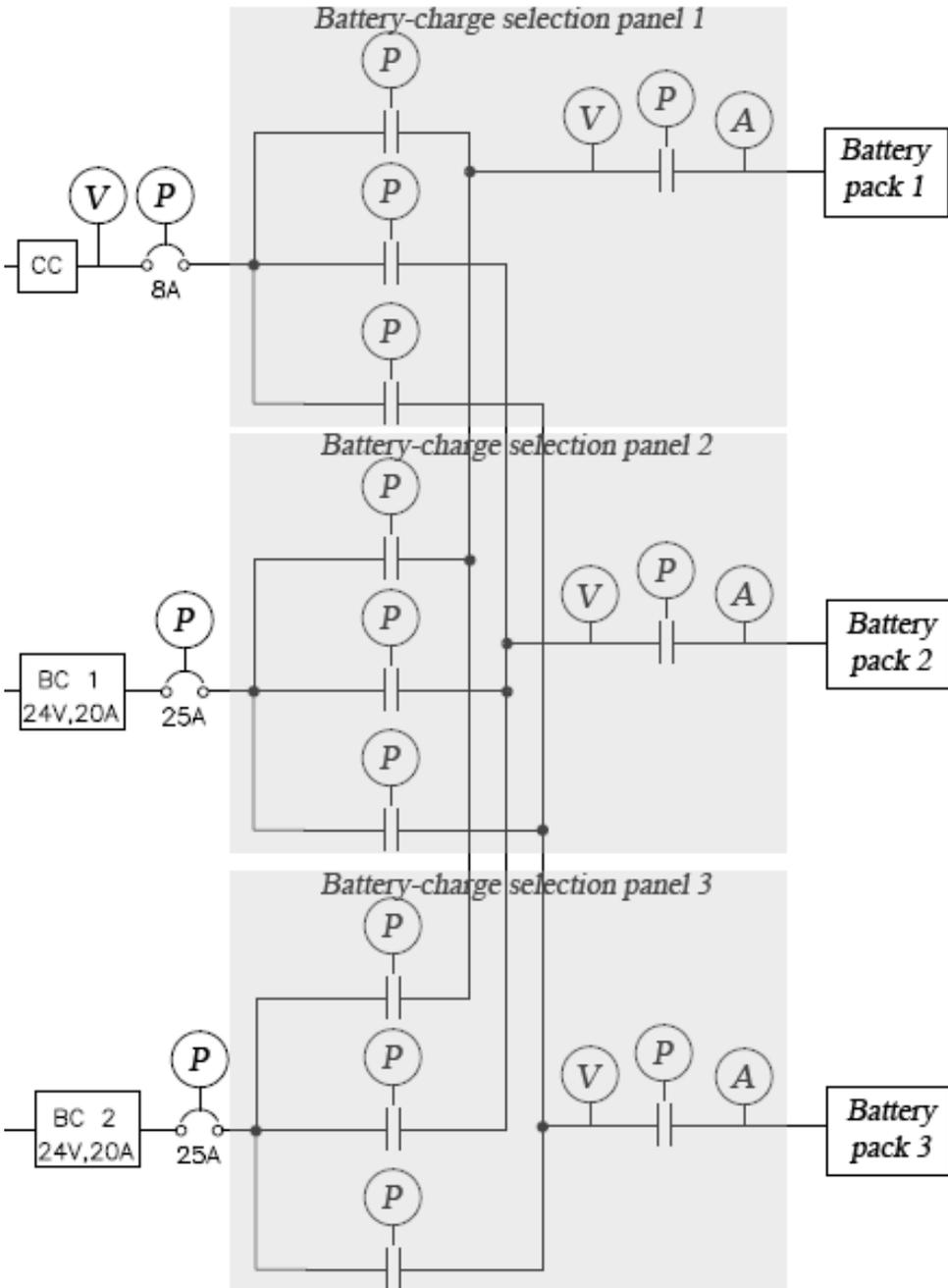


Figure 3.12. Circuit diagram of the three battery-charge selection panels and how they are connected to the chargers and batteries.

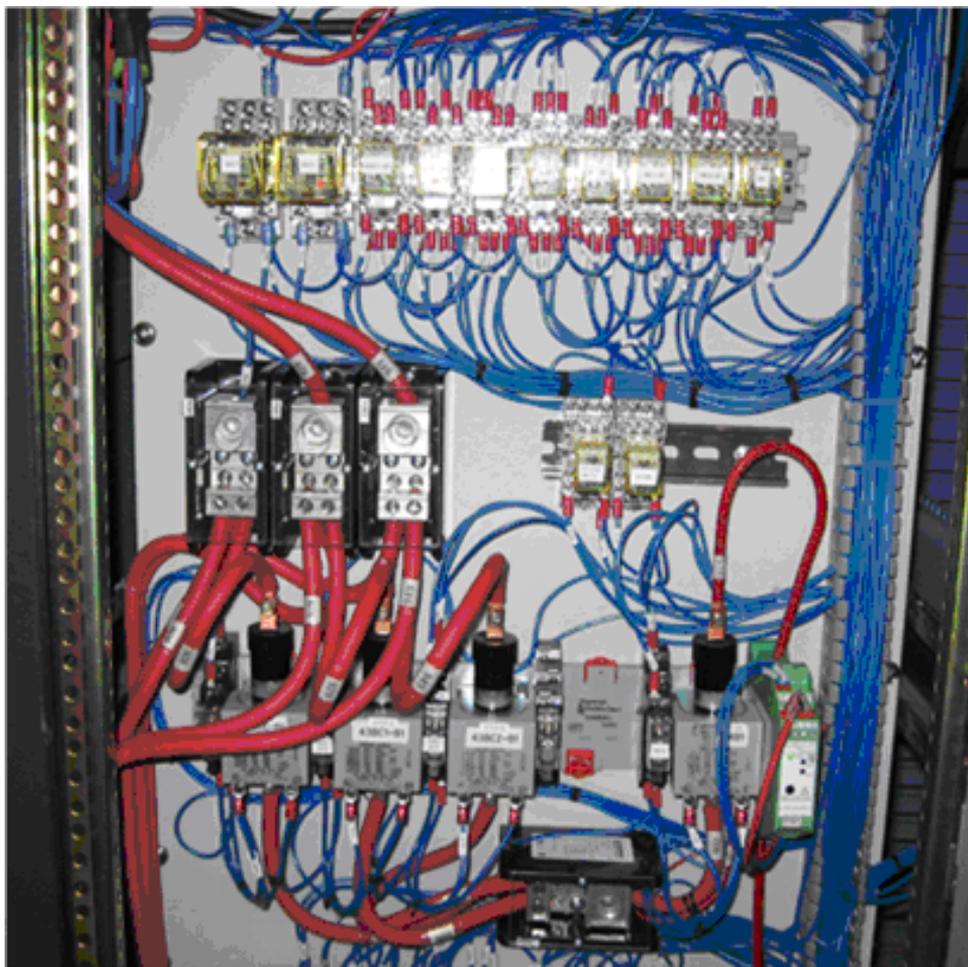


Figure 3.13. Picture of a battery-charge selection panel taken from [17].

3.4.2 Power storage unit

The power storage unit contains three battery packs and several relays that controls which load bank each battery should be connected to. Circuit breakers protects the power distribution unit from dangerously high currents coming from the batteries. The power storage unit can be divided into two major subsystems: the battery cabinet and the battery-load selection panel which are both further described in the following sections.

Battery cabinet

The battery cabinet is connected to the three battery-charge selection panels inside the power generation unit and to the battery-load selection panel inside the power storage unit. The battery cabinet contains three battery packs which consists of two 12 V, 100 Ah batteries connected in series which makes up a total pack voltage of 24 V. Each battery has an approximate internal resistance of 3.4 m Ω when fully charged. Please note that the battery voltage isn't fixed at 12 V but typically varies between 10 and 13.6 V depending on the batteries state of charge and the ambient temperature. The total voltage output of each pack is measured by a voltage meter and the temperature of each battery is measured by resistance temperature detectors (RTD's). To get a reference temperature of the environment temperature surrounding the batteries, an additional RTD has been installed inside the battery cabinet not too close to the actual batteries. It is also worth mentioning that the total area in contact with the surrounding environment of each battery is approximately 0.3 m² and the mass is about 31.2 kg. Between each pack and the battery-load selection panel a circuit breaker has been inserted to protect the power distribution unit. A circuit diagram of the battery cabinet can be found in Figure 3.14 and a picture of it in Figure 3.5.

Battery-load selection panel

The battery-load selection panel is connected to the power distribution unit and to the battery cabinet. This subsystem is very similar to the three battery-charge selection panels in the power generation unit. It contains relays that controls which load bank each battery should be connected to. Nominally, each battery pack is only connected to one load bank at most and each load bank is only connected to one battery pack at most. Additionally, under nominal conditions, each battery pack is prevented from being connected to another battery pack. Three voltage meters and three current meters measure the output of the three battery packs. Two voltage meters and two current meters measures the input current and voltage into the two load banks in the power distribution unit. Two additional voltage meters measures the voltage between the relays. To get a clearer picture of the location of the relays and sensors see the circuit diagram of the battery-load selection panel in Figure 3.15. A picture of the battery-load selection panel can be seen in Figure 3.16.

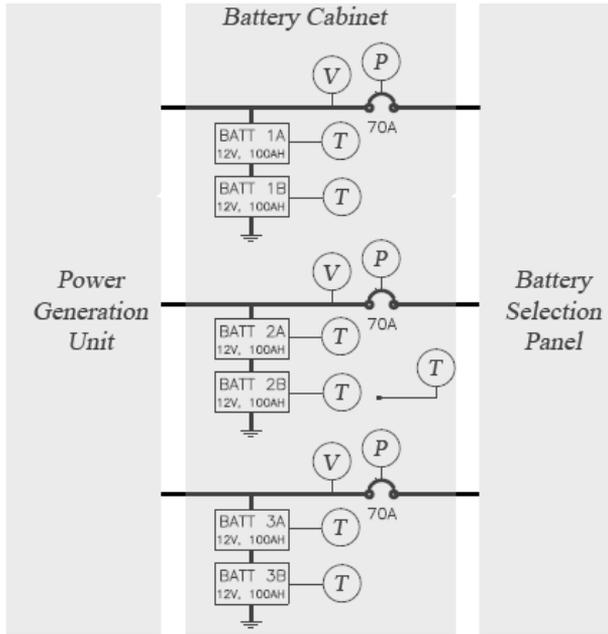


Figure 3.14. Circuit diagram of the battery cabinet.

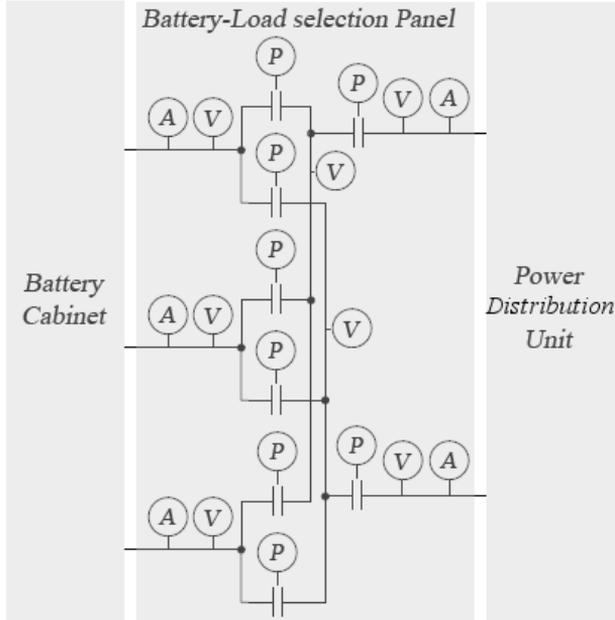


Figure 3.15. Circuit diagram of the battery-load selection panel.

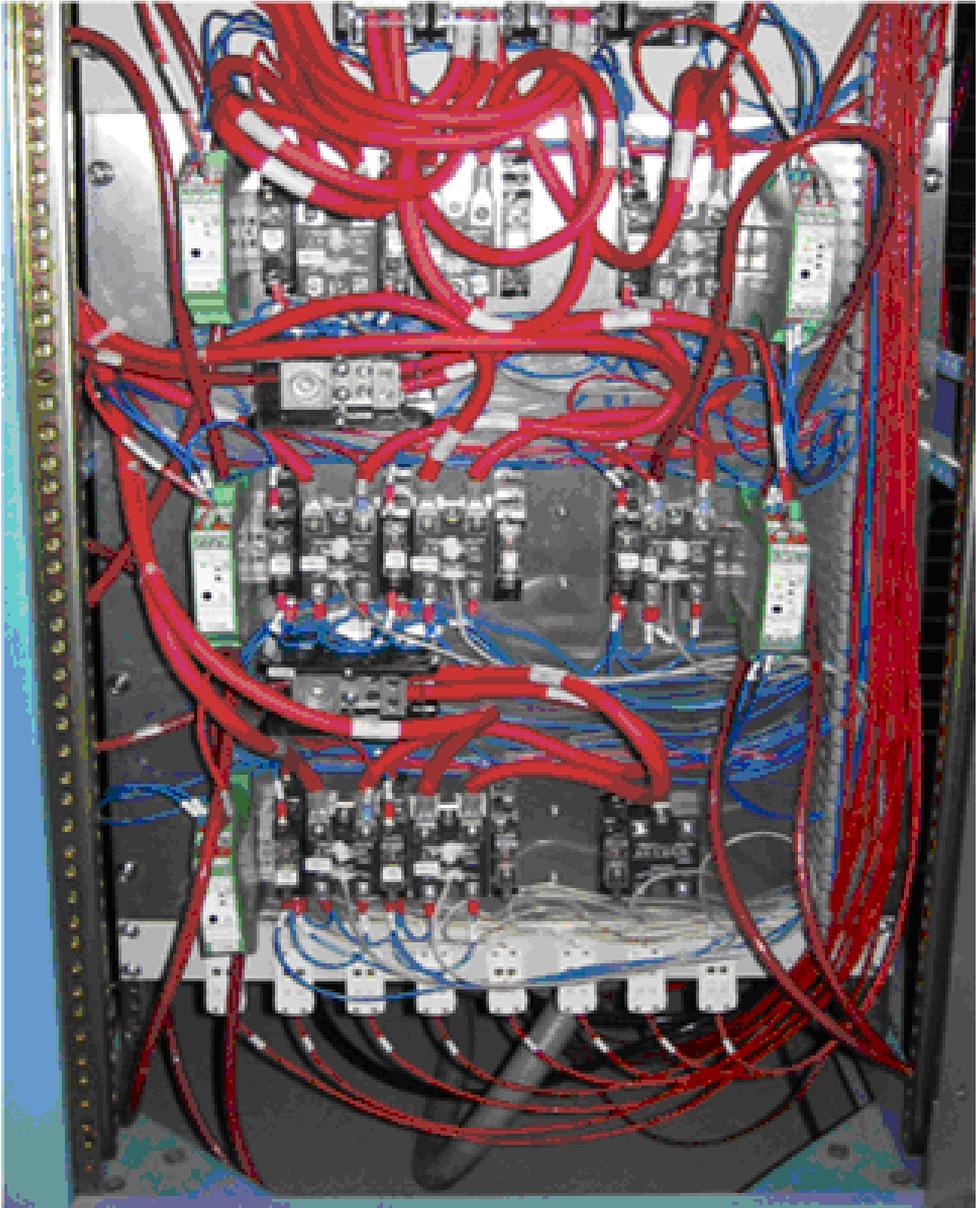


Figure 3.16. Picture of the battery-load selection panel taken from [17].

3.4.3 Power distribution unit

The power distribution unit consists of two identical load banks. Each load bank is connected to the power storage unit and powers two DC loads and six AC loads. Since the power supplied from the power storage unit is only DC each load bank contains an inverter. Each load bank is protected by a circuit breaker before the inverters. The output current and voltage is measured by current meters, voltage meters and frequency transducers. To protect the loads, circuit breakers have been installed between each inverter and the loads. The current and voltage provided to the DC loads are also measured and the DC loads are protected by a circuit breaker. A group of relays controls which relays should be connected to the load banks. A circuit diagram of one load bank can be seen in Figure 3.17. The hardware of both load banks is located inside two inverter panels, a transducer panel and a load selection panel as depicted in Figure 3.18.

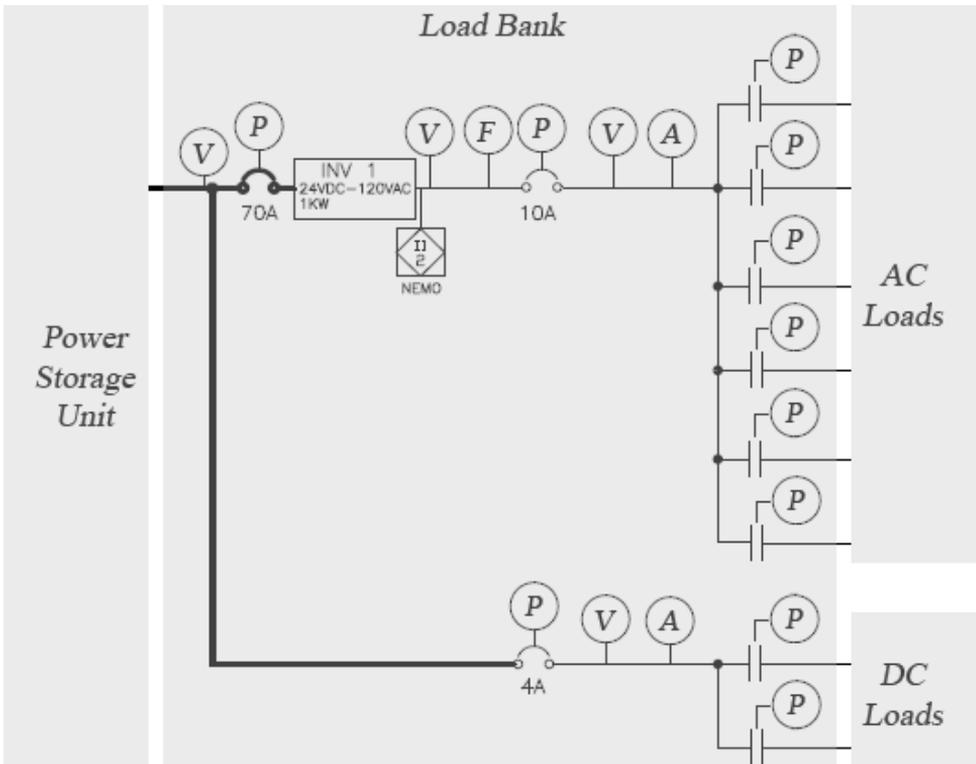


Figure 3.17. Circuit diagram of one of the two identical load banks.

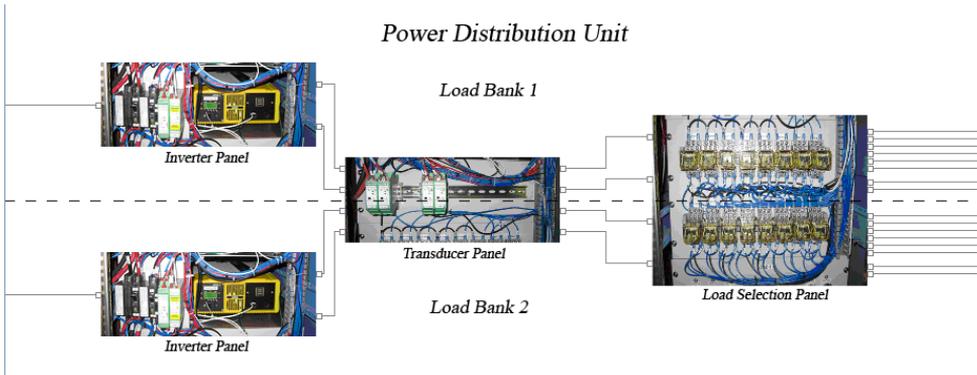


Figure 3.18. Picture of the hardware of the two load banks, based on pictures taken from [17].

3.4.4 Control and monitoring

All relays and circuit breakers in the testbed are connected to position sensors. To send commands to the relays and acquire sensor data from the testbed, National Instrument's LabVIEW software and Compact FieldPoint hardware are used. The hardware consists of two identical Compact FieldPoint backplanes. Each backplane, depicted in Figure 3.19, has eight I/O modules, eight connector blocks and one real-time controller. The two backplanes are connected to a data acquisition computer (DAC) which is connected to a local area network where the user, antagonist, observer and the health management application are connected. The I/O modules found in each backplane are listed below. More detailed information on the Compact FieldPoint hardware used can be found on the National Instruments webpage [5] and a description of the software used in the ADAPT operations and safety manual [17].

- Two cFP-DI-301 digital input modules for relay/circuit breaker position monitoring.
- Two cFP-DO-401 digital output modules for relay control.
- One cFP-AI-100 12 bit analog input module for analog signal monitoring, i.e. currents and voltages.
- Two cFP-AI-102 12 bit analog input modules for analog signal monitoring, i.e. currents and voltages.
- One cFP-RTD-122 12 bit RTD input module for temperature monitoring.



Figure 3.19. Picture of a Compact FieldPoint backplane taken from [17].

3.5 The Advanced Caution And Warning System (ACAWS) scenarios

The advanced caution and warning system (ACAWS) scenarios are a couple of defined scenarios where the loads connected to the EPS are used for environment control and life support systems (ECLSS). In these scenarios the ECLSS function is disrupted by faults injected into the testbed and the user tries to maintain as much as possible of these functions. As mentioned in Section 3.2 the user can use the help of a health management application to detect where the faults have been injected which makes it easier to select an appropriate recovery action.

3.5.1 Scenario loads

As mentioned in Section 3.4.3, the EPS can power two load banks with up to two DC loads and eight AC loads connected. In this case only one DC and eight AC loads are connected to each load bank. The first two AC loads in each load bank are considered critical, therefore each load bank powers two backup loads that can be turned on if the other load bank fails. Additionally, the DC load powered by the first load bank is also considered critical and therefore the DC load connected to the second load bank is a backup of this critical DC load. There are also two non-critical loads connected to each load bank. Figure 3.20 show which loads are connected to which load bank and a short description of the loads.

Loadbank 1		Critical	Load	Description
AC load 1	Yes		Lampbox1	A box containing three 25 W lamps in parallell.
AC load 2	Yes		Fan 1	Fan with low (645 rpm, 87 W), medium (785 rpm, 117 W) and high (955rpm, 166 W) speed settings
AC load 3	No		Small fan A	Small fan consuming 30 W.
AC load 4	No		Silver clamp light 1	Lamp consuming 60 W.
AC load 5	No		Pump 2	Backup pump consuming 35 W.
AC load 6	No		Black clamp light	Lamp consuming 55 W.
DC load 1	Yes		NEMO loads	Loads used by the crew. The size is not specified.
DC load 2	No		Empty	No load connected.
Loadbank 2		Critical		
AC load 1	Yes		Pump 1	Pump consuming 35 W.
AC load 2	Yes		Goose neck light	Lamp consuming 60 W.
AC load 3	No		Small fan B	Small fan consuming 30 W.
AC load 4	No		Silver clamp light 2	Lamp consuming 60 W.
AC load 5	No		Lampbox2	A backup box containing three 25 W lamps in parallell.
AC load 6	No		Fan 2	Fan with low (645 rpm, 87 W), medium (785 rpm, 117 W) and high (955rpm, 166 W) speed settings
DC load 1	No		Empty	No load connected.
DC load 2	No		DC load box	A box containing three resistors with 8, 12 and 24 ohm resistors. Nominally, only the 24 ohm resistor is connected.

Figure 3.20. Table showing which load is connected to which load bank and a short description of each load.

3.5.2 Load monitoring

All the critical loads and their corresponding backup are monitored by a set of sensors. Eight RTD's have been attached to the critical lamps and measures the temperature of their bulbs. Two turbine flow sensors measures the amount of water pumped by the connected pumps and two photoelectric optical pick-up rotation sensors measures the rotation of the two critical fans. In addition to the temperature sensors inside each lamp box a light meter measures the amount of light generated from each lamp box. Figure 3.21 show how the connected loads are monitored.

Loadbank 1	Load	Monitoring
AC load 1	Lampbox1	Three thermistors and one light meter.
AC load 2	Fan 1	A photoelectric optical pick-up sensor measures the rotation.
AC load 3	Small fan A	Not monitored.
AC load 4	Silver clamp light 1	Not monitored.
AC load 5	Pump 2	A turbine flow sensor measures the water flow.
AC load 6	Black clamp light	One thermistor.
DC load 1	NEMO loads	Not monitored.
DC load 2	Empty	No load connected.
Loadbank 2		
AC load 1	Pump 1	A turbine flow sensor measures the water flow.
AC load 2	Goose neck light	One thermistor.
AC load 3	Small fan B	Not monitored.
AC load 4	Silver clamp light 2	Not monitored.
AC load 5	Lampbox2	Three thermistors and one light meter.
AC load 6	Fan 2	A photoelectric optical pick-up sensor measures the rotation.
DC load 1	Empty	No load connected.
DC load 2	DC load box	Not monitored.

Figure 3.21. Table showing how the loads are monitored.

3.5.3 Scenario descriptions

There are fifteen different scenarios specified where faults are injected into the loads, the power distribution unit or the power storage unit. In other words the scope of scenarios only includes the system from the batteries and downstream to the loads as seen in Figure 3.22. The scenarios includes both hardware and software injected faults as well as single and double faults. All scenarios starts in the same way and the first fault is always injected when the system has reached a certain, nominal configuration shown in Figure 3.22. The following sections further describes each scenario.

Scenario 1a

In this scenario, a fault is injected into relay EY170 which enables or disables the first AC load in the first load bank. The three lights in the first lamp box will

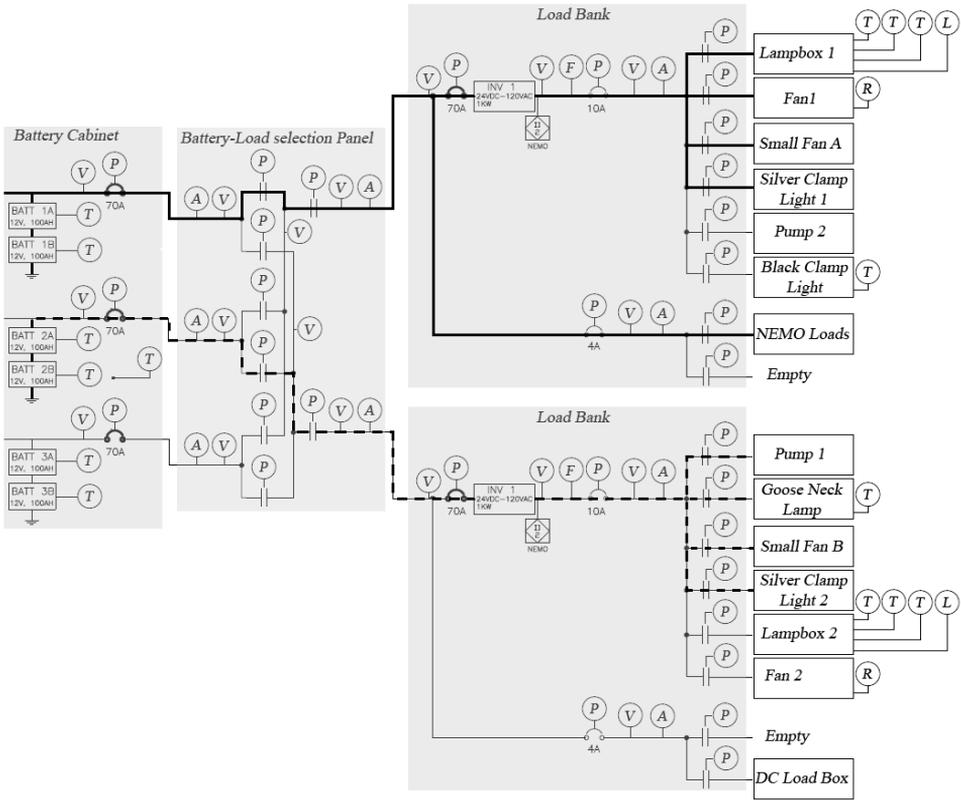


Figure 3.22. Illustration showing the scope of scenarios and the configuration when the first fault is injected.

lose power and the relay position sensor ESH170 will report that the relay is open. Figure 3.23 show where the fault is injected.

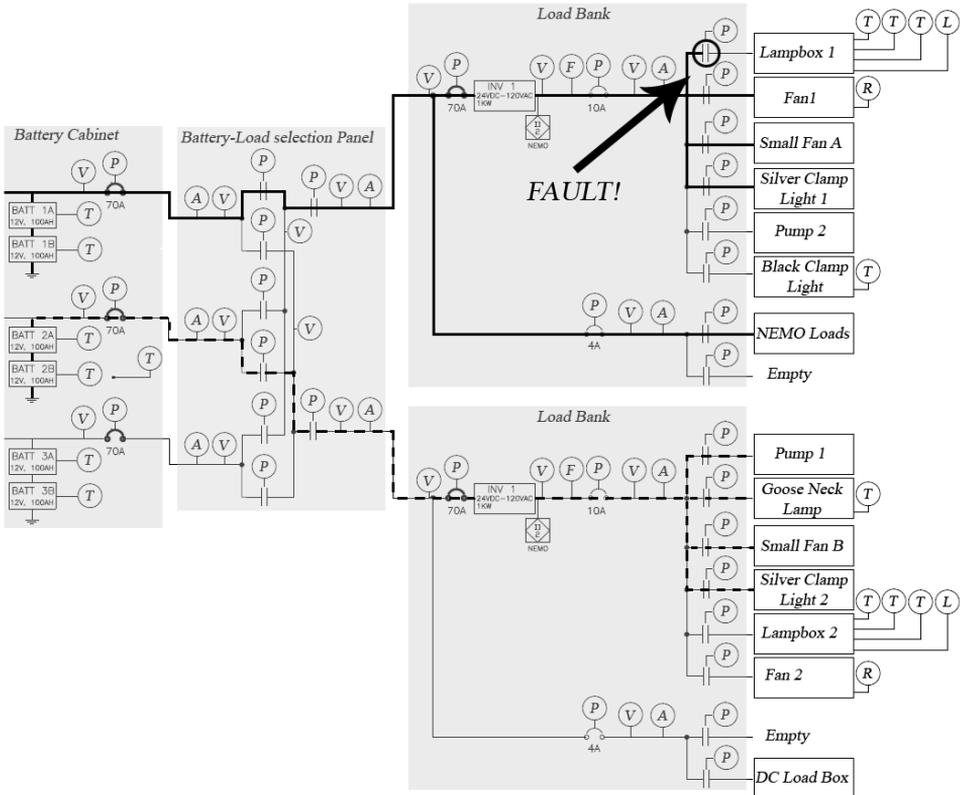


Figure 3.23. Illustration showing where the fault is injected in scenario 1a. In this scenario, the injected fault is relay EY170 stuck open.

Scenario 1b

In this scenario, a fault is injected into relay position sensor ESH271. This sensor monitors the position of relay EY271 which enables or disables the second AC load on the second load bank. The symptom in this case is that the sensor will report that the relay is open despite that the controlled load is operating as if it is powered. Figure 3.24 show where the fault is injected.

Scenario 1c

In this scenario, the first fault is injected into relay EY260 which enables or disables the second load bank. The entire second load bank will lose power. Figure 3.25 show where the first fault is injected. The user will attempt to cycle the relay and will succeed with it. After the system has recovered from the first fault, a

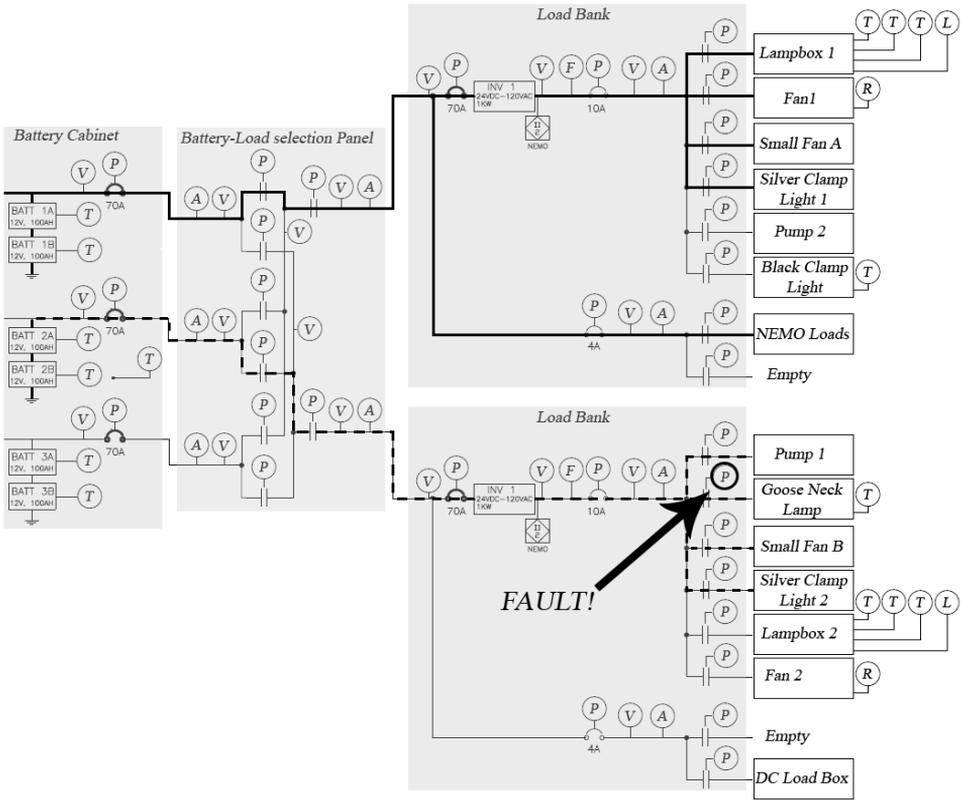


Figure 3.24. Illustration showing where the fault is injected in scenario 1b. In this scenario, the injected fault is relay position sensor ESH271 disconnected.

second faults is injected into relay position sensor ESH171. The sensor will report that relay EY171 is open even though the controlled load is operating as if it is powered. Figure 3.26 show where the second fault is injected.

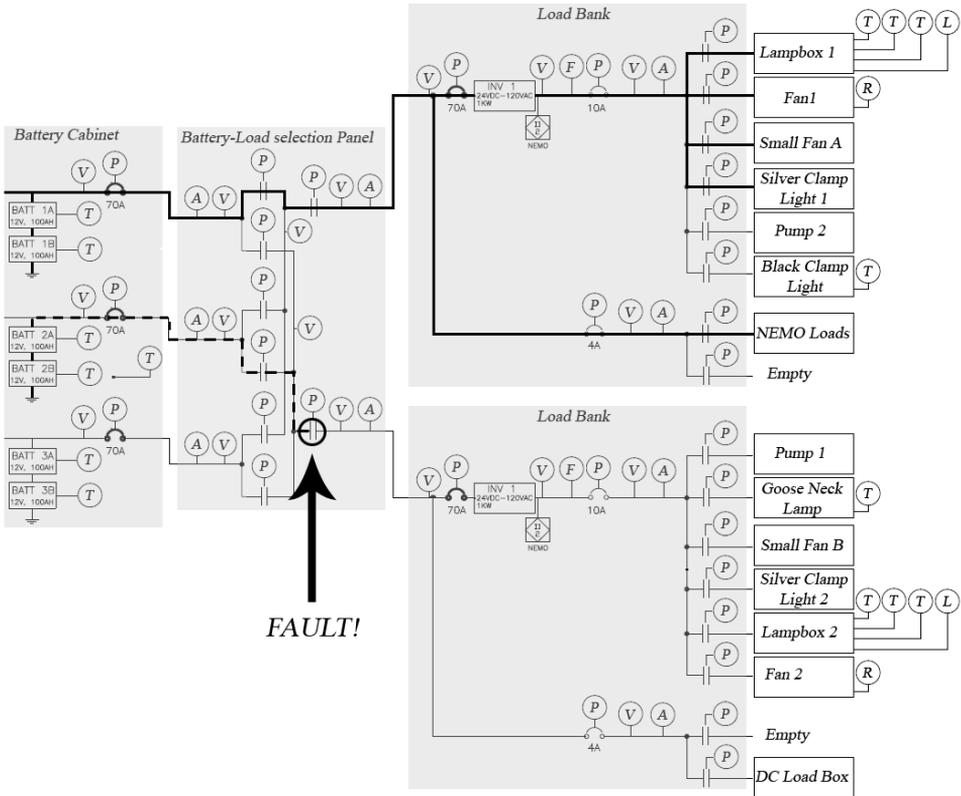


Figure 3.25. Illustration showing where the first fault is injected in scenario 1c. In this scenario, the first injected fault is relay EY260 stuck open.

Scenario 1d

In this scenario, the first fault is injected into relay EY160 which enables or disables the first load bank. The entire first load bank will get turned off. Figure 3.27 show where the first fault is injected. In this scenario, the user will attempt to cycle the relay and will succeed with it. After the system has recovered from the first fault, a second one is injected into relay EY270. The controlled load is off and the relay position sensor ESH270 will report that the relay is open. Figure 3.28 show where the second fault is injected.

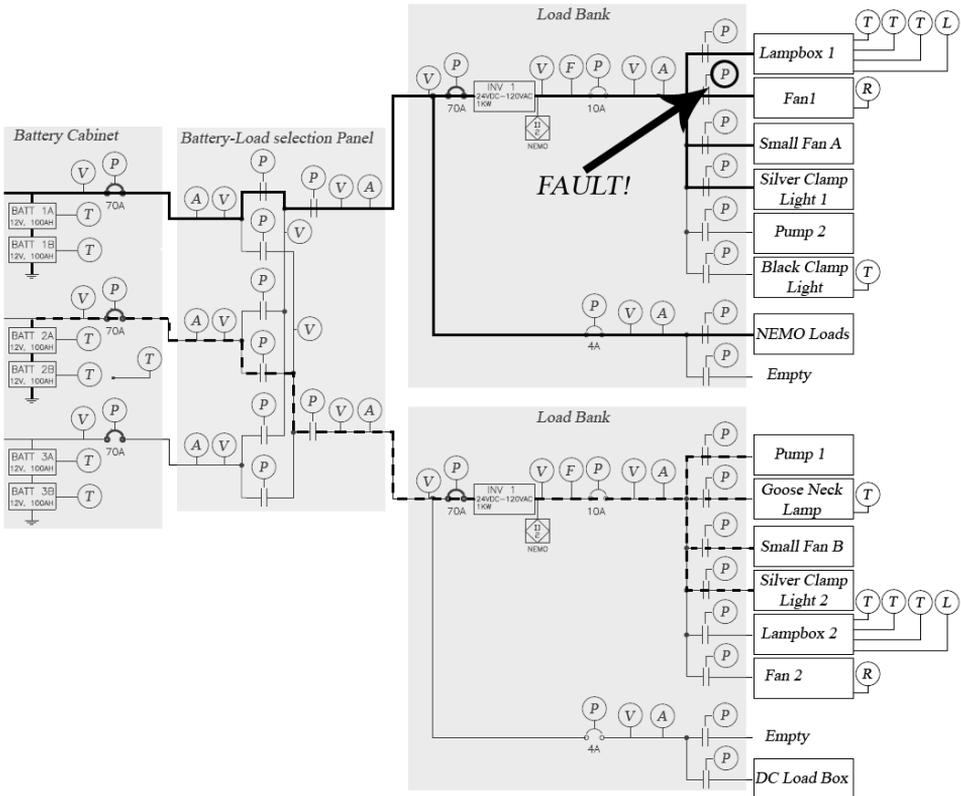


Figure 3.26. Illustration showing where the second fault is injected in scenario 1c. In this scenario, the second injected fault is relay position sensor ESH171 disconnected.

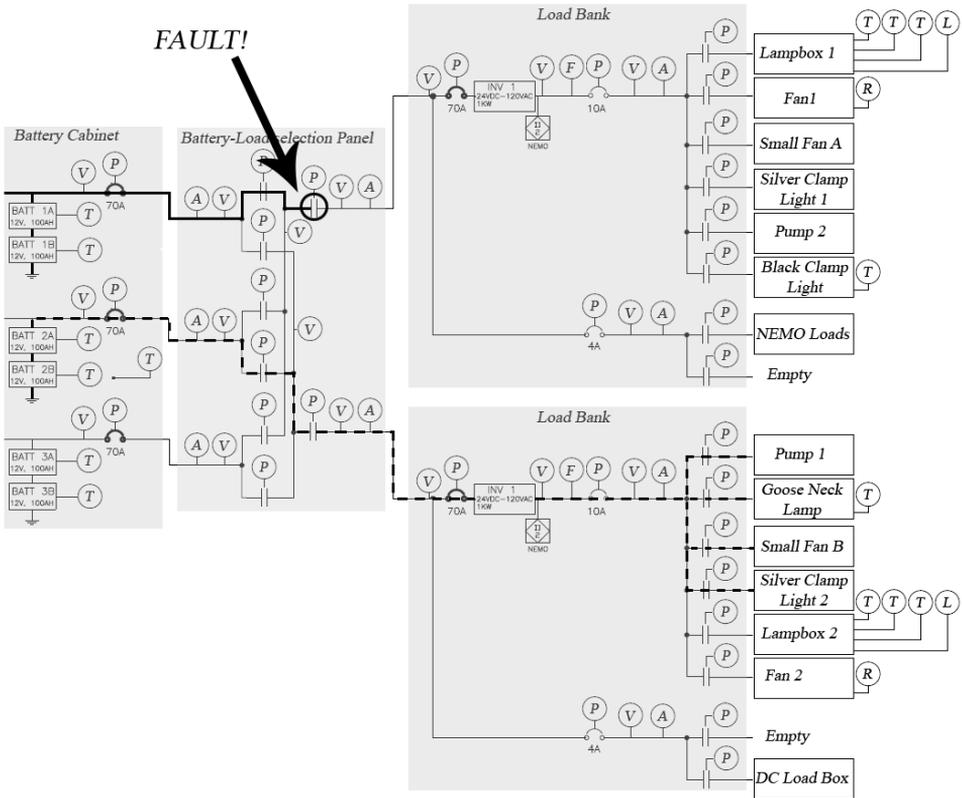


Figure 3.27. Illustration showing where the first fault is injected in scenario 1d. In this scenario, the first injected fault is relay EY160 stuck open.

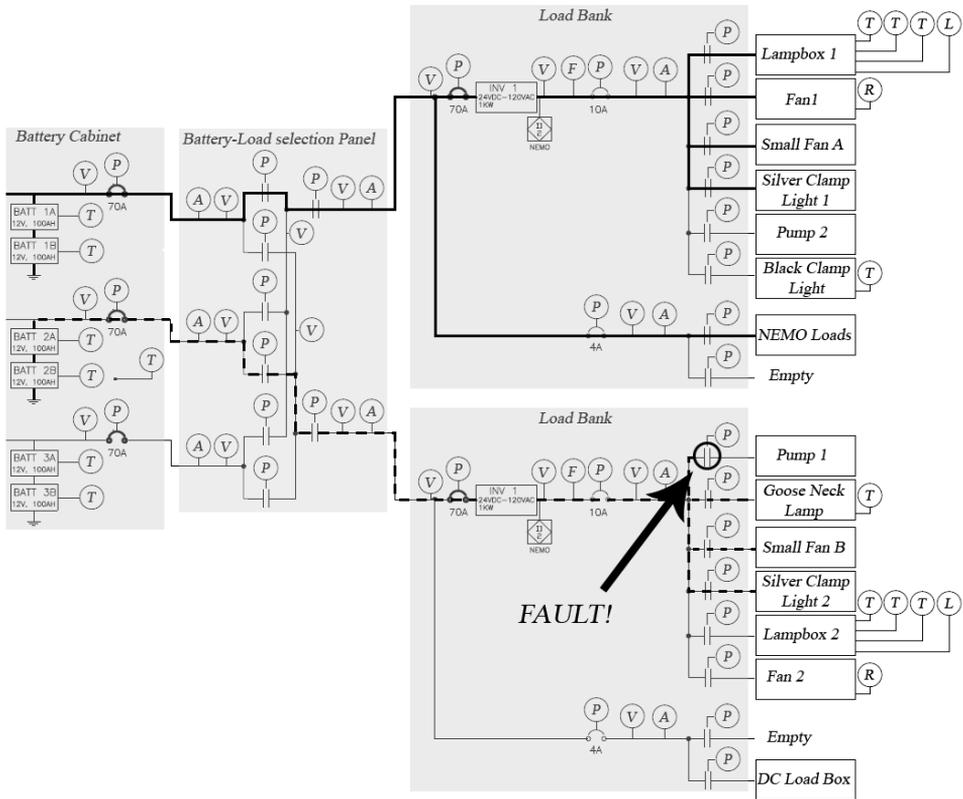


Figure 3.28. Illustration showing where the second fault is injected in scenario 1d. In this scenario, the second injected fault is relay EY270 stuck open.

Scenario 2a and 2b

In scenario 2a, a fault is injected into relay EY141 which controls if the first battery is connected to the first load bank. The entire first load bank will lose power. Figure 3.29 show where the fault is injected. Scenario 2b is very similar to 2a since the fault is injected into relay EY244 which controls if the second battery is connected to the second load bank and the entire second load bank will lose power.

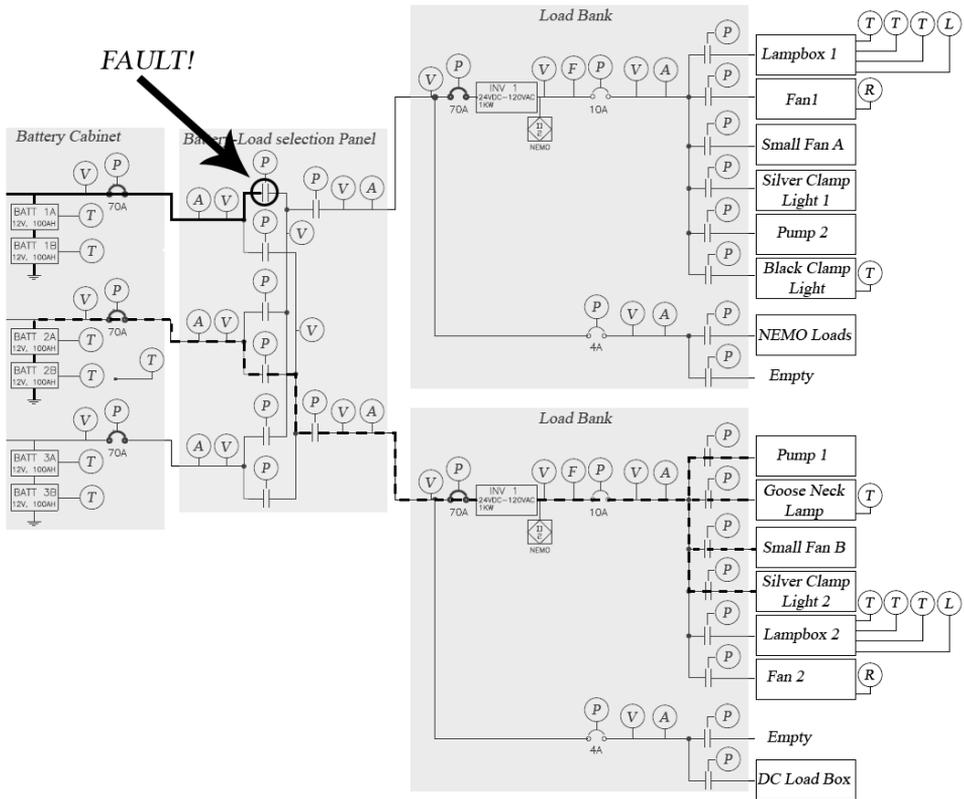


Figure 3.29. Illustration showing where the fault is injected in scenario 2a. In this scenario, the injected fault is relay EY141 stuck open.

Scenario 3a and 3b

In scenario 3a, a fault is injected into battery pack A. The entire first load bank will get turned off. Figure 3.30 shows where the fault is injected. Scenario 3b is very similar to 3a since the fault is injected into battery pack B.

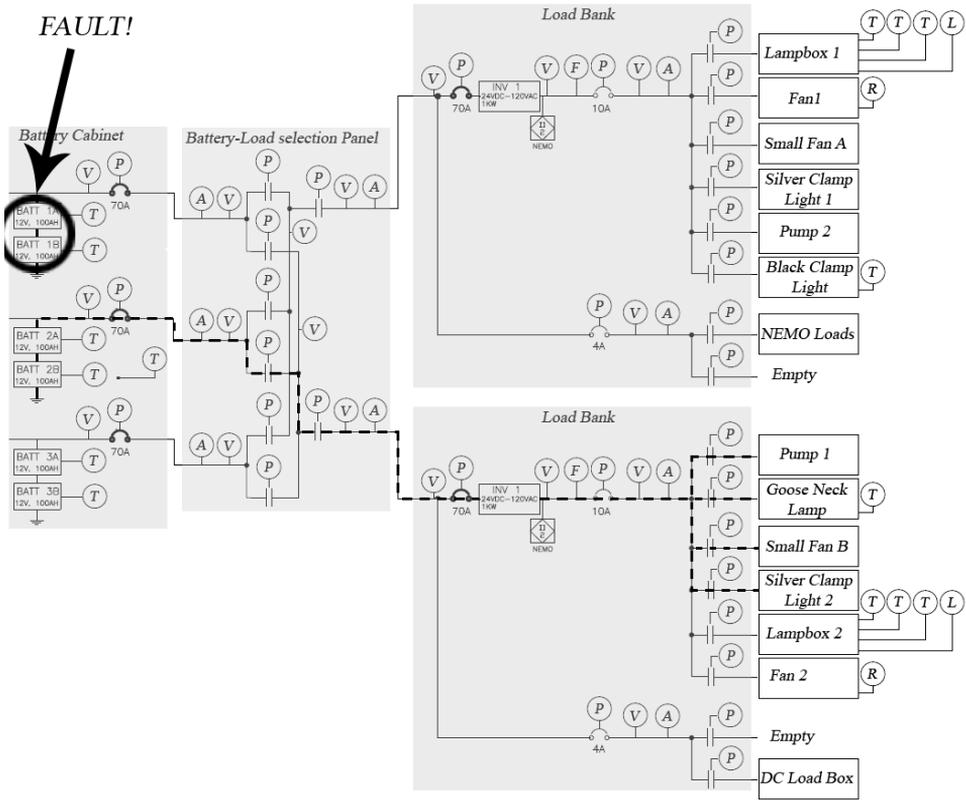


Figure 3.30. Illustration showing where the fault is injected in scenario 3a. In this scenario, the injected fault is battery pack B disconnected.

Scenario 4a and 4b

In scenario 4a, a fault is injected into the inverter in the first load bank. All AC loads in this load bank will lose power. Figure 3.31 show where the fault is injected. Scenario 4b is very similar to 4a since the fault is injected into the inverter in the second load bank.

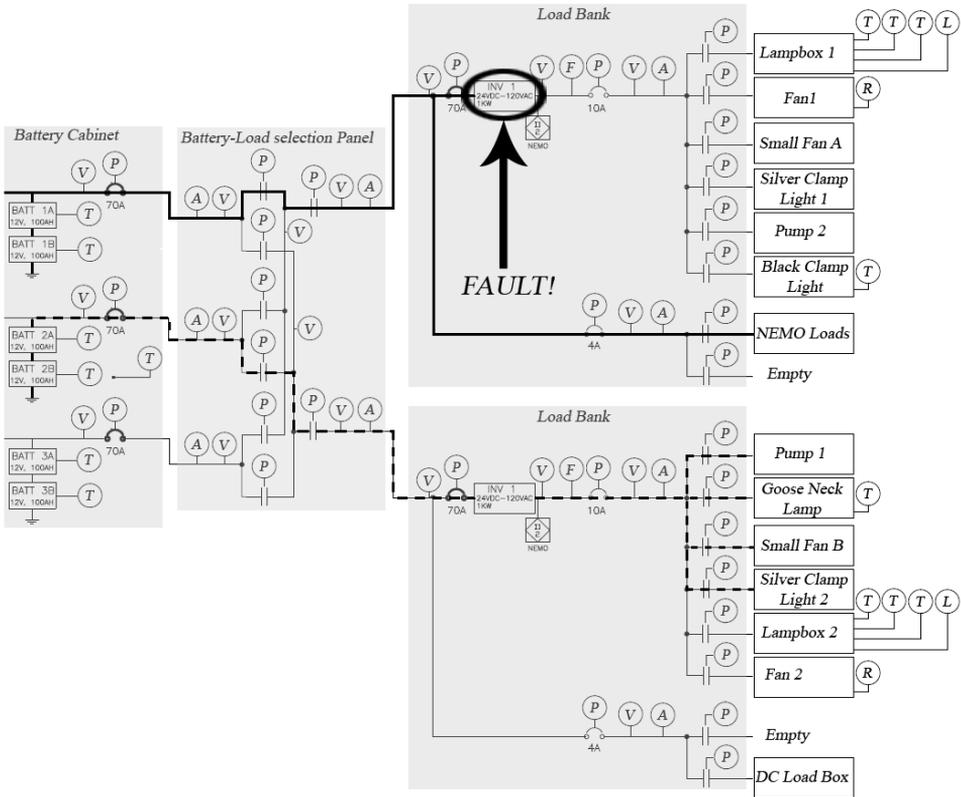


Figure 3.31. Illustration showing where the fault is injected in scenario 4a. In this scenario, the injected fault is the inverter in the second load bank disconnected.

Scenario 5a and 5b

In scenario 5a, the first fault is injected into temperature sensor TE500 which will report a low value. The second fault is injected into the inverter in the second load bank and all AC loads in the second load bank will lose power. Figure 3.32 shows where the two faults are injected. Scenario 5b is very similar to 5a since the faults are injected into the inverter in the first load bank and RTD TE502.

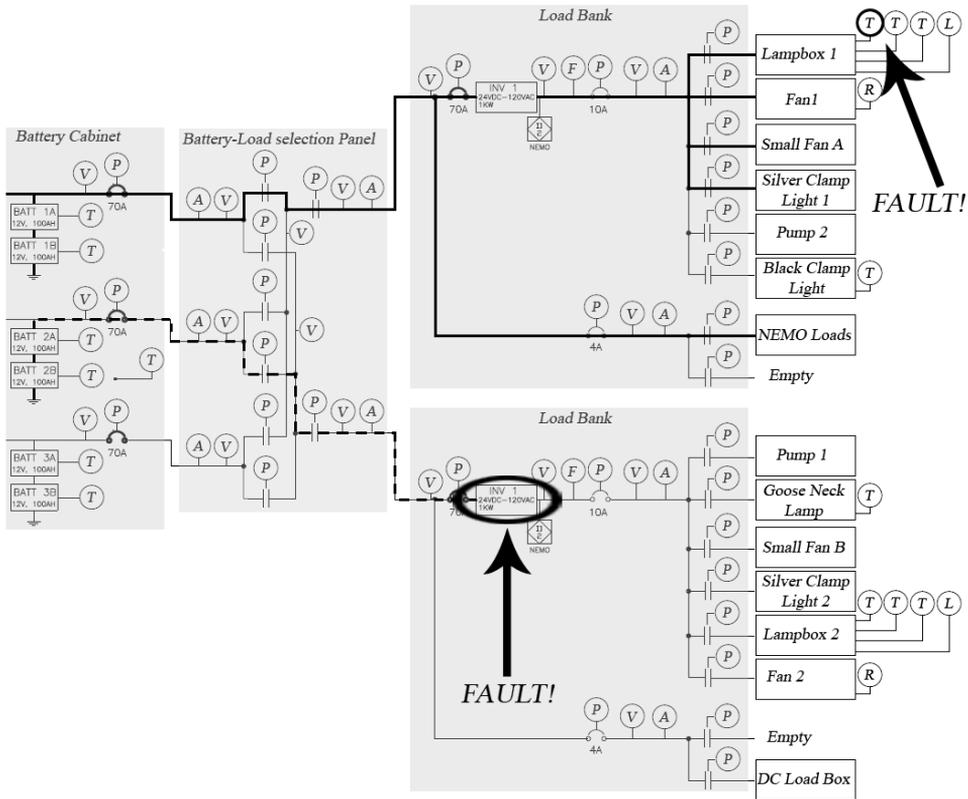


Figure 3.32. Illustration showing where the two faults are injected in scenario 5a. In this scenario, the injected faults are the inverter in the second load bank disconnected and temperature sensor TE500 stuck on a low value.

Scenario 6a and 6b

In scenario 6a, the first fault is injected into the inverter in the first load bank and all AC loads in the first load bank will lose power. The user will turn on the critical backup loads and turn off the non-critical loads connected to the second load bank. Figure 3.33 shows where the first fault is injected and the configuration of the system when the user has taken action. In this configuration, the second fault is injected into circuit breaker ISH180 and the critical DC load connected to the first load bank will lose power. Figure 3.34 show where the second fault is injected. Scenario 6b is very similar to 6a since the same faults are injected, but the faults are injected in the reverse order.

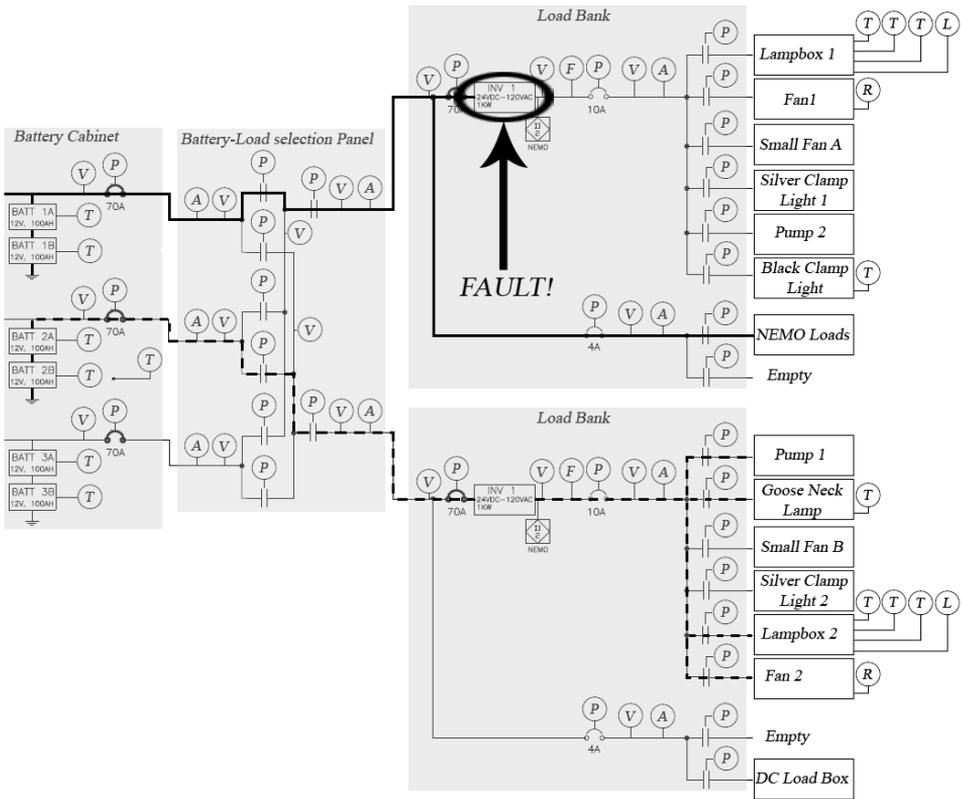


Figure 3.33. Illustration showing where the first fault is injected in scenario 6a. The system has this configuration when the user has taken action against the first fault. In this scenario, the first injected fault is the inverter in the first load bank disconnected.

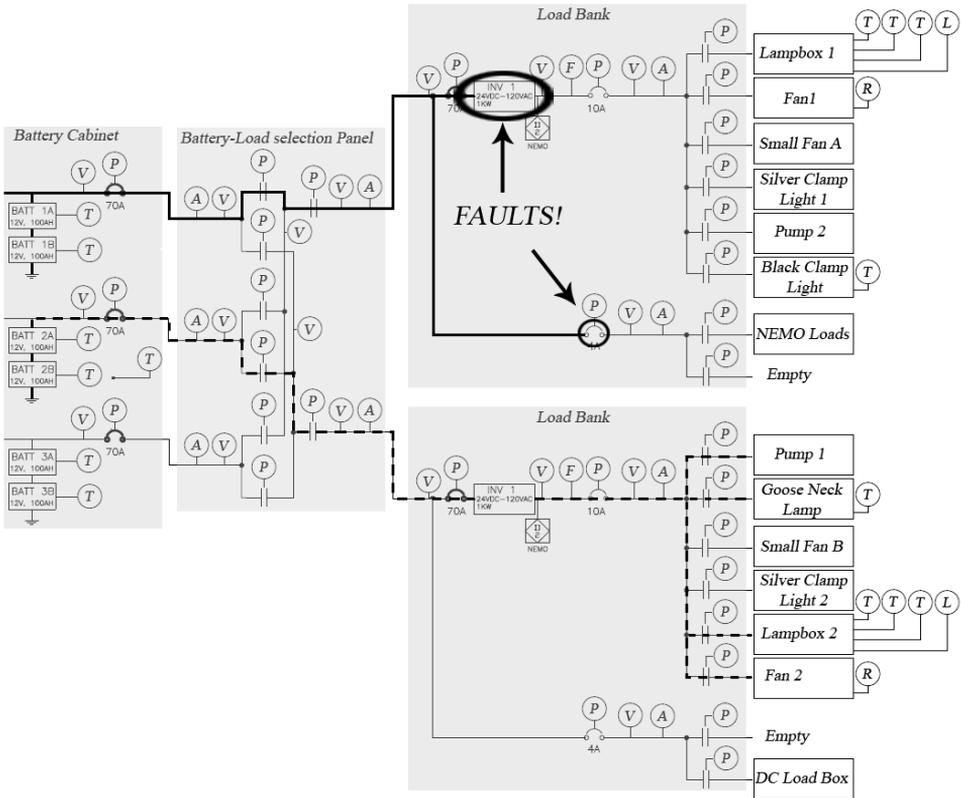


Figure 3.34. Illustration showing where the second fault is injected in scenario 6a. In this scenario, the second injected fault is circuit breaker ISH180 disconnected.

Chapter 4

Model of the Advanced Diagnostics and Prognostics Testbed (ADAPT)

To be able to perform model-based diagnosis, a model is needed. As mentioned in chapter 2, RODON is a component based modeling tool that supports failure modes. The model is comprised of 884 components with both nominal and faulty behavior. The model has been restricted to the stationary case and only uses data from one time instance. This chapter covers the physics behind each component and how it was implemented in RODON.

4.1 Physical models of the components

This section describes the physics behind each component. The components have both nominal and faulty behaviors which have been modeled.

4.1.1 Wire

The electrical wire is the simplest component in the system. An ideal wire has been modeled which means that the resistance of the wire is zero. The small resistance of the real wires in the system would be lost in the noise of the voltage sensors. The size of the sensor noise is seen in Figure 4.1. In the nominal case there is no voltage drop over the wire and the currents at both ends are equal. The model contains two failure modes: "disconnected" and "short to ground". When the wire is disconnected, there is no relation between the voltages at both ends and the current through the wire is zero. When the wire is shorted to ground the potential at both ends are zero and the current balance between both ends is broken.

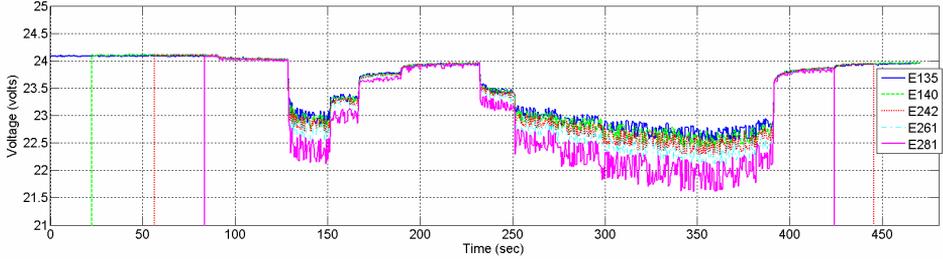


Figure 4.1. Plot of the measured voltage from five voltage meters. The level of noise depends on the connected loads. The figure is taken from *Evaluation, Selection, and Application of Model-based Diagnosis Tools and Approaches* by Scott Poll et al. [18].

4.1.2 Resistor

When the resistor is in its nominal state, the voltage drop across it is equal to the current flowing through it multiplied with its resistance as in Ohm’s law: $U = I \cdot R$. The model contains two failure modes: ”disconnected” and ”shorted”. When the resistor is disconnected it behaves like a disconnected wire, see Section 4.1.1. When the resistor is shorted it behaves like an ideal wire in its nominal state, see Section 4.1.1.

4.1.3 Relays

The relay is the most common component in the testbed and the model contains 39 of them enabling over 549 billions system configurations. ADAPT contains two types of relays: electromechanical relays and solid state relays (SSR). Both relay types are modeled by the same equivalent circuit containing a resistor and a switch. In the electromechanical case, the resistor represents the coil and in the solid state case it could represent a diode if it’s a photo coupled SSR for example. In either case, when the resistor is consuming enough power it will change the position of the switch to ”closed”. If the resistor isn’t consuming enough power the switch will remain in position ”open”. The resistor is connected to the real-time controller through wires.

Electromechanical relay

Two failure modes have been modeled in the electromechanical relay model: ”stuck open” and ”stuck closed”. When the relay has failed in either of these modes, the position of the switch is stuck at a position and is not affected by the resistor.

Solid state relay

The model of the SSR is the same as the model for the electromechanical one with one exception. It contains an extra failure mode: ”overheated”. The behavior of this failure mode is equal to the behavior of the ”stuck closed” failure mode.

4.1.4 Circuit breaker

A circuit breaker is like a fuse, it will trip and cut the connection if a high current is flowing through it. The model contains one failure mode: "disconnected" and its resistance has been neglected for the same reason mentioned in Section 4.1.1. When the circuit breaker is disconnected it will behave like it has tripped regardless of the amount of current flowing through it.

4.1.5 Battery charger

The battery chargers used in the system uses 120 V AC to charge a 24 V battery pack. The model is split in two parts: the AC and DC side. The AC side is modeled as a resistor connected to ground. When there is a current going through the resistor the battery charger is "on" unless it is "disconnected". If the same current is zero the battery charger is "off".

The DC side of the battery charger is modeled as an ideal battery in series with a resistor which represents its internal resistance. In the stationary state, the charging current is low and the charging voltage is equal to the maximum battery voltage, see Section 3.8. However, this stationary state is reached after several hours of charging making a detailed stationary model unpractical to use. Therefore the charging current and voltage are left undefined in the ranges of 0 to 20 A and 20 to 50 V respectively if the battery charger is on. When the battery charger is off the DC side is disconnected and the charging current is 0 A.

In addition to the failure mode disconnected, a failure mode called "overcharging" has been modeled. This happens when the control circuits inside the charger have failed in some way and the battery is being charged incorrectly. Since the charging current and voltage is practically undefined it is not possible to say if the battery is being overcharged by just looking at the electrical properties of the charger. Instead, the model uses the battery temperature and determines if it is higher than a certain threshold. In that case, the battery is being overcharged and the battery charger is faulty and running in the overcharging mode. This temperature threshold is further discussed in Section 4.1.8.

4.1.6 Charge controller

The charge controller model is very similar to the battery charger model. The only difference is that it uses DC power from the solar panel instead of the AC power supplied from the power grid to charge the battery.

4.1.7 Sensors

The testbed model contains 99 sensors of different types. Each sensor is connected through wires to the I/O-hardware where the signals are decoded and presented to the user. All sensors in the model except the temperature and position sensors have three different failure modes: "disconnected", "short to ground" and "stuck". When a sensor fails it will report an incorrect value and the system being monitored is not affected by its failure. When the sensor is disconnected the current going to

the I/O-hardware is 0 A. When the sensor is shorted to ground the voltage on the side leading to the I/O-hardware is 0 V. When the sensor is stuck the reported value could be anything within the sensors operating range. All sensors except the temperature and position sensors produce a voltage proportional to the quantity being measured. Figure 4.3 shows how all sensors except the temperature and position sensors are modeled.

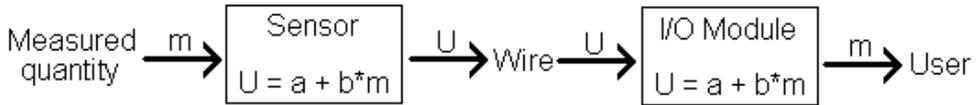


Figure 4.2. Illustration of how all sensors except the temperature and position sensors are modeled. The decoded voltage is proportional to the quantity the sensor is measuring.

Temperature sensor

The temperature sensors in the testbed are all platinum resistance temperature detectors (platinum RTD's). An RTD is a resistor with a resistance that varies with temperature. The resistance of the RTD is approximated to be proportional to the temperature. The approximation is sufficient [19] because the measured temperatures are in a small interval. To get the temperature, the I/O-hardware simply measures the resistance of the RTD and calculates the temperature. The RTD can fail in the same way as the resistor described in Section 4.1.2. In addition to those failure modes the RTD can also get stuck in the same way as the sensors described earlier.

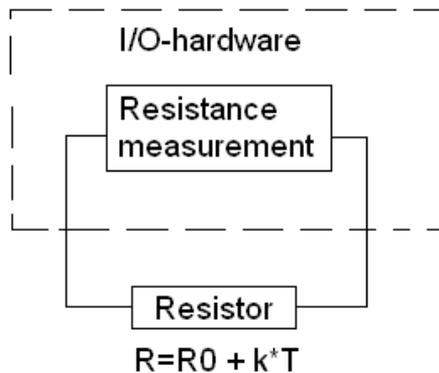


Figure 4.3. Illustration of how a RTD is connected to the I/O-hardware.

Position sensor

The position sensors in the testbed report the position of the relays and circuit breakers. The position sensor is modeled as a switch with a position equal to the position of the monitored component, unless the sensor has failed. The position sensor is electrically isolated from the measured component. Figure 4.4 shows how the position sensor model interacts with the relay model and the I/O-hardware. The model contains two failure modes: "disconnected" and "shorted". When the position sensor is disconnected the switch is stuck open and when it is shorted the switch is stuck closed.

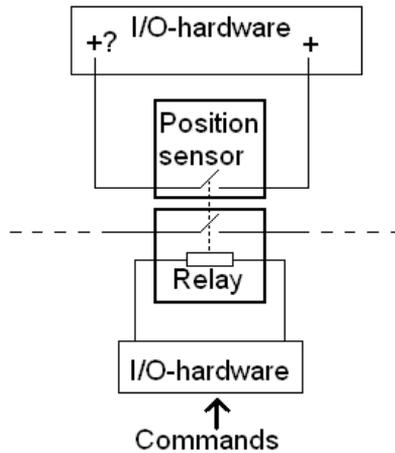


Figure 4.4. Illustration of how the position sensor model interacts with the relay model and the I/O-hardware.

4.1.8 Accumulator

An accumulator is a rechargeable battery. The accumulator is modeled as an ideal battery in series with a resistor which represents the internal resistance of the battery. The size of the internal resistance, R_{int} , is about $3.4 \text{ m}\Omega$ when fully charged [4] but it is approximated to be $3.4 \text{ m}\Omega$ independent of the battery's state of charge in the nominal case. When the battery is being charged the total battery voltage, the potential difference between the two battery poles $V_p - V_n$, is higher than during discharge [4]. During charging there will be a current, $I > 0$, flowing from the positive to the negative pole causing a voltage drop across the internal resistance which adds up with the voltage of the ideal battery U_{batt} . During discharge the current flowing through the resistance, $I < 0$, will flow in the opposite direction causing a voltage drop that works against the ideal battery voltage. This behavior is depicted in Figure 4.5 and is expressed in equation 4.1. If the charger is trying to charge the battery while loads are powered by the battery, the charging current is diverted to the loads because it will chose the path which has the lowest

potential. The model contains three failure modes: "disconnected", "shorted" and "damaged cell". When the battery is disconnected, there is no relation between the potential of the two poles and the current at each pole is zero. If the battery is shorted it will behave as an ideal wire in its nominal state. If one or more of the cells in the battery is damaged, the internal resistance is larger than normal: $R_{int} > 3.4m\Omega$.

$$V_p - V_n = U_{batt} + IR_{int} \tag{4.1}$$

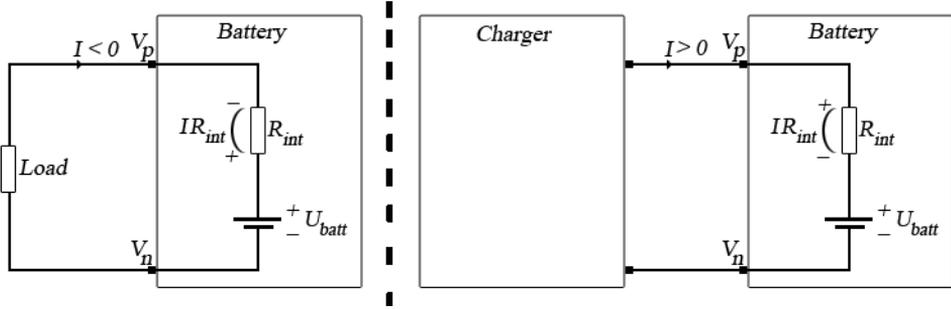


Figure 4.5. The total battery voltage when the battery is discharging to a load to the left and when the battery is being charged to the right.

Temperature dependence

The temperature of each battery is monitored which makes it interesting to model what the battery temperature depends on. One thing which is always true when the battery is discharging to loads or when it is being charged, is that the internal resistance will consume some power which is converted directly to heat, Q_{int} . The time derivative of this heat is equal to the current flowing through the battery, I , squared times the size of the internal resistance, R_{int} . This relation is expressed in equation 4.2.

$$\dot{Q}_{int} = I^2 R_{int} \tag{4.2}$$

When the battery is being charged the total charging power, P_{charge} , is equal to the charging current, I , multiplied with the potential difference between the two battery poles $V_p - V_n$. This relation is expressed below.

$$P_{charge} = (V_p - V_n)I \tag{4.3}$$

The part of the total charging power that is not consumed by the internal resistance is consumed by chemical reactions inside the ideal battery. This relation is expressed in equation 4.4 where the power consumed by these chemical reactions is called P_{chem} .

$$P_{charge} = \dot{Q}_{int} + P_{chem} \tag{4.4}$$

The temperature of each cell depends on the pressure inside it [21]. The size of the pressure depends on how much gas is generated inside the cell. The pressure stays very low during most of the charging process but when the battery is approaching its fully charged state, the amount of generated gas increases [21], [8]. When the cell is fully charged, all [21] or nearly all [8] of the power P_{chem} is consumed by gas-generating reactions which increases the battery temperature. Based on this information, the power P_{chem} has been modeled to behave in the way listed below.

- During normal charging, all of the power P_{chem} will be used to change the battery's state of charge without generating heat.
- During overcharging, all of the power P_{chem} will be converted into heat.

Since the battery is in contact with the surrounding environment, it will dissipate heat to it. The temperature of the battery, T_{batt} , depends on how effectively it can dissipate heat to the environment. *Newtons law of coling* gives the time derivative of the total amount of dissipated heat to the environment \dot{Q}_{env} . \dot{Q}_{env} is proportional to the temperature difference between the body and the environment. This relation is expressed in equation 4.5 where α is the heat transfer number of the battery and T_{env} is the environment temperature.

$$\dot{Q}_{env} = \alpha(T_{batt} - T_{env}) \quad (4.5)$$

The difference between \dot{Q}_{int} and \dot{Q}_{env} is equal to the part converted to a temperature rise. This relation is expressed in the *basic formula of calorimetrics* used in equation 4.6 where c is the specific heat capacity and m is the mass of the battery.

$$\dot{Q}_{int} - \dot{Q}_{env} = cm\dot{T}_{batt} \Leftrightarrow I^2 R_{int} - \alpha(T_{batt} - T_{env}) = cm\dot{T}_{batt} \quad (4.6)$$

The stationary battery temperature can now be expressed in terms of the internal resistance and the current flowing through it. In the stationary case the time derivative of the battery temperature \dot{T}_{batt} is zero. If this information is used in equation 4.6 we get equation 4.8 below.

$$I^2 R_{int} - \alpha(T_{batt} - T_{env}) = 0 \Leftrightarrow T_{batt} = T_{env} + \frac{I^2 R_{int}}{\alpha} \quad (4.7)$$

This model contains one unknown parameter that had to be determined: the heat transfer number of the battery to the environment, α . The area of the battery (0.3 m^2) was used to determine the thermal resistance R_T of a rectangular box with flat surfaces. The thermal resistance is inversely proportional to the heat transfer number α . This relation and the result is presented in the equation below.

$$R_T = 2.5 \text{ } ^\circ\text{C}/\text{W} = \frac{1}{\alpha} \Leftrightarrow \alpha = \frac{1}{2.5} \text{ W}/^\circ\text{C} \quad (4.8)$$

Overcharging

During overcharging, all of the power P_{chem} is converted into heat. Equation 4.4 states that in this case, all charging power P_{charge} is converted into heat. The analysis in Section 4.1.8 is repeated with all charging power converted into heat instead of only using the heat from the internal resistance. The analysis gives the stationary battery temperature expressed in equation 4.9.

$$P_{charge} - \alpha(T_{batt} - T_{env}) = 0 \Leftrightarrow T_{batt} = T_{env} + \frac{P_{charge}}{\alpha} \quad (4.9)$$

This relation combined with equation 4.3 gives the equation below.

$$T_{batt} = T_{env} + \frac{(V_p - V_n)I}{\alpha} \quad (4.10)$$

The model uses the battery temperature to determine if the battery is being overcharged. If the battery temperature is higher than T_{env} and lower than $T_{env} + \frac{I^2 R_{int}}{\alpha}$ the battery is not being overcharged. However, if the temperature is larger than $T_{env} + \frac{I^2 R_{int}}{\alpha}$ it is being overcharged. If a battery is being overcharged, the connected charger is considered faulty, not the battery.

4.1.9 Solar panel

The solar panel is basically 72 photovoltaic polycrystalline cells connected in series in an array as seen in Figure 4.6. Each cell contributes with a small voltage of approximately 0.5 to 0.65 V under normal conditions [15], [2]. The model contains three failure modes: "disconnected", "blocked" and "partially blocked". When the solar panel is blocked, the exposed area in the model is zero. If it is partially blocked the exposed area in the model is between zero and the nominal exposed area of 0.864 m^2 . Finally, if the solar panel is disconnected there is no relation between the electrical potentials of two terminals and the current flowing through them is zero.

The P-N junction

To understand how a photovoltaic cell works, knowledge of P-N junctions [15], [2] is required. A photovoltaic cell is basically a large P-N junction. In pure silicon structures, four atoms with four valence electrons each form a covalent bond. When one of these atoms are replaced by an impurity atom the material is *doped*. The P-N junction contains two doped sides: the P-doped and the N-doped side. The N-doped side is doped with an impurity atom with five valence electrons, typically phosphorus. Four of the five valence electrons in the phosphorus atom is used in the covalent bond and the fifth is promoted to the conduction band creating a free negative charge carrier. The same thing is done on the P-side but with an atom with three valence electrons, typically boron. The covalent bond is now one electron short which attracts electrons from neighboring Si_4 covalent bonds. At room temperature an electron from a neighboring bond will always jump to repair the unsatisfied bond leaving a *hole* which is a positive charge carrier that can move

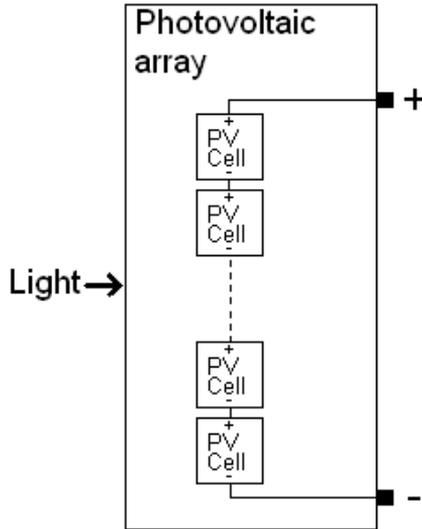


Figure 4.6. Illustration of how the photovoltaic cells are connected in series in an array.

around in the crystal and carry a current. Both the electrons and the holes can travel around freely in the crystal structure and are called *mobile charge carriers*.

When the two doped sides are put in contact with each other, there is an imbalance between the electron and hole concentrations causing them to diffuse into each others regions, following a gradient of electrochemical potential. When the electrons diffuse into the P-doped side it is very likely that it will recombine with a hole creating a negatively charged ion near the P-N interface. The same thing happens when the holes diffuse into the N-doped side and recombine with an electron, creating a positively charged ion near the P-N interface. As the recombination takes place, increasingly more ions are created near the P-N interface until an electrical potential is built up that opposes the diffusion currents. This electrically charged region is called the *depletion region* or the *space charge region*. When no external voltage source is connected, the built up voltage in the equilibrium state is called the *built in voltage*. Figure 4.7 shows the two doped sides and the space charge region in a P-N junction. The hole concentration is high in the P-doped side, low in the N-doped side and decreases exponentially in the space charge region. The electron concentration is high in the N-doped side, low in the P-doped side and decreases exponentially in the space charge region.

When the N-doped side is connected to the positive terminal of a battery and the P-doped side is connected to the negative terminal of a battery, the mobile charge carriers are pushed towards the P-N interface. When the battery is connected this way the P-N junction is *forward biased*. The positively charged terminal repels the holes in the P-doped side while the negatively charged terminal repels the electrons in the N-doped side. When this happens the width of the

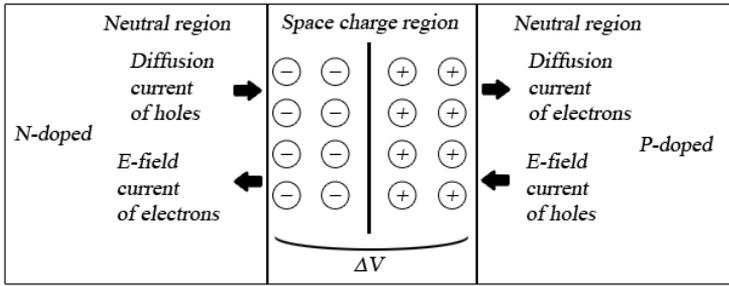


Figure 4.7. Illustration of a P-N junction in thermal equilibrium.

space charge region decreases making it easier for charge carriers to penetrate it and carry a current through it. This setup is depicted in Figure 4.8.

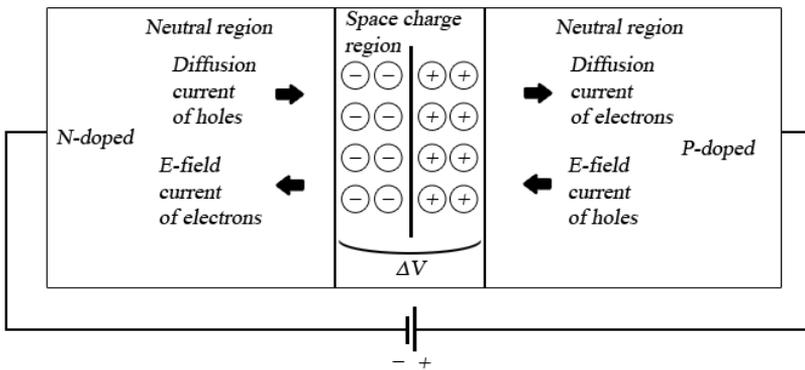


Figure 4.8. Illustration of how the P-N junction when forward biased. The width of the space charge region is decreased.

When the terminals of the battery are switched, the mobile charge carriers are pulled away from the junction which increases the width of the space charge region. When the battery is connected this way the P-N junction is *reverse biased*. When this happens the flow of charge carriers through the junction is greatly reduced. This setup is depicted in Figure 4.9.

Because of this rectifying behavior, the P-N junction will behave like a diode that promotes current to flow in only one direction across the junction. Electrons may pass from the n-type side into the p-type side, and holes may pass from the p-type side to the n-type side, but not the other way around.

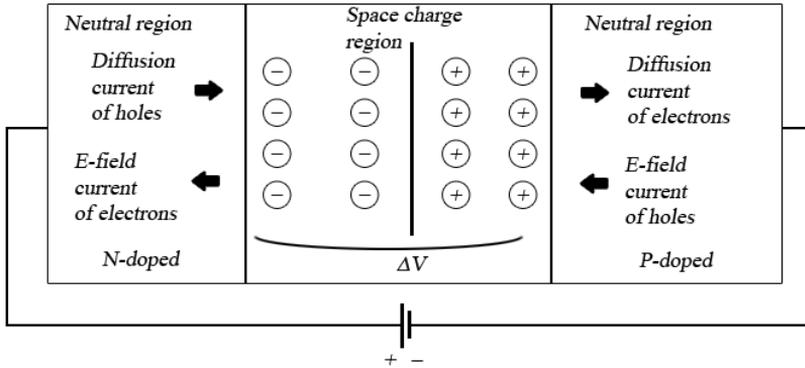


Figure 4.9. Illustration of how the P-N junction when reverse biased. The width of the space charge region is increased.

The photovoltaic cell

When a photon hits the surface of a photovoltaic cell, one out of three things happens:

1. The photon passes straight through the material. This is common for photons which carries low energy.
2. The photon is reflected.
3. The photon is absorbed by the material if the photon energy is higher than the band gap voltage E_g of the material.

When the energetic photon is absorbed by the material, its energy is given to an electron which usually is in the valence band, stuck in a covalent bonding. When this happens, the electron gets excited into the conduction band where it can move freely through the semiconductor. When the electron is removed from the valence band, the covalent bond which it was a part of is now missing an electron and a hole has been created. This way an absorbed photon creates a mobile electron-hole pair. When this happens in the N-doped side, the concentration of electrons is barely affected because of the already existing large numbers in that region. However, the generated hole is in minority and the concentration of holes in that region is greatly increased. The same thing happens when a photon creates an electron-hole pair in the P-doped side, but then the electron concentration is affected the most.

This minority carrier concentration increase affects the balance between the diffusion currents and the electrical field. The minority carrier will eventually diffuse into the space charge region where the electrical field forces it into the other side of the P-N interface. This movement of charge carriers, created by photons, carries a *photocurrent* which is the purpose of the photovoltaic cell.

Ideal model

The modeling strategy is to use a simple, electrically equivalent circuit of the photovoltaic cell. The first approach is to model the cell as an ideal diode in parallel with a current source which represents the generation of the photocurrent I_{ph} . When the cell is illuminated, the generated photocurrent is proportional to the light intensity [15] and the illuminated area. The behavior of an ideal diode is expressed in the *Shockley diode equation* seen in equation 4.11 with the ideality factor, n equal to one. The parameters used in the equation are explained below.

- q is the elementary charge
- k_B is the boltzmann constant
- I_d is the current flowing through the diode
- V_d is the voltage across the diode
- T is the diode temperature
- I_0 is the reverse saturation current and depends on the material of the cell
- n is the diode ideality factor which is equal to one for an ideal diode

$$I_d = I_0(e^{qV_d/nk_B T} - 1) \Leftrightarrow I_d = I_0(e^{qV_d/k_B T} - 1) \tag{4.11}$$

The *Shockley diode equation* and *Kirchoff's current law* gives the relation between the cell voltage, V and current I seen below.

$$I = I_{ph} - I_d \Leftrightarrow I = I_{ph} - I_0(e^{qV/k_B T} - 1) \tag{4.12}$$

This simple model represents the rectifying behavior of the P-N junction combined with the generation of photocurrent. Figure 4.10 shows a circuit diagram of the ideal model.

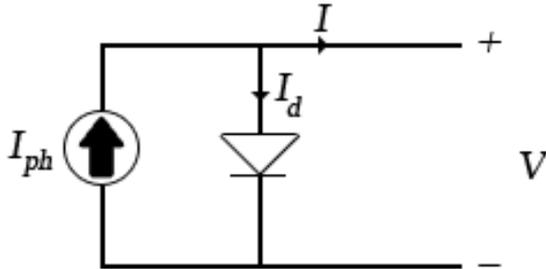


Figure 4.10. Circuit diagram of an ideal photovoltaic cell.

When the cell is not connected to a load, all the generated photocurrent is pushed through the diode creating an open circuit voltage V_{oc} between the two terminals. The size of this voltage is calculated in equation 4.13 by manipulating equation 4.12 and taking into account that the current between the terminals is zero.

$$0 = I_{ph} - I_0(e^{qV_{oc}/k_B T} - 1) \Leftrightarrow V_{oc} = \frac{k_B T}{q} \ln(I_{ph}/I_0 + 1) \quad (4.13)$$

If we introduce a short circuit between the two terminals, the voltage between the terminals is zero and the short circuit current I_{sc} will be equal to the photocurrent. This can be expressed mathematically by manipulating equation 4.12 and inserting $V = 0$ as seen below.

$$I_{sc} = I_{ph} - I_0(e^{q*0/k_B T} - 1) \Leftrightarrow I_{sc} = I_{ph} \quad (4.14)$$

As mentioned earlier, the generated photocurrent I_{ph} is proportional to the light intensity and the illuminated area. This behavior is described by equation 4.15 where L is the light intensity, A is the exposed cell area and $I_{ph,0}$ is the photocurrent under standard test conditions. The light intensity is $1000W/m^2$ and the exposed area is $0.864/72m^2$ under standard test conditions.

$$I_{ph} = I_{ph,0} \frac{L}{1000} \frac{72A}{0.864} \quad (4.15)$$

Non-ideal model

One of the goals with the model is to get the same electrical characteristics as specified by the manufacturer in Section 3.4.1. In order to achieve this goal properly, the cell model must satisfy four mathematical constraints listed below. The total power output of the cell is P , the cell current I_p and voltage V_p is the maximum power point.

1. $V = V_{oc} = 40/72 \approx 0.56$ [V] when $I = 0$
2. $I = I_{sc} = 3.75$ [A] when $V = 0$
3. $I = I_p$ when $V = V_p$
4. $max(P(I, V)) = P(I_p, V_p) = P(3.15, 32.4/72) = P(3.15, 0.45)$

The ideal model is not enough to satisfy all of the above constraints at the same time. This happens because the ideal model only has two unknown parameters: I_0 and $I_{ph,0}$. The solution to the problem is to expand the solution space by introducing more unknown parameters. There are a couple of non-ideal factors listed below that can be used to expand the model [15], [2].

1. Power loss through the resistance of the contacts.
2. Power loss through leakage currents through the cell, around the edges of the device and between contacts of different polarity.

3. Non-ideal diode behavior.

In order to get a solvable equation system, only two of the three non-ideal factors can be modeled. That way the equation system will contain four unknown parameters and four constraints which is solvable. The best result was achieved with ideal diode behavior with the first two factors included in the model. The first factor is modeled by a parasitic resistance, R_s in series with the cell. The second factor is modeled by a parasitic shunt resistance R_{sh} in parallel with the diode and the current source. The new electrical circuit diagram is depicted in Figure 4.11 where V_s and V_{sh} are the voltage drops over R_s and R_{sh} respectively.

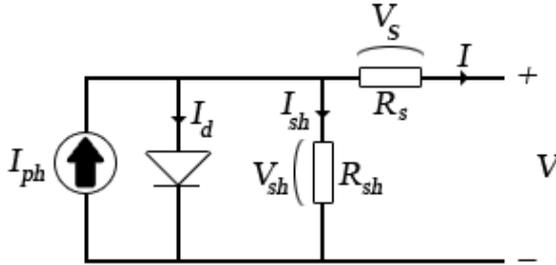


Figure 4.11. Circuit diagram of a non-ideal photovoltaic cell.

The photocurrent is now split up into three parts: current through the diode I_d , current through the shunt resistor I_{sh} and current through the series resistor I which is the cell output current. This relation is expressed in equation 4.16.

$$I_{ph} = I_d + I_{sh} + I \tag{4.16}$$

Ohm's law applied on the series resistor gives: $I_s R_s = V_s$. The potential before the current enters the series resistance becomes $V + IR_s$ which also is the voltage drops V_{sh} over R_{sh} and V_d over the diode. Ohm's law applied on the shunt resistor gives the current I_{sh} expressed in the equation below.

$$I_{sh} = V_{sh}/R_{sh} = (V + IR_s)/R_{sh} \tag{4.17}$$

The diode voltage drop $V_d = V + IR_s$ inserted in the ideal Shockley diode equation 4.11 gives the equation below.

$$I_d = I_0(e^{q(V+IR_s)/k_B T} - 1) \tag{4.18}$$

Equations 4.15, 4.17 and 4.18 gives the equation below.

$$I_{ph,0} \frac{L}{1000} \frac{72A}{0.864} = I_0(e^{q(V+IR_s)/k_B T} - 1) + \frac{V + IR_s}{R_{sh}} + I \tag{4.19}$$

The specifications from the manufacturer were produced under standard test conditions which means that the light intensity L is $1000W/m^2$, the total illuminated area A is $0.864m^2$ and the cell temperature T is $25^\circ C = T_0$. Equation

4.19 can be used to determine the four unknown parameters if the standard test conditions are taken into account. Equation 4.19 together with the standard test conditions becomes the equation below.

$$I_{ph,0} = I_0(e^{q(V+IR_s)/k_B T_0} - 1) + \frac{V + IR_s}{R_{sh}} + I \quad (4.20)$$

The first constraint inserted into equation 4.20 gives the constraint equation below.

$$I_{ph,0} = I_0(e^{q(V_{oc})/k_B T_0} - 1) + \frac{V_{oc}}{R_{sh}} \quad (4.21)$$

The second constraint inserted into equation 4.20 gives the constraint equation below.

$$I_{ph,0} = I_0(e^{q(I_{sc}R_s)/k_B T_0} - 1) + \frac{I_{sc}R_s}{R_{sh}} + I_{sc} \quad (4.22)$$

The third constraint inserted into equation 4.20 gives the constraint equation below.

$$I_{ph,0} = I_0(e^{q(V_p+I_p R_s)/k_B T_0} - 1) + \frac{V_p + I_p R_s}{R_{sh}} + I_p \quad (4.23)$$

In order to get the fourth constraint equation, an analytical expression for the maximum power point is needed. A new function $H(I, V)$ is introduced which is zero when equation 4.20 holds. $H(I, V)$ and the total power $P(I, V)$ is expressed below.

$$H(I, V) = I_0(e^{q(V+IR_s)/k_B T_0} - 1) + \frac{V}{R_{sh}} + I(1 + \frac{R_s}{R_{sh}}) - I_{ph,0} \quad (4.24)$$

$$P(I, V) = IV \quad (4.25)$$

The derivatives of $H(I, V)$ and $P(I, V)$ with respect to I and V gives the maximum power point expressed in the equation below.

$$\begin{vmatrix} \frac{\delta H}{\delta I} & \frac{\delta P}{\delta I} \\ \frac{\delta H}{\delta V} & \frac{\delta P}{\delta V} \end{vmatrix} = 0 \Leftrightarrow \frac{\delta H}{\delta I} \frac{\delta P}{\delta V} - \frac{\delta H}{\delta V} \frac{\delta P}{\delta I} = 0 \quad (4.26)$$

The four derivatives used in 4.26 are expressed in equations 4.27 to 4.30 below.

$$\frac{\delta P}{\delta V} = I \quad (4.27)$$

$$\frac{\delta P}{\delta I} = V \quad (4.28)$$

$$\frac{\delta H}{\delta V} = \frac{1}{R_{sh}} + \frac{I_0 q}{k_B T_0} e^{q(V+IR_s)/k_B T_0} \quad (4.29)$$

$$\frac{\delta H}{\delta I} = 1 + \frac{R_s}{R_{sh}} + \frac{I_0 q R_s}{k_B T_0} e^{q(V+IR_s)/k_B T_0} \quad (4.30)$$

Equations 4.27 to 4.30 inserted into equation 4.26 gives the fourth constraint equation 4.31.

$$\left(1 + \frac{R_s}{R_{sh}} + \frac{I_0 q R_s}{k_B T_0} e^{q(V+IR_s)/k_B T_0}\right) I - \left(\frac{1}{R_{sh}} + \frac{I_0 q}{k_B T_0} e^{q(V+IR_s)/k_B T_0}\right) V = 0 \quad (4.31)$$

Finally, the four constraint equations 4.21 to 4.23 and 4.31 can be used to calculate the four unknown parameters: I_0 , $I_{ph,0}$, R_s and R_{sh} . This system of equations contains several unknown parameters which occur in both polynomial and exponential terms, making an analytical solution extremely difficult if not impossible to find. Therefore the equation system was implemented in RODON which calculated the following numerical solution:

$$\begin{cases} I_0 = 1.3188 \cdot 10^{-9} A \\ I_{ph,0} = 3.7841 A \\ R_s = 9.3351 \cdot 10^{-3} \Omega \\ R_{sh} = 1.0273 \Omega \end{cases}$$

The solution is realistic. The reverse saturation current I_0 is normally very small, the photocurrent $I_{ph,0}$ is normally close to the short circuit current I_{sc} (3.75 A) and the series resistance R_s is normally small. However the shunt resistance R_{sh} is normally quite large and ideally its resistance is infinitely large. The calculated shunt resistance is not unrealistic but it is a bit smaller than expected. An explanation to this is that the somewhat small size of the shunt resistance is compensating for the non-ideal behavior of the diode which was neglected.

Given the model of one cell, it is possible to plot the current-voltage characteristics of the entire array. Figure 4.12 depicts the modeled current-voltage characteristics at different light intensities. It is also possible to plot the total power output of the array against the voltage and current, see Figures 4.13 and 4.14.

Temperature dependence

In Section 3.4.1 it is mentioned that a temperature sensor is attached to the solar cell. This makes it interesting to model how the solar panel is affected by the temperature. In the non-ideal model the temperature only occurs in the exponential term of the diode behavior. However, that single temperature dependence is not enough to satisfy the specified temperature behavior given by the manufacturer in Section 3.4.1. The temperature effect on the photocurrent is approximated to be equal to its effect on the short circuit current: $0.09\%/^{\circ}C$. What this approximation actually does is to neglect the small series resistance with respect to the temperature dependence. Equation 4.32 shows the linear temperature dependence of the photocurrent.

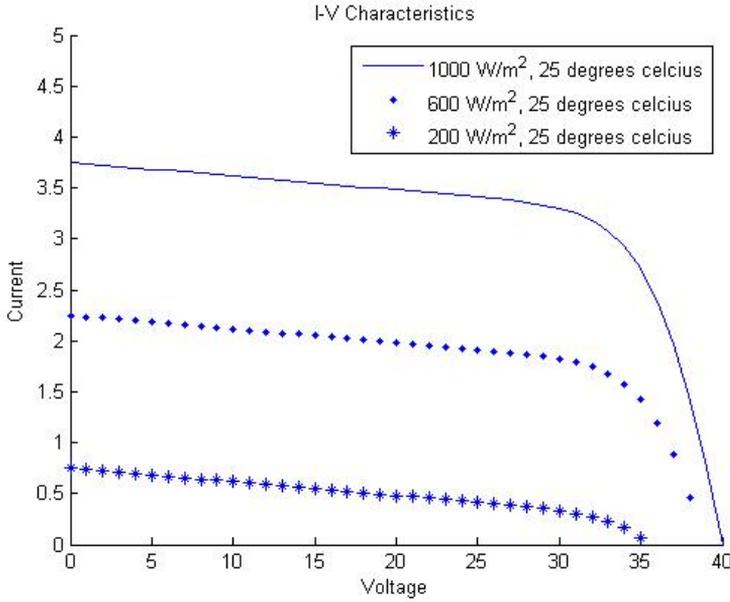


Figure 4.12. The effects of light intensity on the current-voltage characteristics of the entire array.

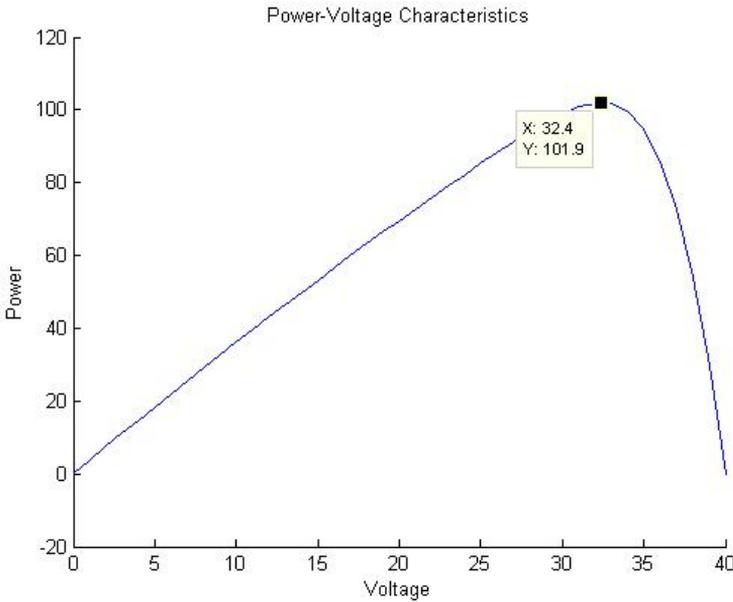


Figure 4.13. The total power output of the solar array plotted against the voltage under standard test conditions. Notice the maximum power point of 32.4 Volts.

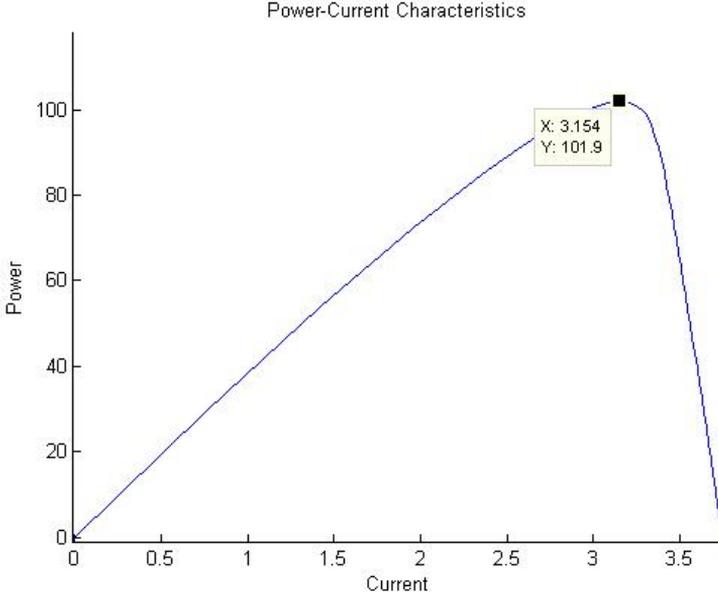


Figure 4.14. The total power output of the solar array plotted against the current under standard test conditions. Notice the maximum power point of 3.15 Amperé.

$$I_{ph} = I_{ph,0} \frac{L}{1000} \frac{72A}{0.864} \left(1 + (T - T_0) \frac{0.09}{100}\right) \quad (4.32)$$

Another temperature dependent parameter is the reverse saturation current I_0 . It is proportional to the intrinsic carrier concentration n_i squared. The temperature dependence of the intrinsic carrier concentration dominates in the reverse saturation current. The interested reader can find an extensive derivation of this relation in [2]. The temperature dependence of the intrinsic carrier concentration is expressed in the equation below where E_g is the band gap voltage of silicon and β is a constant.

$$n_i = \beta T^{3/2} e^{-E_g/(2k_B T)} \quad (4.33)$$

The interested reader can find an extensive derivation of this relation in [2] or [22]. The temperature dependence of the reverse saturation current is expressed in the equation below where γ and C are constants.

$$I_0 = \gamma n_i^2 = \gamma (\beta T^{3/2} e^{-E_g/(2k_B T)})^2 = C T^3 e^{-E_g/(k_B T)} \quad (4.34)$$

The constant C can be determined by the the following constraint: the calculated value of I_0 is $1.3188 \cdot 10^{-9} A = I_{0,0}$ at standard test conditions. The constant C is determined in the equation below.

$$I_0(T_0) = I_{0,0} \Leftrightarrow I_{0,0} = CT_0^3 e^{-E_g/(k_B T_0)} \Leftrightarrow C = \frac{I_{0,0}}{T_0^3 e^{-E_g/(k_B T_0)}} \quad (4.35)$$

The expression for C in equation 4.35 inserted into equation 4.36 gives the final temperature dependence of the reverse saturation current. This is expressed in the equation below.

$$I_0 = \frac{I_{0,0}}{T_0^3 e^{-E_g/(k_B T_0)}} T^3 e^{-E_g/(k_B T)} = I_{0,0} \left(\frac{T}{T_0}\right)^3 e^{-E_g/k_B(1/T-1/T_0)} \quad (4.36)$$

The temperature effect on the current-voltage characteristics of the entire array is depicted in Figure 4.15. The modeled temperature effect on the short circuit current, open circuit voltage and the maximum power is depicted in Figures 4.16, 4.17 and 4.18. The modeled short circuit current increases 0.09 %/°C until the temperature reaches 120 °C and the linear behavior stops. The modeled open circuit voltage decreases 0.36 to 0.38 %/°C which is close to the specified value of 0.41 %/°C. The modeled maximum power decreases 0.36 to 0.38 %/°C which is relatively close to the specified value of 0.49 %/°C.

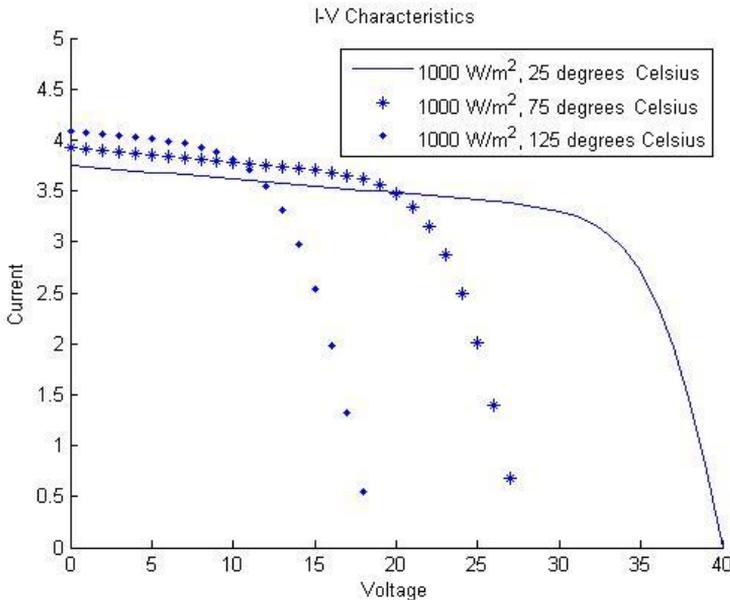


Figure 4.15. The temperature effects on the current-voltage characteristics of the entire array.

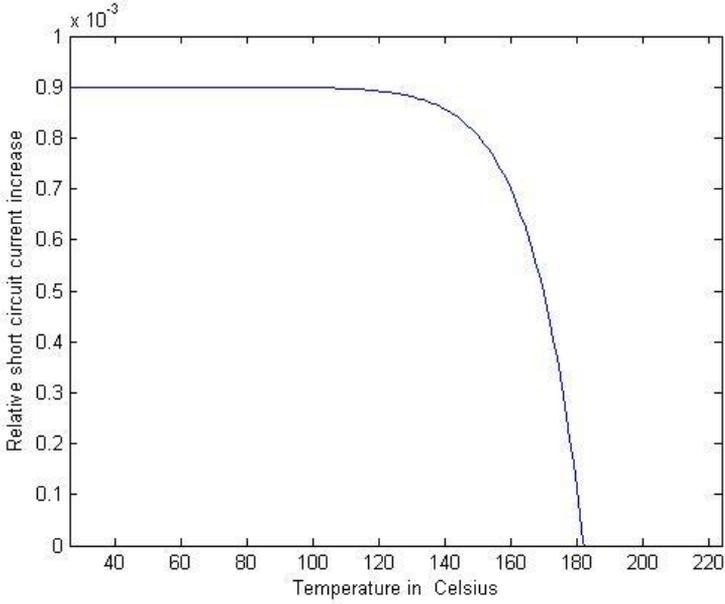


Figure 4.16. The temperature effects on the short circuit current. The specified value from the manufacturer is $0.09\%/^{\circ}C$.

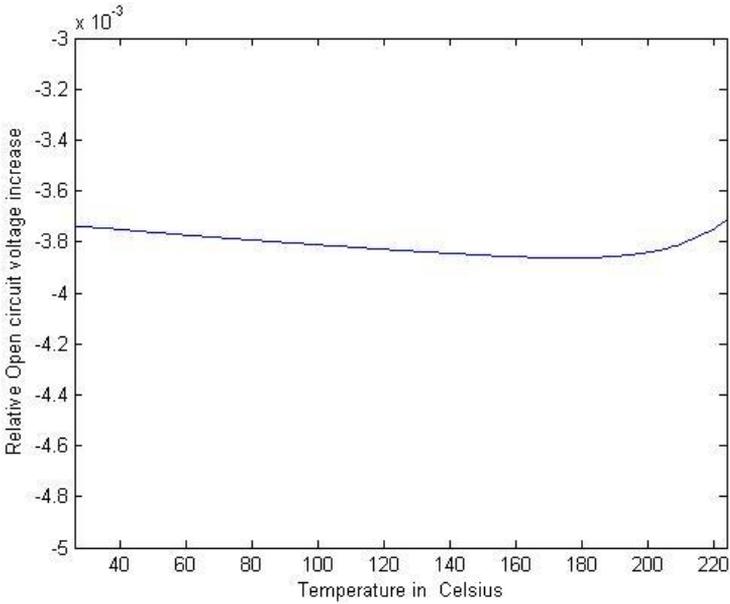


Figure 4.17. The temperature effects on the open circuit voltage. The specified value from the manufacturer is $-0.41\%/^{\circ}C$.

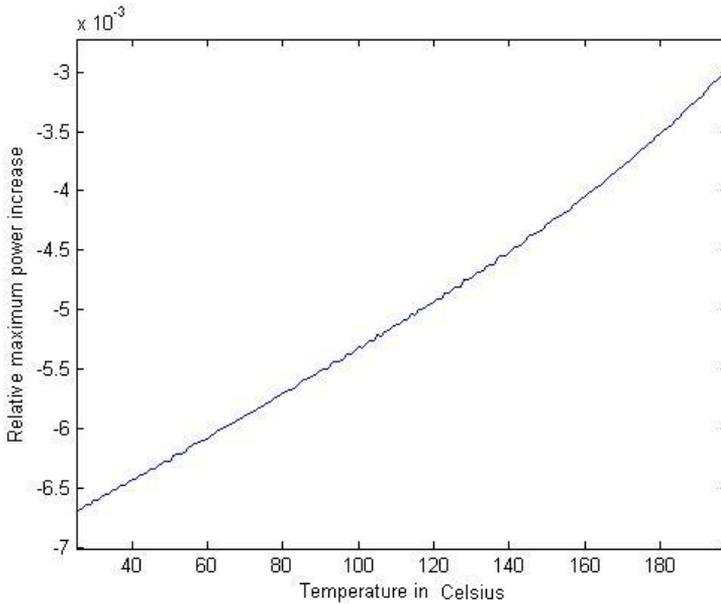


Figure 4.18. The temperature effects on the maximum power. The specified value from the manufacturer is $-0.49\ \%/^{\circ}\text{C}$.

4.1.10 Inverter

The inverter converts 24 V DC into 120 V AC with a frequency of 60 Hz. The AC voltage is generated with an accuracy of $+4\% / -10\%$ and the frequency with an accuracy of $\pm 0.05\ \text{Hz}$ [6]. If the supplied DC voltage is below 20 V or above 32 V it will shut down. The inverter model is very similar to the battery charger model described in Section 4.1.5. The model has been separated into two sides: the DC and the AC side. If the DC voltage is outside the range of 20 to 32 V the battery will be "off", otherwise "on". When the inverter is off, there is no current going in or out of the inverter and there is no AC voltage generated. The model contains three failure modes: "disconnected", "crystal failure" and "unknown". When the inverter is disconnected, it is off independently of the DC voltage. When the AC output frequency is outside the nominal range, something is wrong with the crystal that is used to control it. When this happens, the inverter has a crystal failure. The unknown failure mode is used to catch faults that have not been anticipated.

4.1.11 Loads

All scenario loads has been modeled as resistors. If the load has some kind of output, it is proportional to the power consumption, i.e the pumped water by "Pump 1" is proportional to the consumed power. Eight of the connected lamps are monitored by temperature sensors. Initially, the lamp models had a very similar temperature model as the accumulator model, seen in equation 4.37. However,

when the lamp is suddenly turned off the bulb temperature is high despite that no power is consumed. This would not cause any problems if the bulbs were allowed to cool down before diagnosis. Unfortunately, that was not the case in the experiments done by NASA and diagnosis had to be made in a small time window before the operator turned on the backup loads. Therefore the temperature model of the lamps had to be disabled.

$$T_{env} \leq T_{bulb} \leq T_{env} + \dot{Q}/\alpha \tag{4.37}$$

4.2 RODON implementation

The physical behavior of each component has been implemented in classes. The model is built up of instances of these classes which support reusable components. Equations are programmed into these classes in Rodelica directly. This simple way of programming promotes a fast and efficient model implementation. All the components described in Section 4.1 are implemented directly as described, except for the solar panel and the battery charger-accumulator interaction. The entire testbed model would be significantly slowed down if the 72 photovoltaic cells had been implemented as 72 separate components. In order to make the model run more smoothly, an equivalent circuit was implemented with the same physical behavior but with faster computational times. The 72 cells were replaced by one big cell. The difference between a small and a big cell is the scaling factor of 72 inserted at suitable locations. The resulting big cell model is expressed in the equations below.

$$\left\{ \begin{array}{l} R_{s,big} = 72R_s \\ R_{sh,big} = 72R_{sh} \\ I_{0,big} = I_{0,0} \left(\frac{T}{T_0}\right)^3 e^{-E_g/k_B(1/T-1/T_0)} \\ I_{d,big} = I_{0,big} (e^{q(V+IR_{s,big})/72k_B T} - 1) \\ I_{ph,big} = I_{ph,0} \frac{L}{1000} \frac{A}{0.864} (1 + (T - T_0) \frac{0.09}{100}) \\ I_{sh,big} = (V + IR_{s,big})/R_{sh,big} \\ I_{ph,big} = I_{d,big} + I_{sh,big} + I \end{array} \right.$$

The interaction between the chargers and the accumulators had to be implemented at a higher level. This was necessary because the charger is in the failure mode "overcharging" if the battery temperature is too high. This means that the overcharging is impossible to detect by just looking at the charger.

4.2.1 NASA Library

All the classes of different components and subsystems make up the NASA library. The library has been separated into two parts: *components* and *systems*. The *components* part contains all the lowest level building blocks and the *systems* part contains all the high level structures comprised of instances from the *components* part. This library can be reused in other similar projects. A screenshot of the NASA-library is depicted in Figure 4.19.

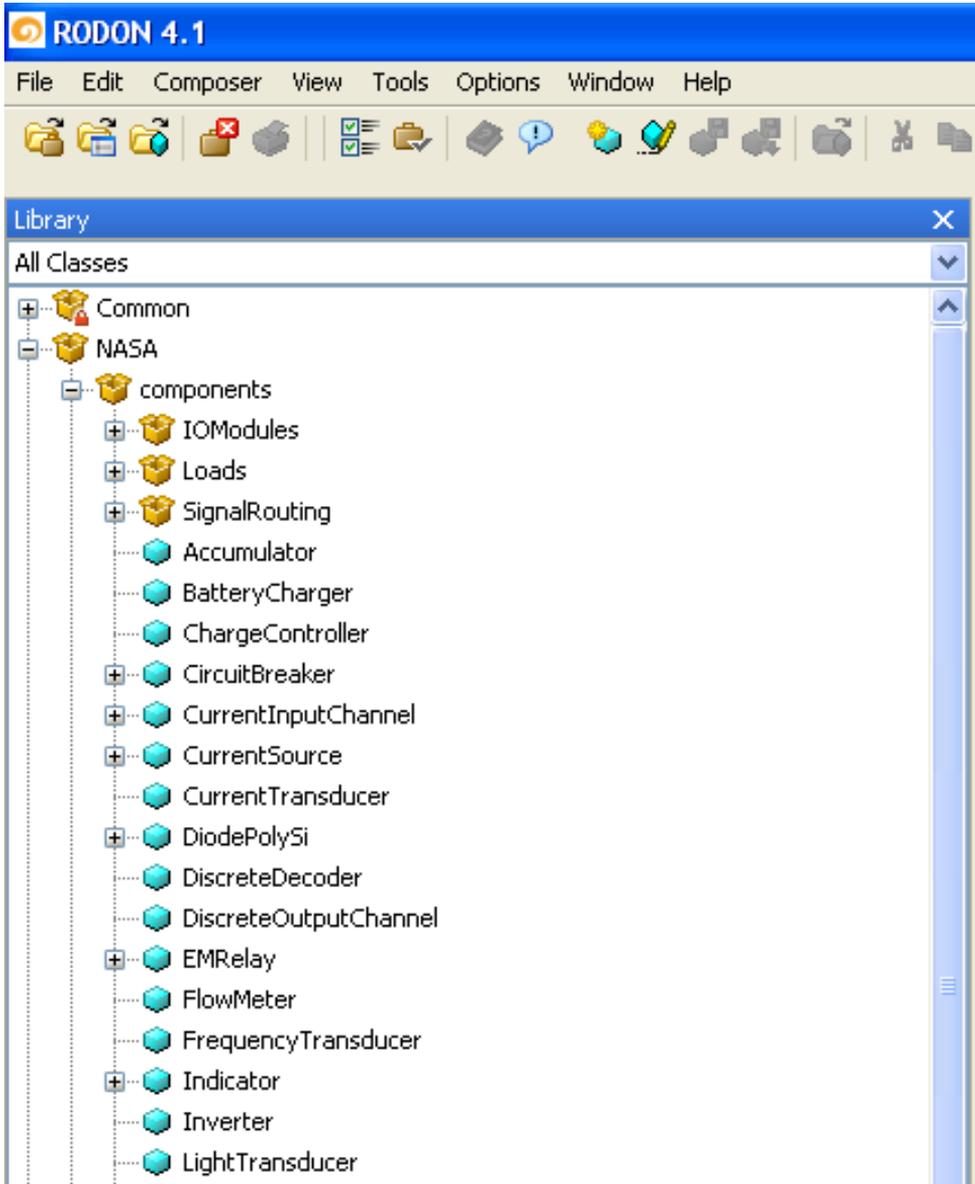


Figure 4.19. Screenshot of the NASA library in the RODON composer.

4.2.2 Model structure

The modeling strategy was to make the model easy to understand and in many ways self-explanatory. All low level components used in the model are located in a subsystem of some kind. The subsystems have a picture of what the actual hardware looks like. This way, an operator can relate RODON candidates with the actual hardware in the lab (or space shuttle). The hierarchical model structure of the EPS and the advanced caution and warning system (ACAWS) is depicted in Figures 4.20 and 4.21. The loads and the load sensors have been modeled in a separate subsystem outside the EPS model. This makes the model easy to update if a new set of loads are connected.

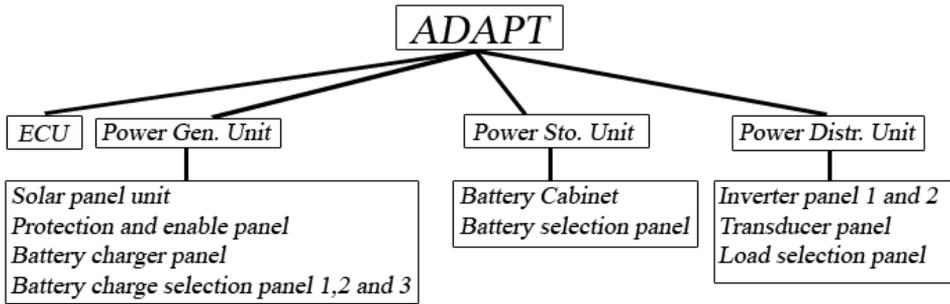


Figure 4.20. The ADAPT model structure. The lowest level is not included.

4.2.3 Model-based diagnosis of the testbed

Diagnosis is performed on the testbed in the same way as described in Section 2.2.2. The model contains 99 sensors and 39 relays which can make it time-consuming to insert sensor measurements and relay commands. To automate this, all I/O signals have been organized in a list in the model where the observations are inserted. A script is used to automatically insert experiment data from NASA into the list. The list is located in an object called DAC (Data Acquisition Computer). The observations are propagated from the DAC into an object called ECU (Electronic Control Unit). The ECU represents the Compact Fieldpoint backplanes where the I/O hardware and the real-time controllers are located. From there, the measurements are propagated into the EPS. The signal flow during diagnosis is depicted in Figure 4.22. Sensor measurements of the loads are inserted in the same way into an object called "LoadMonitoring". From there the signals are propagated into a separate ECU object where the load sensors are connected. The process of inserting observations can of course be automated even further enabling on-board monitoring of the testbed.

When the observations are inconsistent with the nominal model, conflicts will arise and a set of candidates will be created. These candidates are highlighted in the RODON model making it easy to find faulty low level components. The

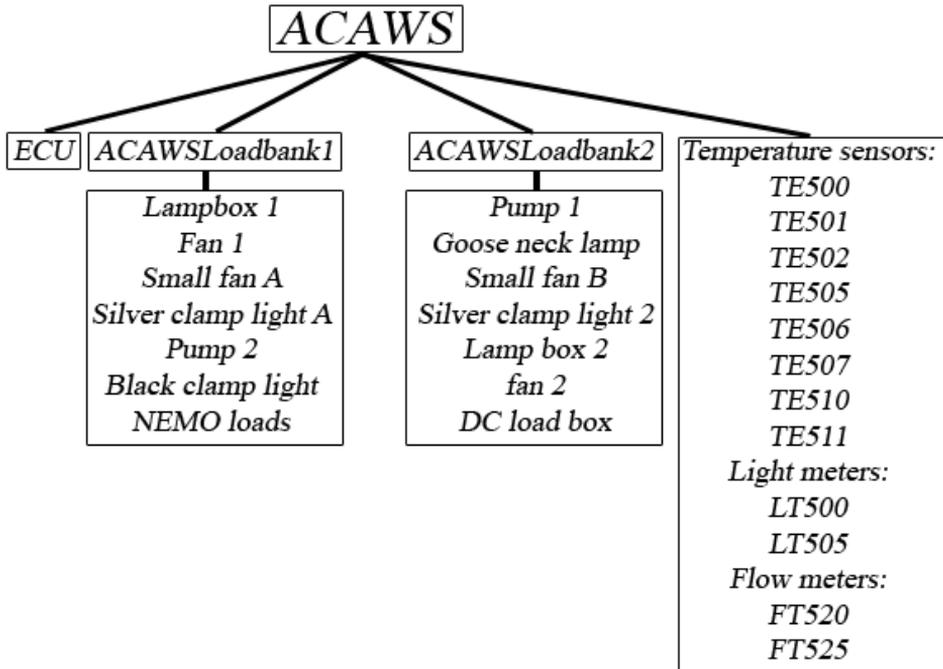


Figure 4.21. The ACAWS model structure.

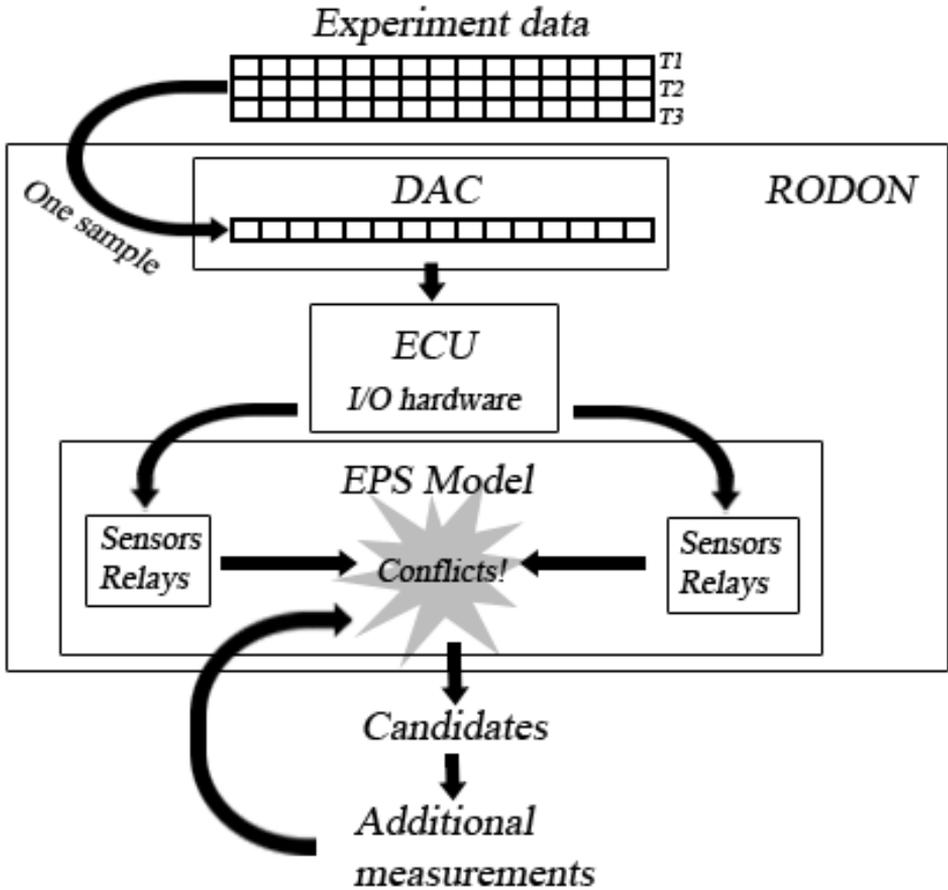


Figure 4.22. The signal flow during diagnosis. Observations are propagated through the different components of the model and will cause conflicts if the system is faulty.

number of candidates can be reduced interactively by inserting additional measurements. Section 5.3 explains how this is done. Figures 4.23 to 4.26 depicts how a low level candidate can be located from the top level in the RODON model. In this example the candidate is a disconnected inverter.

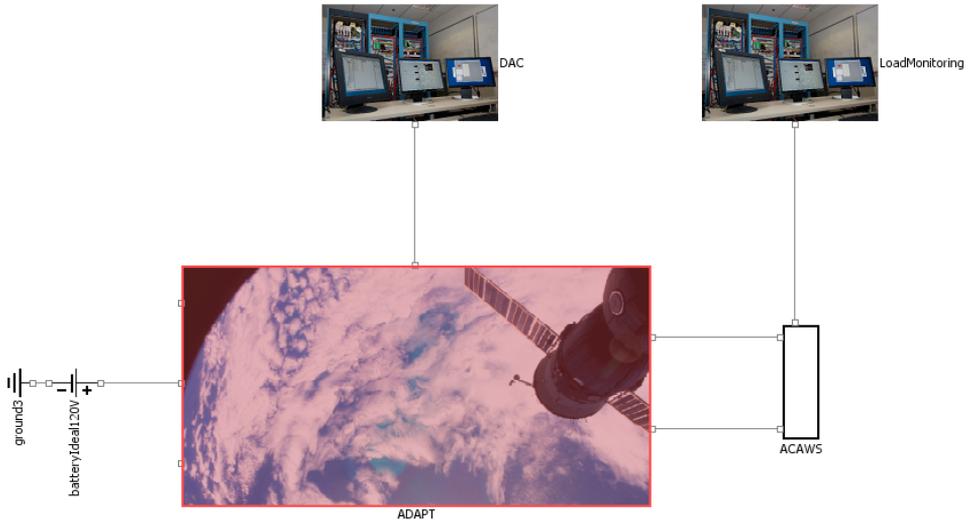


Figure 4.23. The top level view of the RODON model. ADAPT is faulty and highlighted.

4.2.4 Generation of decision trees

The model has been used to create a state data base as explained in Section 2.2. The vast number of system configurations makes it necessary to create the data base for a limited number of operational states. In this case, the state data base has been created for the nominal configuration in the ACAWS scenarios. The state data base has been used to generate decision trees based on a couple of symptoms. In this case the symptoms are the observed behavior in the ACAWS scenarios described in Section 3.5.3. An unpowered AC load results in i.e. a fan stops rotating or a lamp stops shining. However, the DC loads show no obvious sign that they are not powered, that is why the symptom "current sensor IT180 reports 0A" is used. The created symptoms are listed in Figure 4.27.

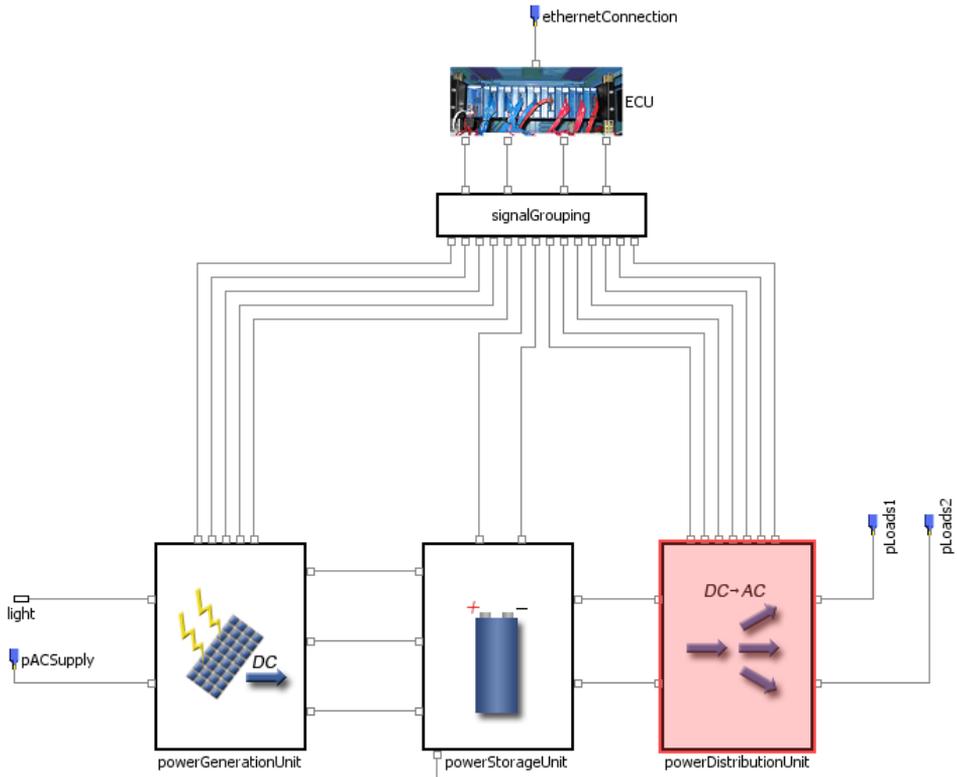


Figure 4.24. The ADAPT level view of the RODON model. The Power Generation Unit is faulty and highlighted.

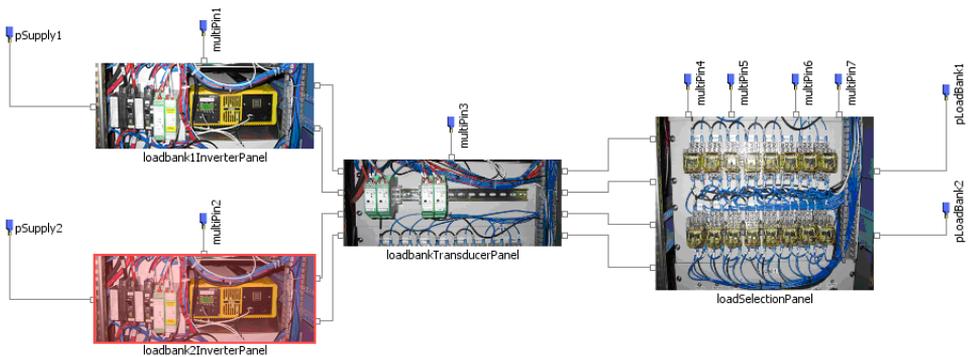


Figure 4.25. The Power Generation Unit level view of the RODON model. The second inverter panel is faulty and highlighted.

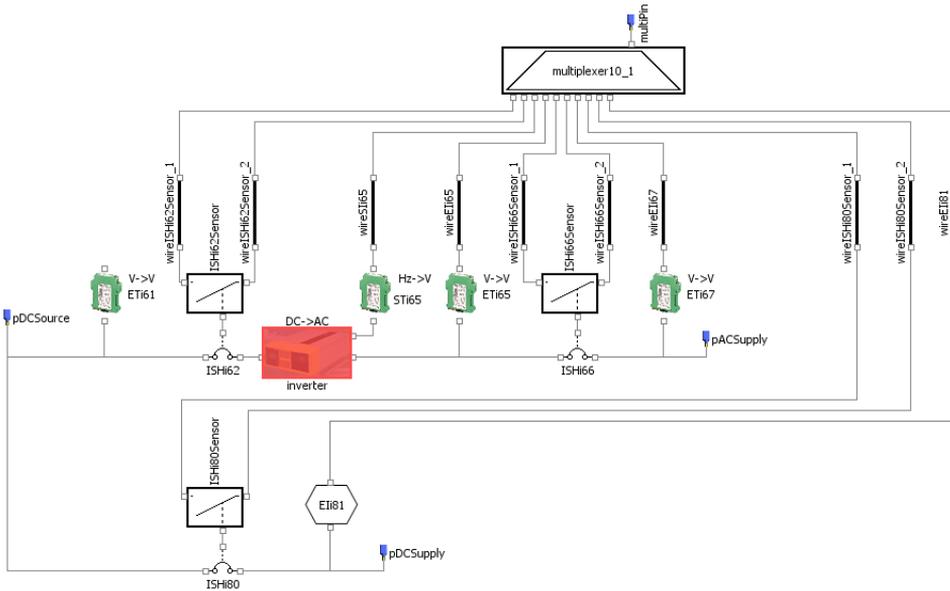


Figure 4.26. The lowest level view of the RODON model. The inverter is faulty and highlighted.

Scenario	Symptoms
Scenario 1 a	Lamp box 1 off
Scenario 1 b	Position sensor ESH271 reports "open"
Scenario 1 c fault1	Load bank 2 AC loads off
Scenario 1 c fault2	Position sensor ESH171 reports "open"
Scenario 1 d fault1	Load bank 1 AC loads off
Scenario 1 d fault2	Pump 1 off
Scenario 2 a	Load bank 1 AC loads off
Scenario 2 b	Load bank 2 AC loads off
Scenario 3 a	Load bank 1 AC loads off
Scenario 3 b	Load bank 2 AC loads off
Scenario 4 a	Load bank 1 AC loads off
Scenario 4 b	Load bank 2 AC loads off
Scenario 5 a fault2	Load bank 2 AC loads off
Scenario 5 b fault1	Load bank 1 AC loads off
Scenario 6 a fault1	Load bank 1 AC loads off
Scenario 6 a fault1 and 2	Load bank 1 AC loads off
Scenario 6 b fault1	Current meter IT181 reports 0A
Scenario 6 b fault1 and 2	Load bank 1 AC loads off

Figure 4.27. Table of created symptoms for the ACAWS scenarios.

Chapter 5

Results and discussion

As mentioned in Section 2.2, a RODON model can be used in several ways. In this project, it has been used to generate decision trees and perform model-based diagnosis on experiment data provided by NASA. The produced results are presented and discussed in this chapter.

5.1 Model-based diagnosis

Model-based diagnosis is performed on sampled experiment data from the ACAWS scenarios. As mentioned in Section 4.2.3, only data from one time instance is used. The used data sample is taken after the fault(s) have been injected and before the operator has taken action. The data is taken when most of the transient behavior has settled down. The generated candidates and the calculation times are presented in Figures 5.1 and 5.2.

5.1.1 Discussion

The injected fault is found among the generated candidates in all cases except in the first fault in scenario 5a and the second fault in scenario 5b. The first fault in scenario 5a is very similar to the second fault in scenario 5b: in both cases a fault is injected into a temperature sensor which is monitoring the bulb temperature of a connected lamp. The reason why this kind of fault is undetectable is because the temperature model of the connected lamps have been disabled for practical reasons explained in Section 4.1.11. A comforting thought is that the temperature sensor failure does not directly affect the function of the EPS.

In scenario 3b, a fault is injected into the second battery pack by spoofing sensor values. Two voltage sensors are spoofed into reporting zero volts. A third and a fourth voltage sensor is measuring the same voltage but is not spoofed. This causes unwanted conflicts between the four measured voltages. In this case, the software fault injection procedure is insufficient and the model is correct. A schematic showing the location of the four voltage meters is depicted in Figure 5.3 and a plot of the reported sensor values is depicted in Figure 5.4. When RODON

Scenario	Injected faults	Candidates	Calc. Time
1a	EY170 stuckOpen	EY170 stuckOpen wireEC170_1 disconnected, wireEC170_1 short_to_gnd, wireEC170_2 disconnected,	2.9 s
1b	ESH271 disconnected	ESH271 disconnected wireESH271_1 disconnected, wireESH271_1 short_to_gnd, wireESH271_2 disconnected, wireESH271_2 short_to_gnd,	3.6 s
1c	EY260 stuckOpen	EY260 stuckOpen wireEC260_1 disconnected, wireEC260_1 short_to_gnd, wireEC260_2 disconnected,	2.9 s
	ESH171 disconnected	ESH171 disconnected wireESH171_1 disconnected, wireESH171_1 short_to_gnd, wireESH171_2 disconnected, wireESH171_2 short_to_gnd,	3.1 s
1d	EY160 stuckOpen	EY160 stuckOpen wireEC160_1 disconnected, wireEC160_1 short_to_gnd, wireEC160_2 disconnected,	2.9 s
	EY270 stuckOpen	EY270 stuckOpen wireEC270_1 disconnected, wireEC270_1 short_to_gnd, wireEC270_2 disconnected,	3.4 s

Figure 5.1. Diagnosis results from ACAWS scenarios 1a to 1d. The injected fault(s) is seen to the left and generated candidates are seen to the right. The correct generated candidate is written in bold.

2a	EY141 stuckOpen	EY141 stuckOpen wireEC141_1 disconnected, wireEC141_1 short_to_gnd, wireEC141_2 disconnected,	3.3 s
3a	BattA disconnected	Batt1A disconnected, Batt1A damagedCell, Batt1B disconnected, Batt1B damagedCell, batteryCabinet.ground disconnected,	3.8 s
3b	BattB disconnected	Batt2A disconnected & E235 short to ground & E240 short to ground	1 min 57 s
4a	InvA disconnected	INV1 disconnected, INV1 unknown,	3.1 s
4b	InvB disconnected	INV2 disconnected, INV2 unknown,	5.2 s
5a	TE500 pinShort InvB disconnected	Need dynamic simulation INV2 disconnected, INV2 unknown,	3.6 s
5b	InvB disconnected TE502 pinShort	INV2 disconnected, INV2 unknown, Need dynamic simulation	4.4 s
6a	InvA disconnected	INV1 disconnected, INV1 unknown,	6.6 s
	InvA disconnected & & ISH180 disconnected	InvA disconnected & & ISH180 disconnected	2.8 s
6b	ISH180 disconnected	ISH180 disconnected	2.4 s
	ISH180 disconnected & & InvA disconnected	InvA disconnected & ISH180 disconnected	3.1 s

Figure 5.2. Diagnosis results from ACAWS scenarios 2a to 6b. The injected fault(s) is seen to the left and generated candidates are seen to the right. The correct generated candidate is written in bold.

uses data from this scenario a triple fault candidate is generated after 1 minute and 57 seconds. The generated candidate is quite interesting since it contains the two spoofed sensors in addition to the correct candidate. In this case, a fault in the software injection procedure was detected in addition to the injected fault.

When a fault is injected into a relay or relay position sensor, the model can not discriminate between the correct candidate and the wires leading to it from the I/O hardware. This is caused by the lack of sensors monitoring those wires. The wire candidates can be excluded interactively by performing additional measurements on them. A model without I/O hardware and wires would achieve perfect isolation in those experiments. However, such a model would be insufficient if a wire would actually fail. When "BattA disconnected" is injected, a number of candidates are generated. All of them are related to the correct battery pack. A model with less depth and one big component for the entire battery pack would achieve perfect isolation in the experiments. The reason to keep the model depth was the temperature sensors attached to each battery inside the battery packs. When "inverter disconnected" is injected, the model can not discriminate between the two failure modes: "disconnected" and "unknown". Either way the correct failed component is detected. As mentioned in Section 4.1.10, the "unknown" failure mode was modeled to catch unanticipated failures. The calculation times exceeds the time between each sample (0.5 s) which is not ideal for online monitoring. However, the model can be used by RODON to generate diagnostics rules which can be used by a small and fast application with much shorter calculation times.

In summary, all injected faults except faults in temperature sensors would be isolated with less model depth. The calculation times would also be decreased. However, such a model might be insufficient if a failure not defined in the experiments would occur. That would not be desirable in a real satellite system. A dynamic model is needed to detect failures in the temperature sensors but those do not affect the actual function of the EPS.

Conflict when BattB disconnected is injected

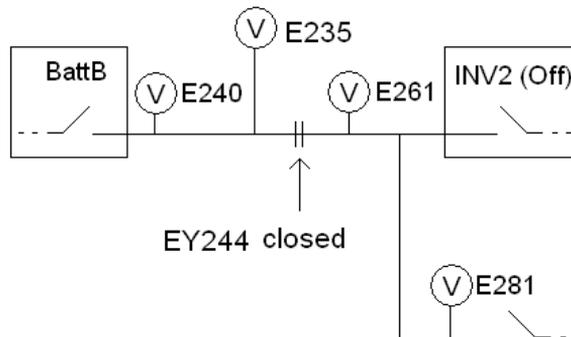


Figure 5.3. A schematic showing the location of the four voltage meters. Only two are spoofed when all four are measuring the same voltage.

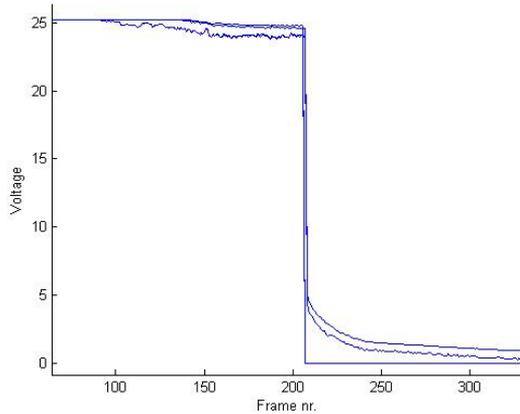


Figure 5.4. A plot of the reported values from the four voltage meters. Only two are spoofed when all four are measuring the same voltage. The four graphs should be close to each other but that is not the case here.

5.2 Decision trees

Decision trees has been generated based on symptoms from the ACAWS scenarios described in Section 4.2.4. Some of the presented trees had to be limited to only contain branches investigated by the operator. This was done because of the size of the trees. The generated decision trees are not static. The user can choose to perform alternative measurements which change the structure of the tree. The presented tree in most of the scenarios is one out of several possible trees. All of the alternative trees lead to the same results but with alternative measurements, i.e. the user can choose to inspect the light emittance of a lamp by looking at a light sensor or by directly checking the brightness of the lamp. This is further explained below. The user can specify which types of measurements are possible.

5.2.1 Scenario 1a

In this scenario relay EY170 has failed open and the symptom used is that the three lamps inside the first lamp box are off. The symptom can be changed into light sensor LT500 reports that the three lamps inside the first lamp box are off. The user is asked to inspect relay position sensor ESH170 (which reports "open/tripped"). If it is not possible to perform measurements on wires, the final leaf contains the correct candidate and the wires leading to the failed relay. Figure 5.5 depicts how a user can find the correct candidate. If those measurements are possible, the user can specify that and another tree depicted in Figure 5.6 is automatically created.

The user can also choose to perform alternative measurements which change the structure of the tree. A screenshot of the tree in RODON and the list of alternative measurements for the first node in scenario 1a is depicted in Figure

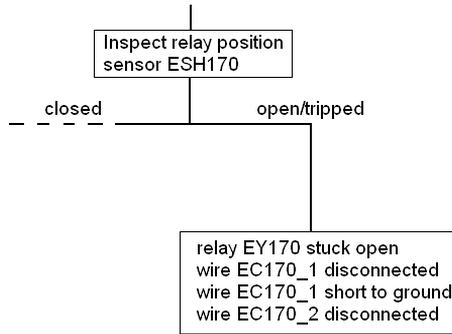


Figure 5.5. Decision tree created from the symptom: the three lamps inside the first lamp box are not shining. This tree is used for scenario 1a.

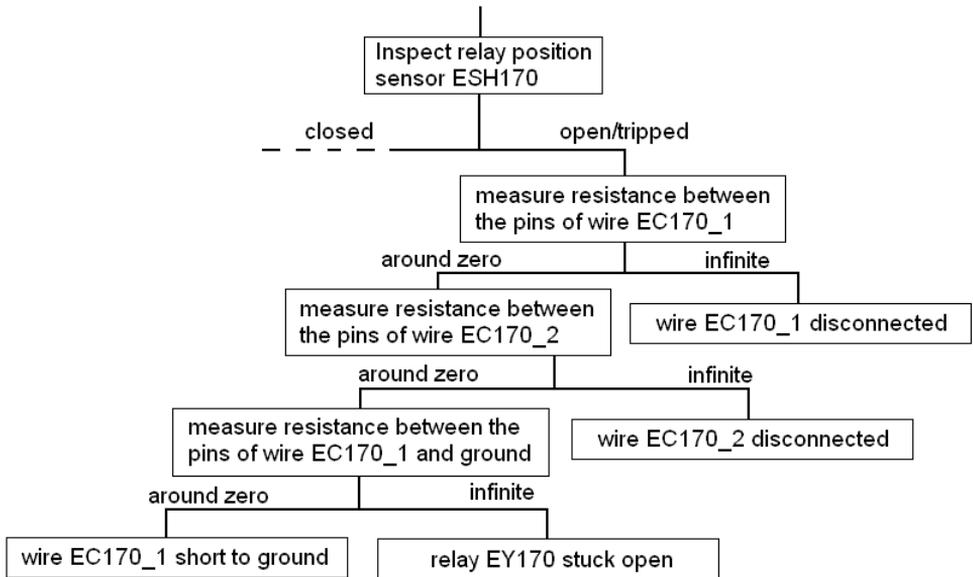


Figure 5.6. Decision tree created from the symptom: the three lamps inside the first lamp box are not shining. Measurements are allowed on the wires. This tree is used for scenario 1a.

5.7. The list is quite long which is caused by the large amounts of sensors in the testbed. The presented tree in most of the scenarios is one out of several possible trees. All of the alternative trees lead to the same results but with alternative measurements. RODON supports the operating mode to be changed interactively to further isolate the faulty candidate. The large number of sensors in the testbed made such operation mode changes unnecessary.

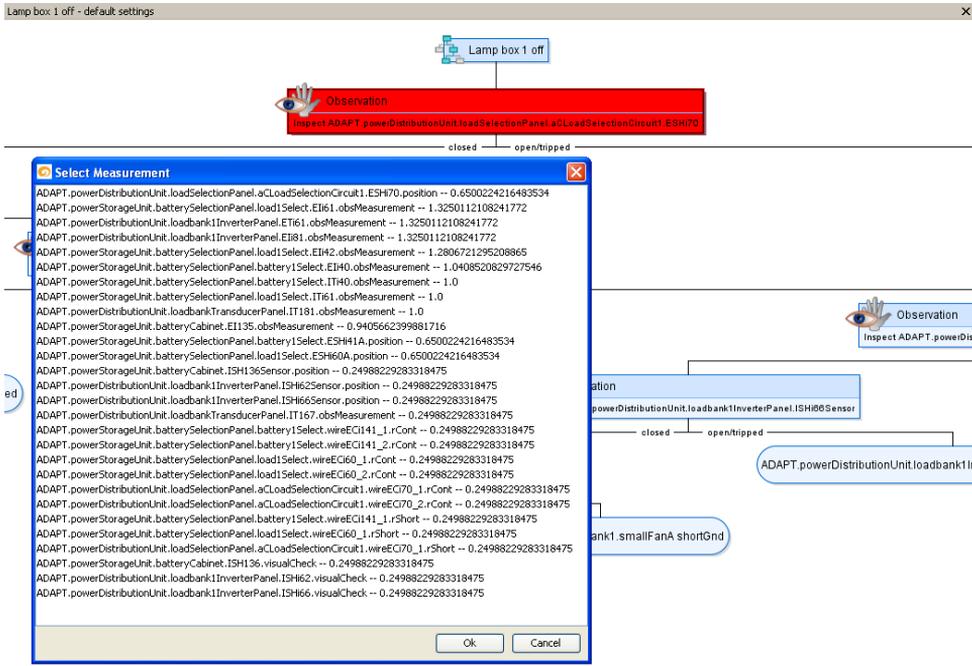


Figure 5.7. A screenshot of the graphical user interface showing possible alternative measurements for the first node.

5.2.2 Scenario 1b

In this scenario relay position sensor ESH271 is disconnected and the symptom is that it is reporting value "open/tripped". The user is asked to check if the connected lamp is bright or off. In this case the lamp is shining bright since the relay is not faulty. The isolated leaf in the tree is relay position sensor ESH271 disconnected which is the correct candidate. Figure 5.8 depicts how a user can isolate the correct candidate.

5.2.3 Scenario 1c fault 1

In this case relay EY260A has failed open and the symptom is loss of the entire second load bank. The user is asked to inspect relay position sensor ESH260A (which reports "open/tripped"). The final leaf contains the correct candidate and

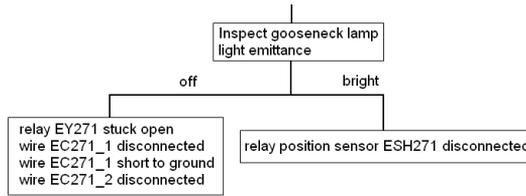


Figure 5.8. Decision tree created from the symptom: position sensor ESH271 reports "open/tripped". This tree is used for scenario 1b.

the wires leading to the failed relay. Additional measurements on the wires are needed for complete isolation. Figure 5.9 depicts how a user can find the correct candidate.

5.2.4 Scenario 1c fault 2

In this case relay position sensor ESH171 is disconnected and the symptom is that it is reporting value "open/tripped". The user is asked to check if the connected fan is rotating or not. In this case the fan is rotating since the relay is not faulty. The isolated leaf in the tree is relay position sensor ESH171 disconnected which is the correct candidate. Figure 5.10 depicts how a user can isolate the correct candidate.

5.2.5 Scenario 1d fault 1

In this case relay EY160 has failed open and the symptom is loss of the entire first load bank. The user is asked to inspect relay position sensor ESH160A (which reports "open/tripped"). The final leaf contains the correct candidate and the wires leading to the failed relay. Additional measurements on the wires are needed for complete isolation. Figure 5.11 depicts how a user can find the correct candidate.

5.2.6 Scenario 1d fault 2

In this case relay EY270 has failed open and the symptom is that the connected load, pump 1 is off. The user is asked to inspect relay position sensor ESH270 (which reports "open/tripped"). The final leaf contains the correct candidate and the wires leading to the failed relay. Additional measurements on the wires are needed for complete isolation. Figure 5.12 depicts how a user can isolate the correct candidate.

5.2.7 Scenario 2a

In this case relay EY141 has failed open and the symptom is loss of the entire first load bank. The user is asked to inspect relay position sensor ESH160A (which reports "closed") and then ESH141A (which reports "open/tripped"). The final

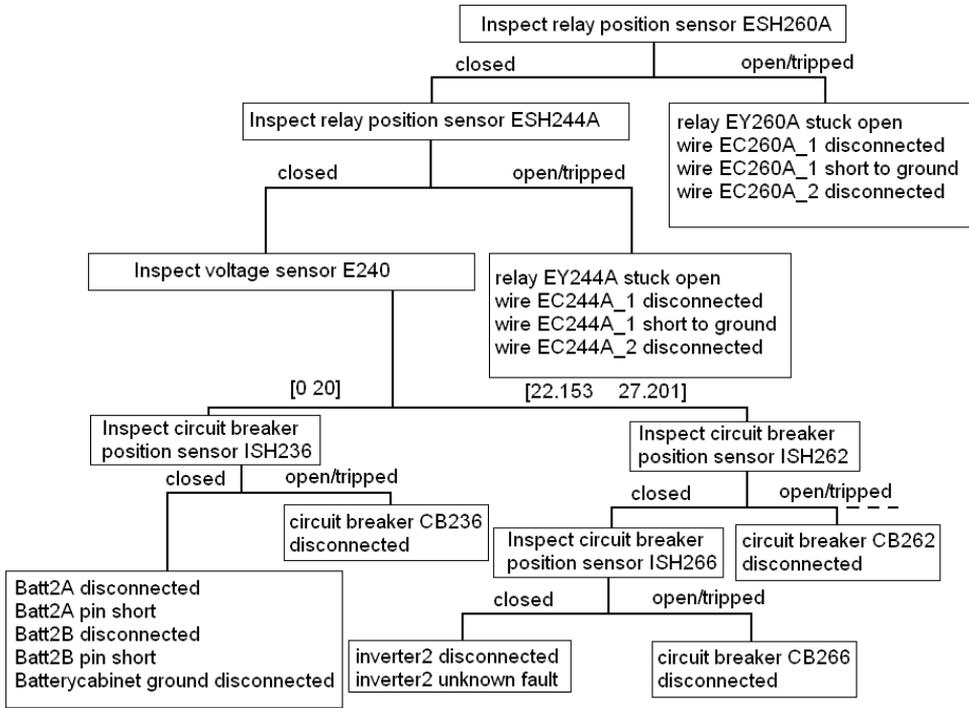


Figure 5.9. Decision tree created from the symptom: AC loads connected to the second load bank off. This tree is used for the first fault in scenario 1c, scenario 2b, 3b, 4b and the second fault in scenario 5a.

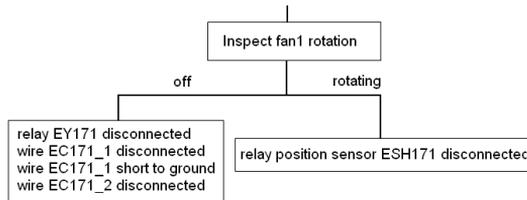


Figure 5.10. Decision tree created from the symptom: position sensor ESH171 is reporting value "open/tripped". This tree is used for the second fault in scenario 1c.

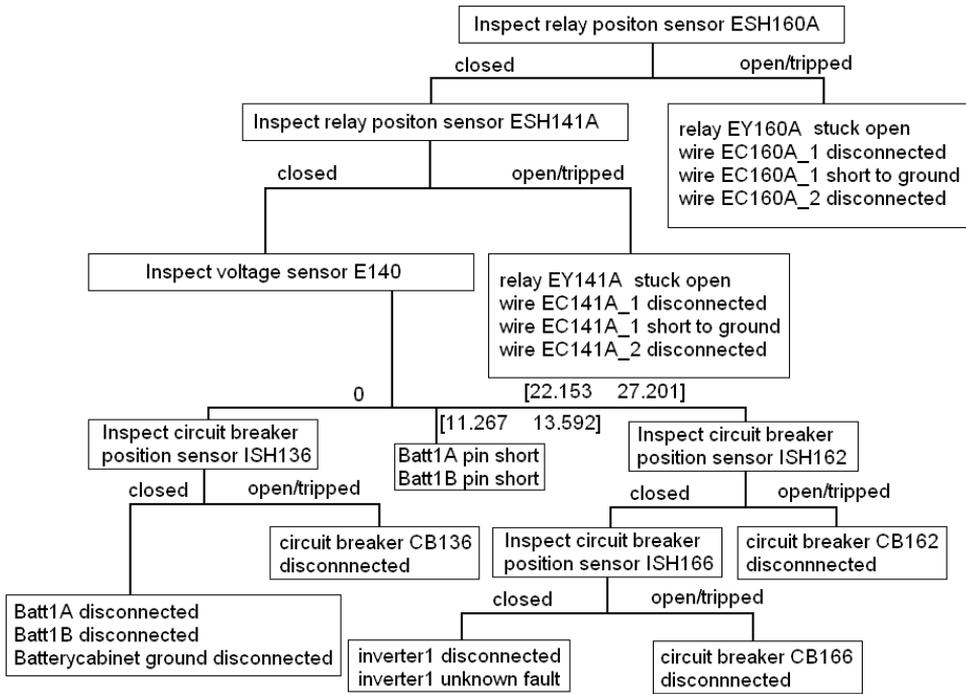


Figure 5.11. Decision tree created from the symptom: AC loads connected to the first load bank off. This tree is used for the first fault in scenario 1d, scenario 2a, 3a, 4a, the first fault in scenario 5b, scenario 6a and the second fault in scenario 6b.

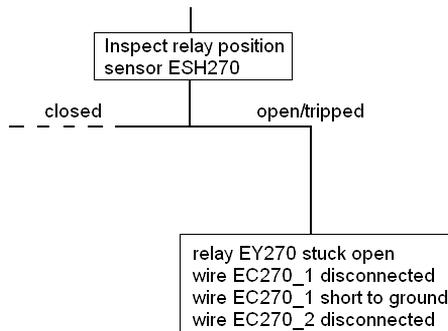


Figure 5.12. Decision tree created from the symptom: pump 1 off. This tree is used for the second fault in scenario 1d.

leaf contains the correct candidate and the wires leading to the failed relay. Additional measurements on the wires are needed for complete isolation. Figure 5.11 depicts how a user can find the correct candidate.

5.2.8 Scenario 2b

In this case relay EY244 has failed open and the symptom is loss of the entire second load bank. The user is asked to inspect relay position sensor ESH260A (which reports "closed") and then ESH244A (which reports "open/tripped"). The final leaf contains the correct candidate and the wires leading to the failed relay. Additional measurements on the wires are needed for complete isolation. Figure 5.9 depicts how a user can find the correct candidate.

5.2.9 Scenario 3a

In this case the first battery pack is disconnected and the symptom is loss of the entire first load bank. The user is asked to inspect relay position sensor ESH160A, then relay position sensor ESH141A (both reports "closed"), then voltage sensor E140 (which reports zero volts) and finally circuit breaker position sensor ISH136 (which reports "closed"). The final leaf contains the correct battery pack. Figure 5.11 depicts how a user can find the correct candidate.

5.2.10 Scenario 3b

In this case the second battery pack is disconnected and the symptom is loss of the entire second load bank. The user is asked to inspect relay position sensor ESH260A, then relay position sensor ESH244A (both reports "closed"), then voltage sensor E240 (which reports zero volts) and finally circuit breaker position sensor ISH236 (which reports "closed"). The final leaf contains the correct battery pack. Figure 5.9 depicts how a user can find the correct candidate.

5.2.11 Scenarios 4a, 5b fault 1 and 6a fault 1

The first inverter disconnected is injected in all three cases and the symptom is loss of the entire first load bank. Therefore the same tree is used. The user is asked to inspect relay position sensor ESH160A, then relay position sensor ESH141A (both reports "closed"), then voltage sensor E140 (which reports high voltage) and finally circuit breaker position sensors ISH166 and ISH162 (both reports "closed"). The final leaf contains the correct inverter. Figure 5.11 depicts how a user can find the correct candidate.

5.2.12 Scenarios 4b and 5a fault 2

The second inverter disconnected is injected in both cases and the symptom is loss of the entire second load bank. Therefore the same tree is used. The user is asked to inspect relay position sensor ESH260A, then relay position sensor ESH244A (both reports "closed"), then voltage sensor E240 (which reports high voltage) and

finally circuit breaker position sensors ISH266 and ISH262 (both reports "closed"). The final leaf contains the correct inverter. Figure 5.9 depicts how a user can find the correct candidate.

5.2.13 Scenario 6a fault 2

Circuit breaker CB180 disconnected is injected after the first inverter has been disconnected. The symptom is that current meter IT181 reports 0A. The user is asked to inspect relay position sensor ESH160A, then relay position sensor ESH141A (both reports "closed"), then voltage sensor E181 (which reports zero volts) and finally circuit breaker position sensor ISH180 (which reports "open/tripped"). The final leaf contains the correct circuit breaker. Figure 5.13 depicts how a user can find the correct candidate.

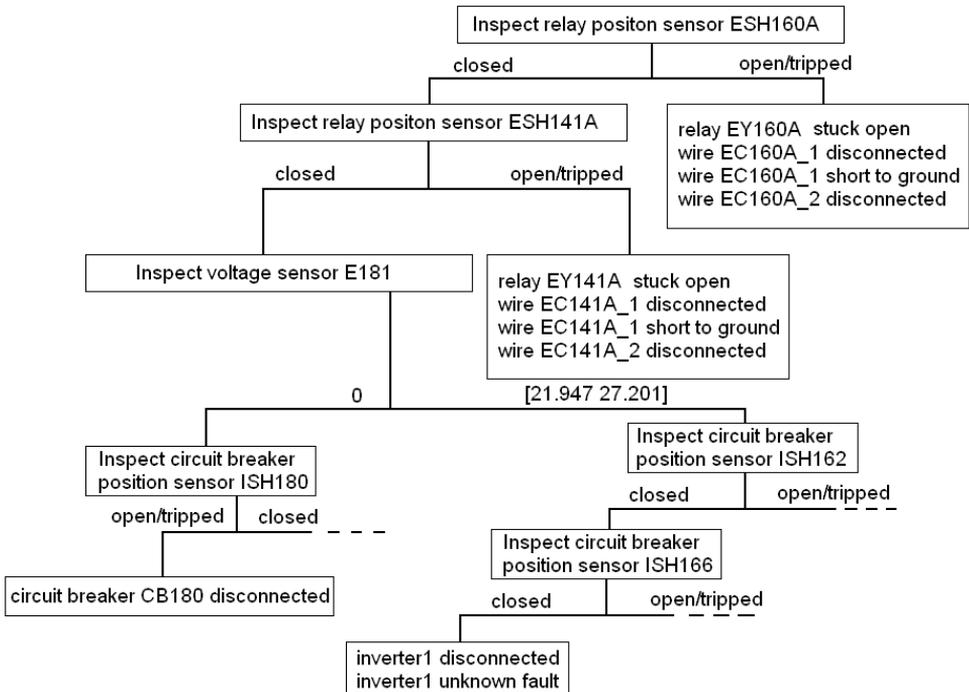


Figure 5.13. Decision tree created from the symptom: current sensor IT181 reports 0A. This decision tree is used for the second fault in scenario 6a and the first fault in scenario 6b.

5.2.14 Scenario 6b fault 1

Circuit breaker CB180 disconnected is injected. The symptom is that current meter IT181 reports 0A. The user is asked to inspect relay position sensor ESH160A, then relay position sensor ESH141A (both reports "closed"), then voltage sensor

E181 (which reports zero volts) and finally circuit breaker position sensor ISH180 (which reports "open/tripped"). The final leaf contains the correct circuit breaker. Figure 5.13 depicts how a user can find the correct candidate. The system configuration is different in this case compared to scenario 6a fault 2 but the same tree depicted in 5.13 was also generated in this case.

5.2.15 Scenario 6b fault 2

The first inverter disconnected is injected after circuit breaker CB180 has been disconnected and appropriate recovery action has been taken. The symptom is loss of the entire first load bank. The user is asked to inspect relay position sensor ESH160A, then relay position sensor ESH141A (both reports "closed"), then voltage sensor E140 (which reports high voltage) and finally circuit breaker position sensors ISH166 and ISH162 (both reports "closed"). The final leaf contains the correct inverter. Figure 5.11 depicts how a user can find the correct candidate. The system configuration is different in this case compared to scenarios 4a, 5b fault 1 and 6a fault 1 but the same tree depicted in 5.11 was also generated in this case.

5.2.16 Discussion

No decision trees have been generated for the first fault in scenario 5a and the second fault in scenario 5b. This was done for the same reasons discussed in Section 5.1.1. The correct faulty component is isolated in all cases except when a fault is injected into a relay or a position sensor. This is caused by the modeling depth which is discussed earlier in Section 5.1.1. When battery pack disconnected is injected, the final leaf contains a number of candidates. All of them are related to the correct battery pack. A model with less depth and one big component for the entire battery pack would achieve perfect isolation in the experiments. The reason to keep the model depth was the temperature sensors attached to each battery inside the battery packs. When "inverter disconnected" is injected, the model can not discriminate between the two failure modes: "disconnected" and "unknown". Either way the correct failed component is detected. As mentioned in Section 4.1.10, the "unknown" failure mode was modeled to catch unanticipated failures.

5.3 Interactive model-based diagnosis in RODON

RODON can also be used to interactively isolate faults by using the IMBD (Interactive model-based diagnosis) environment depicted in Figure 5.14. Given a symptom, i.e. the lamps in lamp box 1 are not shining as in Figure 5.14, a number of measurements are suggested. When the result from a measurement has been inserted the diagnosis engine uses the new information to narrow down the number of candidates as depicted in Figure 5.15. When enough measurements have been performed, the faulty component(s) is isolated and the diagnosis is completed. When this happens, RODON can provide the user with information about the

faulty component, i.e. how to replace it as depicted in Figure 5.16. The component information presented could i.e. be provided by Uptime which is a technical content management system developed by Uptime Solutions.

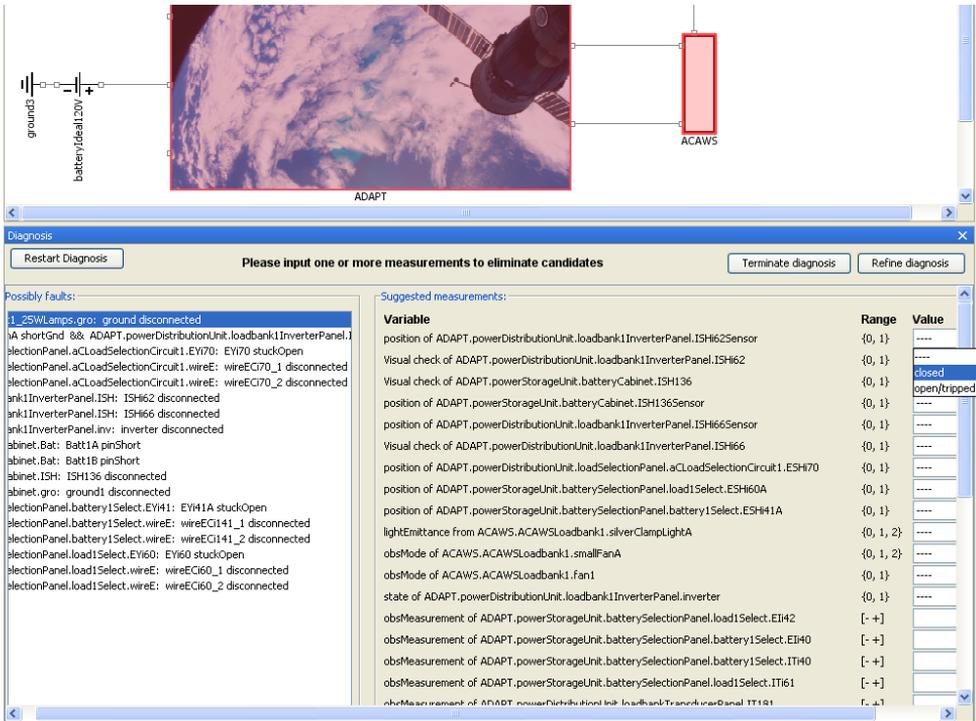


Figure 5.14. Screenshot of the RODON IMBD (Interactive model-based diagnosis) environment when the symptom: lamps in lamp box 1 off have been chosen. The possible fault candidates are depicted in the bottom left and the suggested measurements are depicted in the bottom right. The possible faulty components are highlighted in the model which is depicted in the top of the figure.

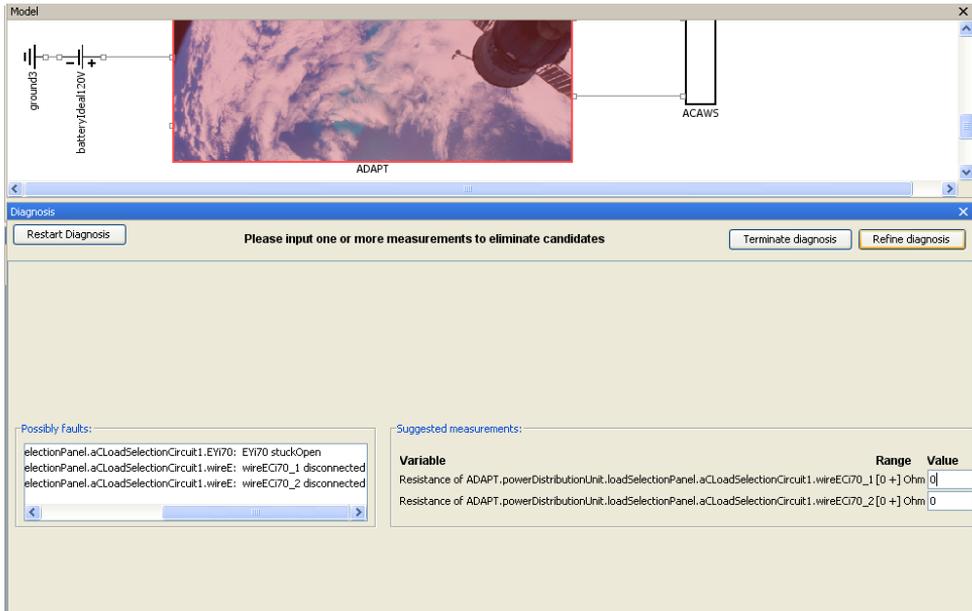


Figure 5.15. Screenshot of the RODON IMBD (Interactive model-based diagnosis) environment. Measurements have been inserted and the number of possible faulty components has been narrowed down to include a faulty relay and the wires leading to it. In the screenshot, the user is in the progress of inserting resistance measurements performed on the wires which will isolate the faulty relay.



Figure 5.16. Screenshot of the RODON IMBD (Interactive model-based diagnosis) environment. Enough measurements have been inserted and the faulty relay has been isolated. RODON can provide the user with information about the component as depicted in the bottom right of the figure.

5.4 Related work

There are several other vehicle health management systems that have been or are being integrated into the testbed. The different systems have been developed by industry, academia or government and they are explained in the following sections.

5.4.1 Testability Engineering and Maintenance System - Real Time (TEAMS-RT)

TEAMS-RT is a commercial model-based reasoning tool developed by Qualtech Systems Inc. TEAMS-RT uses results from tests to monitor the health of the system. The test results are created in other modules and could be i.e. to filter a sensor reading and compare it with a threshold. TEAMS-RT uses colored directed graph models known as multi-signal models [18]. The multi-signal models are created in the TEAMS designer graphical environment and contains information about "failure sources, monitoring and observability (the mechanisms by which sensor information is included), redundancy, and system modes" [18]. The TEAMS model can then be used to extract a table or matrix used by the inference engine to find which components can cause a given test to fail or which tests can be used to check the health of a given component. Figure 5.17 depicts how TEAMS-RT can be used to perform online monitoring on a system.

The TEAMS-RT inference engine uses the tests to match a component with one out of four states: unknown, good, bad and suspect in an iterative process. Initially, the state of all components is unknown. When a test involving a component passes, the component's state is updated to good. Similarly, when a test involving a component fails, the component's state is updated to suspect. The bad components are isolated from the suspected components by successfully eliminating good components from the suspected. The processing times are in the size of about 50 ms for a system with 1000 faults and tests and 80 modes of operation [7]. These processing times are not comparable with the RODON calculation times presented in Figures 5.1 and 5.2 since the complete RODON model was used. However, it would be interesting to compare them with an application that uses diagnostic rules generated from the RODON model. Such an application would have processing times of similar size.

The top level of the TEAMS-RT ADAPT model consists of five subsystems listed below together with their function [18]. Figure 5.18 depicts the top level of the TEAMS-RT ADAPT model.

1. "Power_Generation": supplies the "PowerStorage" subsystem with power.
2. "PowerStorage": consists of three rechargeable batteries and other equipment that facilitates control and safety of the system.
3. "PowerDistribution": supplies the "Loads" with AC and DC power.
4. "Loads": Contains the electrical loads.
5. "Monitor_Control": not developed in detail at this time [18].

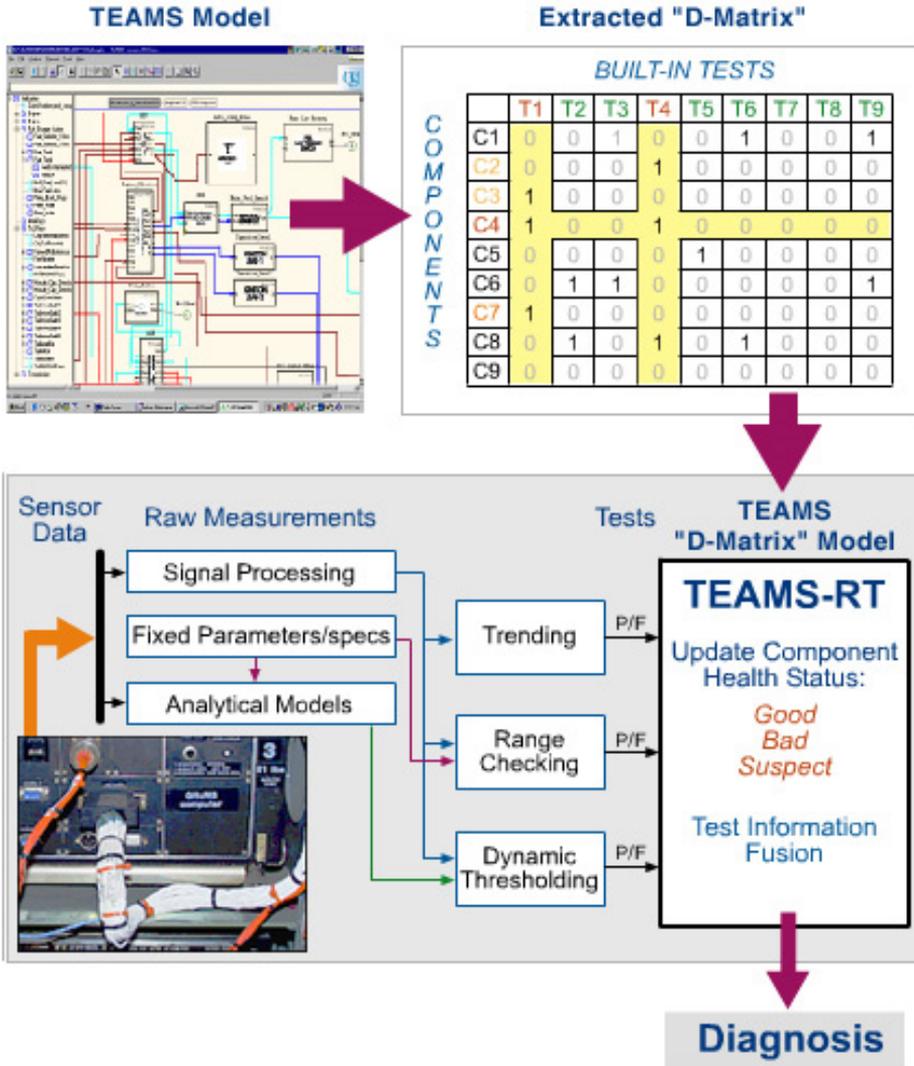


Figure 5.17. Figure showing how TEAMS-RT can be used to perform online monitoring on a system. The picture is taken from the Qualtech Systems Inc. homepage the 12:th of December 2008.

allows users to define new modeling languages using UML-based meta-models [12]. HyDE supports components with failure modes and uses conflict-directed search explained in Section 2.1.2. HyDE can perform on-board diagnosis and can track a target system in fault modes as well as non-observable modes. HyDE models support logical and basic arithmetic constraints as well as first-order differential equations and interval-valued variables. Advanced mathematical functions such as logarithm and trigonometric functions are not yet supported [20]. HyDE is different from many other diagnostic models since it uses predictive models to predict the behavior of the target system. Normally, a diagnostic system starts with sensor data and tries to find the current system behavior mode. HyDE works the other way around since it predicts sensor readings based on the current estimated system state. If the predicted sensor readings are consistent with the real ones, the current estimated system state is kept. If they are inconsistent HyDE generates candidates and a new system state with fault modes. The new system state is used to predict sensor readings which are compared with real ones and if they are consistent, the new estimated system state is kept. The most likely candidate is generated first and the search for candidates goes on until a user-defined termination condition has been fulfilled.

HyDE uses two types of models. One is the transition model where operation modes and conditions for transition between the operation modes are described. Another is the behavior model which consists of three parts: the propagation model, dependency model and the integration model. The propagation model contains information about how unknown variable values can be estimated from known variable values. The dependency model contains information about how different variables depend on each other. The integration model is used to describe how variable values change over time steps. The integration model is only needed if the system contains differential equations.

The HyDE ADAPT model consists of 155 components and 8 different types of failure modes [20]. Sensors have been modeled as components with failure modes and no differential equations have been modeled. The model does not contain real values and instead it uses discrete values. The HyDE ADAPT model is depicted in Figure 5.19 where each box represents a component.

The HyDE ADAPT model has been tested in a number of experiments and a table with the results from these tests is depicted in Figure 5.20. The user-defined termination condition was changed and some of the experiments were repeated. The test results from these experiments are depicted in Figure 5.21. An important note is that these test results are over one year old and some improvements have been made. The presented times include some transient cool-down policy times and are not pure computation times which would be comparable with the RODON results presented in Figure 5.22.

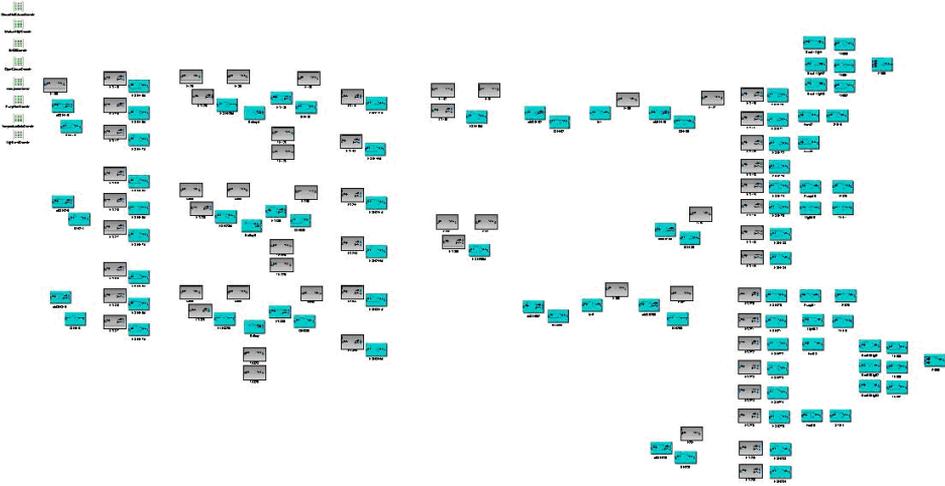


Figure 5.19. Picture of the HyDE ADAPT model. The picture is taken from [20].

5.4.3 Fault Adaptive Control Technology (FACT)

FACT is a tool set developed at Vanderbilt University. FACT has been created by using the Generic Modeling Environment (GME) and it allows modeling, diagnosis and fault adaptive control of complex systems. The tool set consists of three environments: a modeling environment, a simulation environment and a computational environment [18]. The models are created in the modeling environment and they consist of components which are Hybrid Bond Graph (HBG) model fragments. Bond graphs are based on the conservation of energy and continuity of power and are used as a generic modeling language in several types of systems. A HBG is a bond graph which has been extended with junctions that can be switched on and off. Faults are modeled in the HBG model fragments as undesired parameter value changes. The parameters that can be fault candidates are set by an attribute in the modeling environment. The simulation environment enables simulation models to be programmatically derived from the HBG model fragments and simulation experiments executed for both nominal and faulty scenarios. The computational environment provides run-time support for fault detection, isolation, identification, and fault adaptive control.

The FACT ADAPT model is a component-based top-down model where some components contain parameters that can be a fault candidate. Each component contains a model of its own behavior and is connected to other components and the environment through an interface. The interface consists of two types of ports: energy ports and signal ports. The energy ports are used to model energy exchange between the component and other components. The signal ports are used to model input and output of signal values from the component. The FACT accumulator

ADAPT Experiment ID	Injected Fault and Location	HyDE Diagnosis	Correct Fault Isolation?	Time (s)
317	None	No faults reported	yes	-
304	Relay Failed Open, EY260	EY260.recoverablyFailedOpen	yes	35.7
		EY260.failedOpen		
		EY260.recoverablyFailedOpen ESH260A.unknownFault		
305	Relay Feedback Sensor Failed, ESH175	EY175.failedOpen	yes	8.4
		EY175.recoverablyFailedOpen		
		ESH175.unknownFault		
306	Circuit Breaker Tripped, cbISH262	cbISH262.tripped	yes	89.5
		cbISH262.failedOpen		
		Inv2.unknownFault		
		ISH262.unknownFault		
307	Circuit Breaker Feedback Sensor Failed, ISH210	ISH210.unknownFault	yes	11.9
		cbISH210.tripped		
		cbISH210.failedOpen		
308	Voltage Sensor Failed, E261	E261.unknownFault	yes	16.6
		E261.unknownFault		
		E261.unknownFault		
309	Battery Low Voltage, BATT1	BatteryA.lowVoltage	yes	93.1
		EY136.failedOpen		
		ISH136.unknownFault		
		E135.unknownFault		
		EY136.tripped		
ISH136.unknownFault				
E135.unknownFault				
310	Inverter Failed Off, INV1	Inv1.unknownFault	yes	12.2
		Inv1.unknownFault		
		Inv1.unknownFault		
311	Load Sensor Failed, LT500	LT500.unknownFault	yes	11.6
		LT500.unknownFault		
		LT500.unknownFault		

Figure 5.20. Test results when the HyDE ADAPT model was used in a number of experiments. The table is taken from [20].

ADAPT Experiment ID	Injected Fault and Location	HyDE Diagnosis	Correct Fault Isolation?	Time (s)	Time Decrease
314	Battery Low Voltage, BATT1	BatteryA.lowVoltage	yes	19.4	79.2%
315	Circuit Breaker Tripped, cbISH262	cbISH262.tripped cbISH262.failedOpen	yes	38.8	56.6%
316	Relay Failed Open, EY260	EY260.recoverablyFailedOpen EY260.failedOpen	yes	17.5	51.0%

Figure 5.21. Test results when the HyDE ADAPT model was used in a number of experiments. The table is taken from [20].

Exp.	Frame	Injected faults	Candidates	RODON Time
304	150	EY260 stuckOpen	EY260 stuckOpen wireEC260_1 disconnected, wireEC260_1 short_to_gnd, wireEC260_2 disconnected,	3.7 s
305	120	ESH175 disconnected	ESH175 disconnected wireESH175_1 disconnected, wireESH175_1 short_to_gnd, wireESH175_2 disconnected, wireESH175_2 short_to_gnd,	4.2 s
306	125	ISH262 stuckOpen	ISH262 disconnected	2.3 s
308	125	E261 shortGnd	E261 shortGnd E261 stuck, wireE261 short_to_gnd,	2.9 s
309	350	BATT1 disconnected	BATT1A disconnected	2.8 s
310	170	INV1 disconnected	INV1 disconnected, INV1 unknown,	3.1 s
311	120	LT500 shortGnd	LT500 shortGnd, LT500 stuck, wireLT500 short_to_gnd,	2.7 s

Figure 5.22. Test results when the RODON ADAPT model was used in a number of experiments. The presented times are pure computation times.

model is depicted in Figure 5.23 together with an electrically equivalent circuit. The capacitor C_0 models the battery's state of charge, the resistor R_p models loss to parasitic reactions (e.g. gas production) and the capacitor-resistor pairs $C_1 - R_1$ and $C_2 - R_2$ models the battery's internal resistance and the battery's parasitic capacitance. The capacitor-resistor pair $C_3 - R_3$ is only active during charging and that behavior is modeled by a switch. The switch is open during discharging and closed during charging.

The FACT diagnosis engine uses a hybrid observer that uses sensor readings to track the target system. The observer uses an extended Kalman filter (EKF) which includes sensor noise and modeling errors while tracking the system. The observer output is continuously compared with the behavior predicted by the model. If the observed and predicted systems are different with enough probability, the fault isolation scheme is triggered which is used to find the correct candidate.

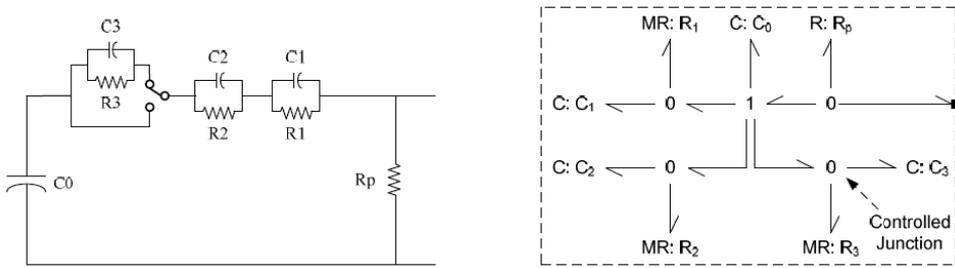


Figure 5.23. An electrically equivalent circuit of the accumulator to the left and the hybrid bond graph to the right.

5.4.4 ADAPT bayesian networks (BN) Model

The current ADAPT BN model was created by NASA Ames Research Center in collaboration with Mark Chavira and Adnan Darwiche of the University of California Los Angeles (UCLA). The nodes in the bayesian network represent random variables and the arcs between the nodes represent relations between the variables creating directed acyclic graphs. A dynamic BN replicates the network for each time instance with relations between the networks at different time instances [18]. The ADAPT BN model was generated offline from a high level system specification. The generated BN could be used directly as an inference tool or compiled again into an arithmetic circuit. The arithmetic circuit has fast execution times and can be used for on-board monitoring applications which is the case with the model used for ADAPT. The arithmetic circuit used with ADAPT was compiled with a system called ACE. Figure 5.24 depicts how a system specification can be used to compile a BN model which could be used to compile an arithmetic circuit.

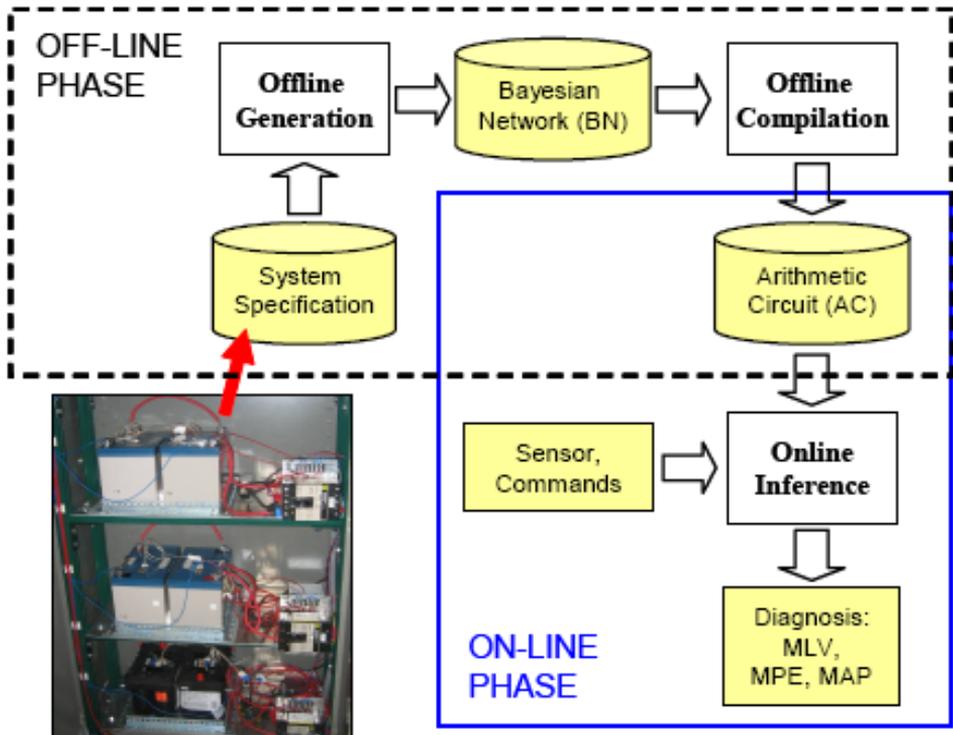


Figure 5.24. A high level system description is used to compile a BN model. The BN model can be used to compile an arithmetic circuit suitable for on-board monitoring. The picture is taken from [13]

The current ADAPT BN model consists of more than 400 nodes and supports failure modes and different operational states. The ADAPT EPS from the batteries

and downstream to the loads have been modeled which is a model of the ACAWS scope of scenarios depicted in Figure 3.22. The model was used with the series of experiments described in HyDE section. The results are presented in Figures 5.25 and 5.26. These execution times are not comparable with the RODON calculation times presented in Figures 5.1 and 5.2 since the complete RODON model was used. However, it would be interesting to compare them with an application that uses diagnostic rules generated from the RODON model.

ID	Fault Description	Diagnosis: MAP, MAP _{MPE} , and MAP _{MLV}	Correct
304	Relay EY260 failed open	<i>Health_relay_ey260_cl = stuckOpen</i>	Yes
305	Relay feedback sensor ESH175 failed	<i>Health_relay_ey175_cl = stuckOpen</i>	Yes
306	Circuit breaker ISH262 tripped	<i>Health_breaker_ey262_op = stuckOpen</i>	Yes
308	Voltage sensor E261 failed	<i>Health_e261 = stuckVoltageLo</i>	Yes
309	Battery BATT1 voltage low	<i>Health_battery1 = stuckDisabled</i>	Yes
310	Inverter INV1 failed off	<i>Health_inv1 = stuckOpen</i>	Yes
311	Load sensor LT500 failed	<i>Health_LT500 = stuckLow</i>	Yes

Figure 5.25. The diagnostic results with the arithmetic circuits compiled from the BN model. The picture is taken from [13].

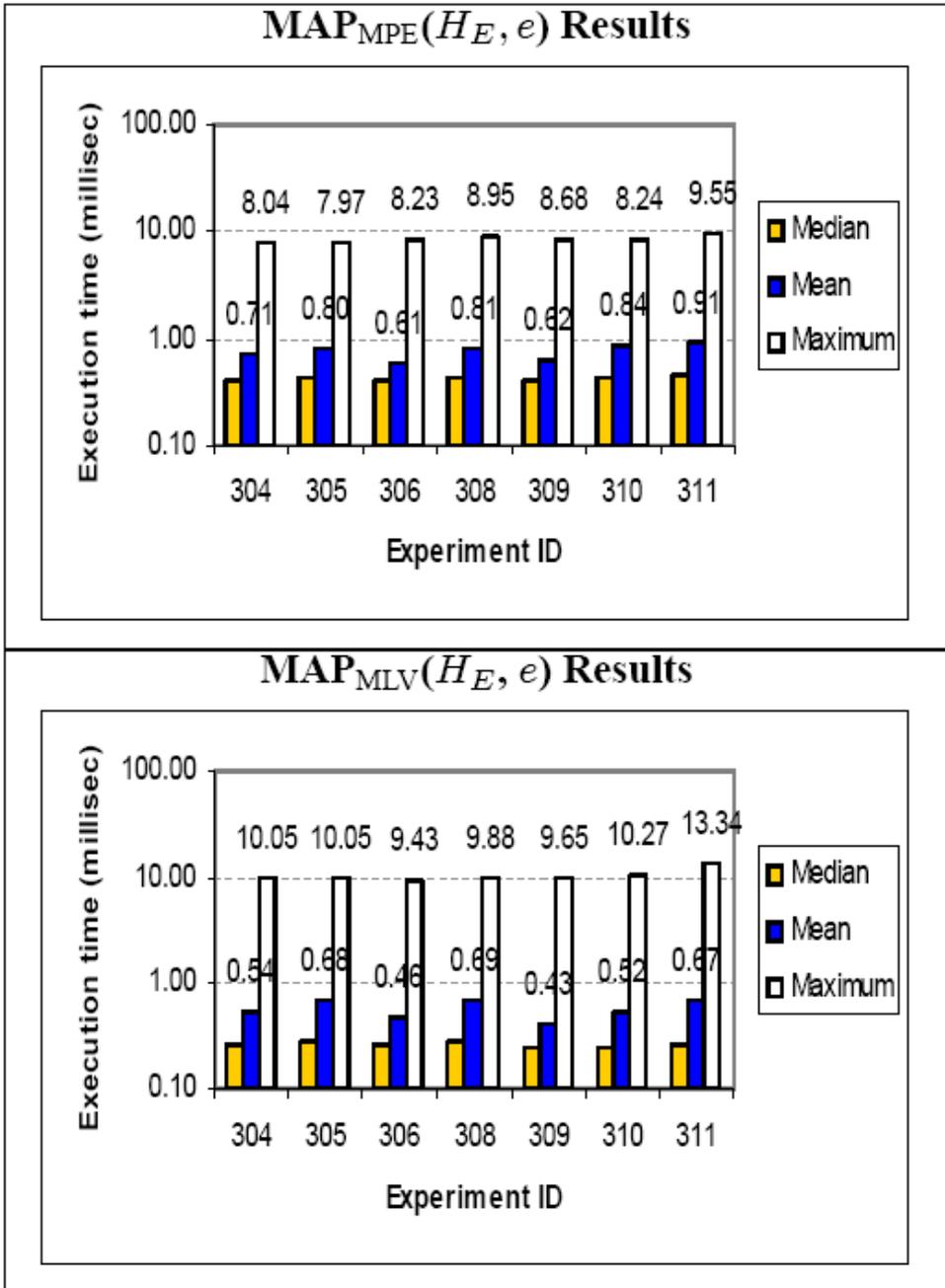


Figure 5.26. The execution times with the arithmetic circuits compiled from the BN model. The picture is taken from [13].

5.5 Conclusions

Based on the presented results, RODON is very well suited for stationary analysis of large systems with a mixture of continuous and discrete signals. It was easy to capture signals taking an unwanted path, i.e. sneak currents, with the use of bi-directional signals. The large number of system configurations was also easy to capture and only one model was needed which also included faulty behavior. It is possible to get very good results using RODON but in turn it requires an equally good model.

A full analysis of the dynamic capabilities of RODON was never conducted in the thesis which is why no conclusions can be drawn for that case. Several of the presented tools besides RODON had faster computation times but in those cases a compiled model was used. RODON has the capability to generate diagnostics rules which would push the computation times down to those levels but that was not done in this thesis.

5.6 Future work

A natural continuation of the thesis project would be to implement a dynamic model where data from more than one time instance is used. With a dynamic model, faults in temperature sensors would be detected and transient behavior of different components could be included. The model could then be used to generate diagnostic rules for online monitoring. After that, an interface between RODON and the ADAPT API would have to be developed and then the online monitoring capabilities of RODON could be evaluated. If the online monitoring would work well, it would be interesting to remove the operator from the loop and use the generated candidates to automatically take an appropriate recovery action to the injected faults.

Bibliography

- [1] The Modelica Association (2007). Modelica - a unified object-oriented language for physical systems modeling - language specification version 3.0. September 2007.
- [2] Steven Hegedus Antonio Luque. *Handbook of Photovoltaic Science and Engineering*. John Wiley and Sons, 2003. ISBN 978-0-471-49196-5.
- [3] Peter Bunus and Karin Lunde. Supporting model-based diagnostics with equation-based object oriented languages. In proceedings of the 2nd International Workshop on Equation-Based Object-Oriented Languages and Tools, Paphos, Cyprus, July 8, 2008.
- [4] Deltran Corporation. Deltran product homepage, battery charging basics. URL: http://batterytender.com/battery_basics.php/, visited 19:th novemeber 2008.
- [5] National Instruments Corporation. National instruments corporation homepage, compact fieldpoint product information. URL: <http://www.ni.com/compactfieldpoint/>, visited 19:th novemeber 2008.
- [6] Xantrex Corporation. Xantrex product homepage, prosine 1000 inverter data sheet. URL: <http://www.xantrex.com/web/id/181/docserve.aspx>, visited 19:th novemeber 2008.
- [7] Somnath Deb, Amit Mathur, Peter K. Willet, and Krishna R. Pattipati. Decentralized real-time monitoring and diagnosis. IEEE Transactions on Systems, Man, and Cybernetics.
- [8] C. Michael Hoff and Kenny Steves. New insights into thermal runaway of valve regulated lead-acid batteries. In proceedings of the International Stationary Battery Conference, Battcon 2005, Miami Beach, FL.
- [9] Evergreen Solar Incorporated. Evergreen solar homepage, datasheet of the ec-series (cedar line) photovoltaic solar panels. URL: http://www.evergreensolar.com/upload/pdf/us/C120_US_010407.pdf, visited 19:th novemeber 2008.

- [10] Evergreen Solar Incorporated. Evergreen solar homepage, frequently asked questions. URL: <http://www.evergreensolar.com/faq.html>, visited 19:th november 2008.
- [11] Evergreen Solar Incorporated. Installation and operation manual for the ec-series (cedar line) photovoltaic solar panels. URL: <http://www.evergreensolar.com/>, visited 19:th june 2006.
- [12] A. Ledeczki, A. Bakay, M. Maroti, P. Völgyesi, G. Nordstrom, J. Sprinkle, and G. Karsai. Composing domain-specific design environments. *IEEE Computer*, pp. 44-51, 2001.
- [13] Ole J. Mengshoel, Adnan Darwiche, Keith Cascio, Mark Chavira, Scott Poll, and Serdar Uckun. Diagnosing faults in electrical power systems of spacecraft and aircraft. In proceedings of the Twentieth IAAI Conference on Artificial Intelligence (IAAI-08), Chicago, IL, 2008.
- [14] Sriram Narasimhan and Lee Brownston. Hyde - a general framework for stochastic and hybrid model-based diagnosis. In proceedings of the 18th International Workshop on Principles of Diagnosis (DX-07), Nashville, May 29-31, 2007.
- [15] Jenny Nelson. *The Physics of Solar Cells*. Imperial College Press, 2003. ISBN 978-1-86094-340-9.
- [16] Mattias Nyberg and Erik Frisk. *Model Based Diagnosis of Technical Processes*. LiU Tryck, Linköping, Sweden, 2006.
- [17] Norm Picker, Shawn Puma, Scott Poll, Ann Patterson-Hine, and Joe Camisa. Advanced diagnostics and prognostics testbed (adapt) system description, operations, and safety manual. URL: <http://ti.arc.nasa.gov/projects/adapt/pubs/ADAPT-OperationsSafetyManual.txt.pdf>, visited 19:th november 2008.
- [18] Scott Poll, Ann Patterson-Hine, Joe Camisa, David Nishikawa, Lilly Spirkovska, David Garcia, David Hall, Christian Neukom, Adam Sweet, Serge Yentus, Charles Lee, John Ossenfort, Ole J. Mengshoel, Indranil Roychoudhury, Matthew Daigle, Gautam Biswas, Xenofon Koutsoukos, and Robyn Lutz. Evaluation, selection, and application of model-based diagnosis tools and approaches. In proceedings of the American Institute of Aeronautics and Astronautics (AIAA) Aerospace Conference, Rohnert Park, California, May 7-10, 2007.
- [19] Honeywell Sensing and Control. Platinum rtds reference data. URL: http://content.honeywell.com/sensing/prodinfo/temperature/technical/c15_136.pdf, visited 19:th november 2008.
- [20] Adam Sweet. Testing hyde on adapt. NASA STI Technical Memorandum, performing organization report number: NASA/TM-2008-214570.

-
- [21] Inc Technical Marketing Staff of Gates Energy Products. *Rechargeable Batteries Applications Handbook*. 1997. ISBN 978-0-7506-7006-7.
- [22] Olav Torheim. Elementary physics of p-n junctions. URL: <http://web.ift.uib.no/~torheim/pnsjikt.pdf>, visited 19:th novemeber 2008.

Appendix A

Abbreviations

- EPS: Electrical Power System
- ADAPT: The Advanced Diagnostics And Prognostics Testbed
- ACAWS: The Advanced Caution And Warning System
- MBD: Model-Based Diagnosis
- GUI: Graphical User Interface
- AC: Alternating Current
- DC: Direct Current
- RTD: Resistance Temperature Detector
- SSR: Solid State Relay
- DAC: Data Acquisition Computer
- ECU: Electronic Control Unit
- IMBD: Interactive Model-Based Diagnosis
- TEAMS-RT: Testability Engineering And Maintenance System - Real Time
- HyDE: Hybrid Diagnostic Engine
- FACT: Fault Adaptive Control Technology
- HBG: Hybrid Bond Graph
- GME: Generic Modeling Environment
- EKF: Extended Kalman Filter
- BN: Bayesian Network
- UCLA: University of California Los Angeles
- API: Application Protocol Interface