Master of Science Thesis in Electrical Engineering
Department of Electrical Engineering, Linköping University, 2025
Department of Electrical Engineering, Chalmers University of Technology, 2025

Thermal System Diagnostics Development For Electric Trucks

Alexander Tuneskog and Parth Vaze

Master of Science Thesis in Electrical Engineering

Thermal System Diagnostics Development For Electric Trucks

Alexander Tuneskog and Parth Vaze

LiTH-ISY-EX–25/5788–SE

Supervisor: **Arman Mohammadi**
ISY, Linköping University
Torsten Wik
E2, Chalmers University of Technology
Johan Engbom
Volvo GTT

Examiner: **Erik Frisk**
ISY, Linköping University
Torsten Wik
E2, Chalmers University of Technology

Division of Vehicular Systems
Department of Electrical Engineering
Linköping University
SE-581 83 Linköping, Sweden

Division of Systems, Control and Mechatronics
Department of Electrical Engineering
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden

Abstract

This thesis explores the development of a system-level diagnostic approach for the thermal management system of an electric truck, using the Global Simulation Platform (GSP). The study begins by collecting information about commonly occurring faults in electric trucks, with a focus on those related to the thermal system. From this list, selected faults are injected into GSP to evaluate the thermal behaviour of the electric vehicle system under fault conditions.

These identified faults are used to assess the impact of fault scenarios on the thermal system, with fault injection employed to evaluate the severity. From this analysis, one specific fault is selected for detailed investigation. A logic based diagnostic system is developed to detect this fault, using insights derived from the simulation based fault behaviour.

The diagnostic system is implemented and tested both on a test bench and in an electric vehicle to verify its function and logic. This thesis demonstrates how simulation driven analysis can be used to inform the design of practical diagnostic systems and presents a structured method for building logic based tools for fault detection in an electric vehicle thermal systems.

Acknowledgments

We would like to sincerely thank everyone who supported us during the course of this project. First, we are grateful to our colleagues at Volvo GTT for their helpful discussions, teamwork, and encouragement throughout the project. Working alongside them made the experience both valuable and enjoyable.

We would especially like to thank our manager, Johan Engbom, for his continuous support, clear guidance, and trust in our work. A special thanks also goes to our company thesis supervisors, Fredrik Lagerlöf and Linnea Rydberg, whose technical advice and thoughtful feedback helped us stay on track and improve the quality of our work. We would also like to give a special thanks to Rahul Nath, Arash Majedi, and Jesper Nordenham for their extra support and collaboration. Their insights, patience, and willingness to share their knowledge greatly impacted this project.

Finally, we would like to thank our academic supervisors and examiners, namely Torsten Wik, Chalmers University of Technology, and Arman Mohammadi and Erik Frisk at Linköping University. They provided us with valuable insights, support, and guidance throughout the research.

Thank you all for making this journey a meaningful and enriching one.

Gothenburg, June 2025
Alexander Tuneskog and Parth Vaze

Table of Contents

Notation

Abbrevation	Meaning
CAN	Controller Area Network
CC	Critical Characteristic
CUSUM	Cumulative SUM
DCP_Clt	Decoupling Coolant
ESS	Energy Storage System
ESS_Clt	Energy Storage System Coolant
FMEA	Failure Modes and Effects Analysis
GSP	Global Simulation Platform
HVAC	Heating Ventilation and Cooling System
MDS	Motor Drive System
MDS_Clt	Motor Drive System Coolant
RPN	Risk Priority Number
SC	Significant Characteristic
VHPCU	Vehicle Hybrid Power Control Unit
Volvo GTT	Volvo Group Trucks Technology

1

Introduction

The shift from internal combustion engine trucks to electric trucks marks an important step toward sustainable transportation. Of importance in this transition is managing battery and electric motor temperatures to ensure performance, safety, and long lifespan. This is managed by a thermal management system in the electric truck. An overview of the thermal system is presented in Figure 1.1.

Faults in the thermal system can lead to reduced battery longevity, electric motor failures and other failures. This master's thesis focuses on development of a thermal system diagnostic and detecting such faults on a system level to help maintain the reliability and longevity of electric trucks. In the present system design the diagnostic functions are on a component level and Volvo Group Truck Technology (GTT) has a need for a thermal system diagnostic. A study has been performed to understand the present system design and what kind of issues that can arise when different components are broken. This has been achieved by performing fault simulations in Volvo GTT's simulation environment, Global Simulation Platform (GSP). Insights from the study have been adapted to design a new diagnostic strategy. The functionality of this strategy has been tested and verified on Volvo GTT's vehicle diagnostic platform.

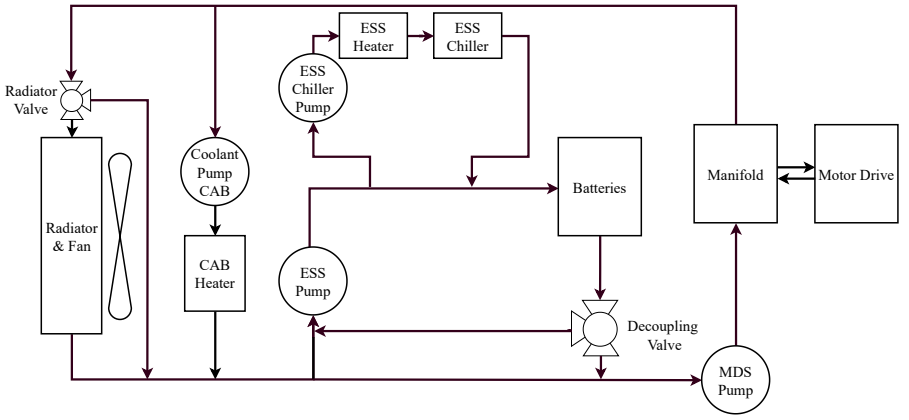


Figure 1.1: *Overview of the thermal system in the electric truck.*

1.1 Background

The project has focused on the thermal system used in Volvo's electric trucks. The thermal system refers to the components for temperature control of the electric motors and batteries for the electric truck. For the Volvo GTT, it is important to develop the safest systems for their trucks to maximise the uptime and to have a robust design. To minimise operational downtime it is important to identify and handle different faults that can occur in a truck. A fault can be defined as an unacceptable deviation from at least one characteristic or value of the system from standard behaviour [1]. Furthermore the complexities involved in building the electric propulsion system for the trucks, lead to numerous components where faults can occur. These faults can affect not only the safety but also the functionality of a truck. There is also the issue of faults leading to replacing complex and expensive components. It is crucial to understand the performance of the system and potential faults that can occur.

The task undertaken in this project focused on leveraging system behaviour learning from the GSP simulation and using it as a base for diagnostic strategy development. This approach differs from the current methodology for fault diagnosis used at Volvo GTT. The current system independently identifies failures in individual components, such as pumps and valves, among other components. In this project, the diagnostic strategy strives to look at the system as a set of interconnected parts instead of independent components.

1.2 Motivation

The investigation focuses on how the GSP behaves when different faults are induced in various components in the thermal system. The behaviour of the system due to these faults will form the basis of the fault identification strategy for system diagnosis. The diagnostic function is designed to capture the effects of fault occurrences in the system and then trigger fault alarms depending on where the system failure has occurred. The diagnostic function is tested and verified in the Vehicle Hybrid Power Control Unit (VHPCU). The thesis project is important for Volvo GTT to evaluate whether the GSP simulation system can be used to develop different diagnostic strategies.

1.3 Research Questions

The goal is to deliver a diagnostic system that can be implemented on the actual electric truck and handle different failures in the thermal system. In particular, the research questions of this thesis project are:

- What are the potential faults and failures that can occur in the thermal system and its components?
- How does the thermal system behave (within the simulation setup) due to injection of different types of faults?
- Can these behavioural outputs (if physically feasible) contribute to building a system level diagnostic function (specifically for the thermal system)?
- How well does the designed diagnostic algorithm detect the thermal system faults when implemented in the VHPCU?
- Does the diagnostic function detect the faults when implemented on an actual electric truck?
- How can the existing Volvo GTT diagnostic system be improved?

1.4 Workflow

The project was divided into various work tasks to investigate the variooand these are enumerated below.

- Test the validity of fault injection in the simulation system.
- Evaluate if the simulation system can contribute to building a system level diagnostic function (specifically for the thermal system).
- Build diagnostic functions on the basis of simulation studies and verify them through virtual tests in the software.

- Test the diagnostic system by emulating faults in the VHPCU and verifying its response.
- Test the diagnostic function on the electric truck.
- Create a temperature estimation model and check its viability and accuracy against the available temperature sensor values.

2

Thermal System

The thermal system is managing the temperature of the batteries and electric motors of the electric truck. It consists of multiple temperature sensors, pumps, pipes for coolant flow, chillers and heaters. A virtual mode built by Volvo GTT in GSP is perform various tests on the system.

2.1 Physical Thermal System

The thermal system consists of two interconnected loops, the Energy Storage System (ESS) loop and the Motor Drive System (MDS) loop. The ESS loop consists of a heater, a chiller and two pumps. It is mainly used for heating or cooling the batteries. The MDS loop is the outer loop containing the MDS pump and the radiator, responsible for cooling the electric motors. There are three coolant temperature sensors in the thermal system :

- ESS Coolant (ESS_Clt).
- DCP Coolant (DCP_Clt).
- MDS Coolant (MDS_Clt).

The ESS and MDS can be coupled or decoupled depending on the situation. When the decoupling valve is open, the ESS and MDS loops are interconnected and coolant mixes into both the ESS and MDS loops, which is presented in Figure 2.1. When the decoupling valve is closed the ESS and MDS loops function as independent cooling loops as presented in Figure 2.2. The other valve, referred to as the radiator valve directs flow to the radiator. When the radiator valve is open, flow passes through the radiator (cooling the system); when closed, it bypasses the radiator.

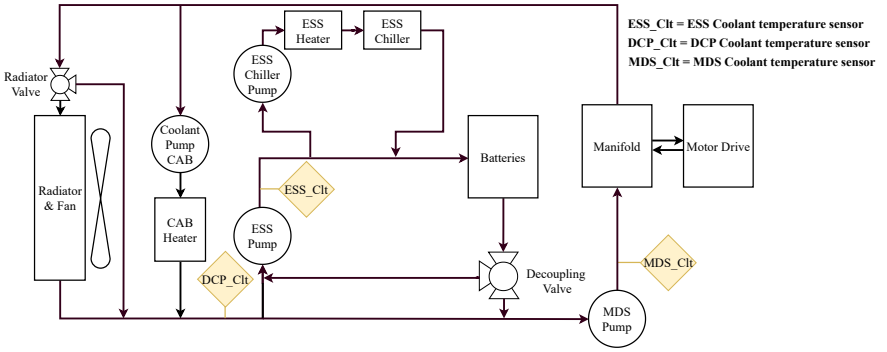


Figure 2.1: *Overview of the thermal system, with arrows indicating the direction of the coolant flow, when the ESS and MDS loops are coupled.*

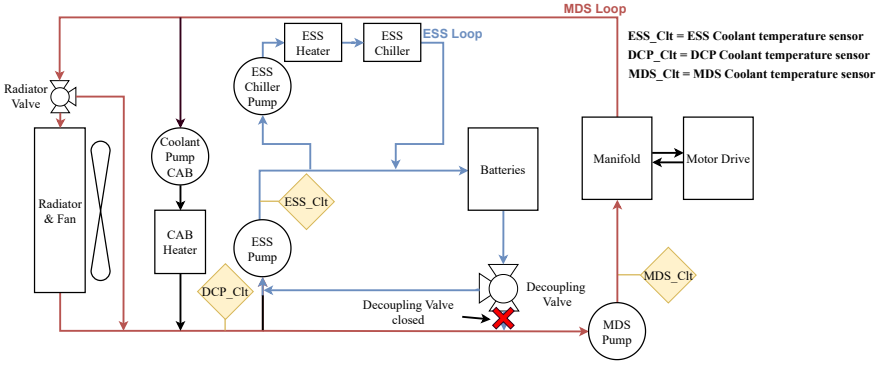


Figure 2.2: *Thermal system when the decoupling valve is closed. The ESS and MDS loops are decoupled.*

2.2 GSP Simulation Platform

Volvo GTT uses an in-house simulation platform, GSP, to run different types of tests and simulate different situations in their vehicles. The main components comprise:

1. Road and environment model: This model includes topographic information about the road and environmental conditions, like ambient temperature.

2. Driver model: This model simulates driver behaviour and reacts to the set speed output from the road and environment model.

3. Vehicle model: This model describes all the different components in the vehicle, including the motor drive system, energy storage system, cooling system, transmission etc. All of these systems also have control algorithms defined within this block. For example, the motor control system is defined for the motor drive system.

The block diagram of the overall simulation model is shown in Figure 2.3.

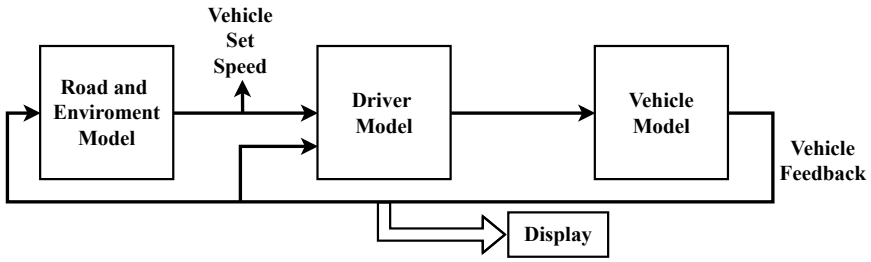


Figure 2.3: *Overview of the GSP model in Simulink.*

The main focus of the simulation analysis in this project is the cooling system plant model. The block diagram is shown in Figure 2.4. This model consists of the cooling system plant and the cooling system control blocks. The cooling plant model in Figure 2.4 represents the actual thermal system in the vehicle shown previously in Figure 2.1. In Figure 2.4 CAN refers to the Controller Area Network, which is used for the vehicle communication.

2.2.1 General Simulation Setup

The simulation is set up by selecting the initial battery temperature, ambient temperature, truck and drive-cycle. The initial battery temperature used for the simulation ranges from 11 to 30°C, which is the normal temperature the batteries operates at. Furthermore the ambient temperature range is -15 to 45°C. The simulation has not been tested and verified by Volvo GTT for temperatures below

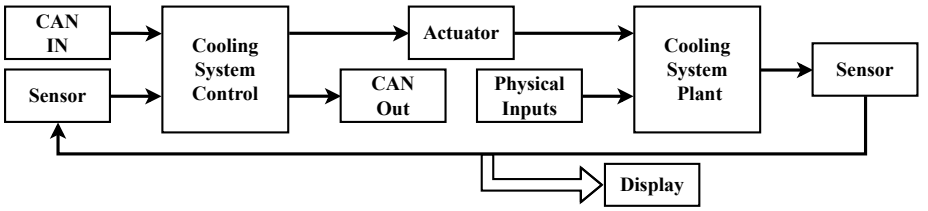


Figure 2.4: *Cooling System Model in Simulink.*

-15°C. Thus no simulation has been done for temperatures lower than -15°C, during the thesis. The simulation used a durability drive cycle. This particular road scenario was selected for a drive cycle containing steep uphill climbs and typical low-inclination road paths. The drive cycle serves to test the durability of the vehicle. Furthermore, the road inclination changes throughout the drive cycle with changes in the speed, torque, and power requirements of the vehicle. The elevation change within the entire drive cycle is shown in Figure 2.5. Furthermore, the same truck configuration was used for all simulations. This general simulation setup is summarised in Table 2.1.

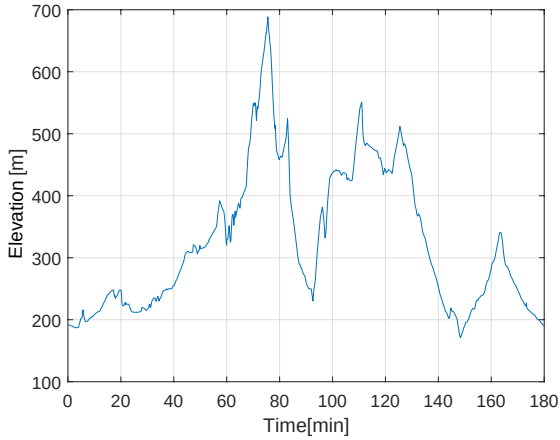


Figure 2.5: *Elevation change during the drive cycle.*

Table 2.1: *Parameters used in the simulation for a heavy duty electric truck.*

Listed Parameters	Selection
Initial battery temperature	11 to 30 °C
Ambient temperature	-15 to 45 °C
Drive-cycle	As seen in Figure 2.5

3

Literature Review

The literature review for the project broadly covers the following topics:

1. Failure mode and effects analysis for vehicle systems and components.
2. Definition of faults and fault diagnostics with a focus on electric trucks.
3. Theoretical methodologies for fault diagnostics.
4. Different faults that can occur on thermal system components.

One common method to rate the criticality of the fault and analyse the behaviour of a system under fault conditions is named FMEA (Failure Modes Effects and Analysis). It lists faults for each component, detailing the faulty component, altered functions, effects on the system, and probable causes. The criticality level is calculated by multiplying occurrence, severity, and detectability [2]. The FMEA can be important to use when trying to understand which faults are most important to check for. Fault injection techniques, which are used for simulating faults during diagnostic testing, are discussed in [3].

3.1 Failure Mode Effects and Analysis

Failure Mode Effects and Analysis (FMEA) is defined as the procedure for the analysis of a system to identify the potential failure modes, their causes and effects on system performance [4]. It is used to determine what effect a failure can have on the operation of a product or a system. The information gained from the FMEA enables the development of a risk profile for a system, which leads to robust system design. The main function of the FMEA is to increase product reliability and safety under failure conditions.

The FMEA is one way of identifying failures, effects and risks for a product or process, and then reducing or eliminating the failures. One way of evaluating the risk of failure is determined by three factors; severity, occurrence and detection [5].

- **Severity:** The consequence of a failure if it occurs.
- **Occurrence:** The probability of failure occurring.
- **Detection:** The probability of detecting the failure.

3.2 Definition of Faults

From [1], important terms related to faults and fault diagnostics can be enumerated as follows :

1. **Fault**: This is an unpermitted deviation of at least one characteristic property or variable of the system from the acceptable, usual, standard, or nominal behaviour.
2. **Failure**: A failure is a fault that implies permanent interruption of a system's ability to perform a required function under specified operating conditions.
3. **Fault Detection**: This is the process of determining if faults are present in the system and also the time when the fault occurred.
4. **Fault Isolation**: This is the process of determining the location of the fault, i.e, which component or components have failed.
5. **Fault Identification**: This is the process of determining the size and time varying behaviour of a fault.
6. **Fault Diagnosis**: The diagnostic system produces a diagnosis that is the conclusion of which fault or combination of faults can explain the behaviour of the process.
7. **False Alarm**: This is an event when an alarm is triggered even when no faults are present.
8. **Missed Alarm**: This is an event in which an alarm is not generated despite a fault occurrence. This event can also be defined as a missed detection event.

It is common to divide faults into different categories, as described in [6]. Figure 3.1 illustrates this and is explained as follows:

- **Plant faults:** Changes the dynamic I/O proprieties of the system.
- **Sensor faults:** Sensor readings have significant errors, but the plant properties are unaffected.

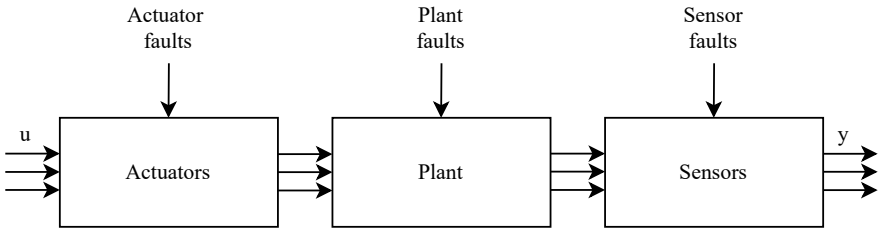


Figure 3.1: *Classification of faults referred from [6].*

- **Actuator faults:** The effect of the control input on the system is reduced, but the plant properties are unaffected.

3.3 Methodologies for Fault Diagnostics

Traditional diagnosis system has used mainly limit checks. For example, an alarm is generated when the signal is above or below the predefined threshold compared to normal operation [1]. Hardware redundancy is another traditional approach which uses duplication of hardware to detect a fault. For example, when two sensors are used to measure the same physical quantity. [7]. By comparing if the output of the process component and the redundant counterpart are similar, it is possible to determine if a fault has occurred or not, as illustrated in Figure 3.2. The advantages of this method are high reliability, direct fault isolation and maintained functionality even though a sensor fails.

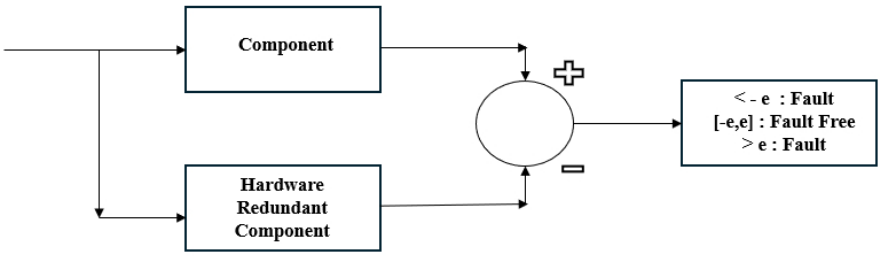


Figure 3.2: *Schematic view of hardware redundant scheme,[7].*

One method that is easy to implement is to use a plausibility check. The goal is to use physical laws under which the process component works. When a fault occurs, the idea is that the model and measurement output will be different, thereby enabling fault detection. The disadvantage is that the information gained about the fault can be limited. For complex processes it can therefore be difficult to isolate faults [7].

Another method is to use model-based fault diagnostics. The intuitive idea is to replace the hardware redundancy with a process model that can describe the physical process, which is called analytical redundancy. To get information about the potential fault, a comparison between the measured and estimated process variable can be done, this is called a residual and can be calculated as $r = y - \hat{y}$, where y is the measurement and $\hat{y}$ is the estimated value. There exist several methods for setting a fault, but generally a residual should be close to zero for the no fault case and otherwise it can be considered a fault [7]. By constructing multiple residuals that are sensitive to different faults, it becomes possible to isolate individual faults. For example, consider the following sensor measurements:

$$y_1 = x + f_1 \tag{3.1}$$

$$y_2 = x + f_2 \tag{3.2}$$

$$y_3 = x + f_3 \tag{3.3}$$

From these, the following residuals can be defined $r_1 = y_1 - y_2$, $r_2 = y_1 - y_3$ and $r_3 = y_2 - y_2$. The fault sensitivity based on residual r_1, r_2 and r_3, is presented in Table 3.1, making it is possible to isolate each fault.

Table 3.1: *Fault sensitivity.*

	f_1	f_2	f_3
r_1	x	x	
r_2	x		x
r_3		x	x

More fault diagnosis methods exist, and they can be categorised as signal-based, knowledge-based, hybrid (the combination method of at least two methods) and active fault diagnosis methods [8].

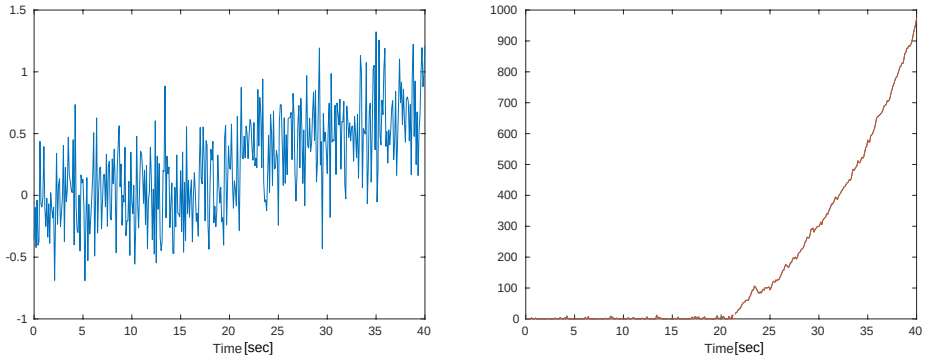
3.3.1 CUSUM

When taking decisions with residuals to trigger an alarm it can be important only to report an alarm when there is a fault and minimise the number of false alarms and missed detection. One way to set an alarm is by using Cumulative SUM (CUSUM). The way that CUSUM works is by using that a residual ideally should be zero or close to zero when in the fault free case and non-zero in case of a fault. The CUSUM is defined as

$$T'(t) = max(0, T'(t-1) + |r(t)| - \nu), \tag{3.4}$$

where the $T'(t)$ is the cumulative sum and $r(t)$ is the residual. ν is called the drift parameter and is a design parameter that is chosen by the user. Generally a large ν is used when the uncertainty is large and a small ν is used when the uncertainty is small [1].

To illustrate the use of the CUSUM, consider a residual signal $r(t)$ that is noisy with a certain variance. After some time, the mean of this residual changes, indicating that a fault or anomaly may have occurred. CUSUM can be employed to detect this change in the mean value. An example of a residual signal is shown in Figure 3.3a, and the result of applying the CUSUM is depicted in Figure 3.3b. As evident from the Figure 3.3b, the CUSUM output highlights a significant change, suggesting the presence of a fault. By defining an appropriate threshold, it is possible to trigger an alarm when the residual exceeds this threshold.



(a) *(b)*

Figure 3.3: *Illustration of CUSUM. (a) Residual signal $r(t)$. (b) CUSUM is applied to $r(t)$.*

3.4 Thermal Dynamics for Temperature Estimation

In order to simulate and estimate the thermal behaviour of electric truck systems, it is important to use underlying thermal dynamics. The equations presented in this section form the theoretical basis for the models used in later chapters to estimate and simulate component temperatures. To model a heater and cooler one way is to use the concept of heat transfer. This can be achieved by applying a simplified steady-state thermal energy balance equation:

$$Q = \dot{m} c_p (T_{out} - T_{in}) \tag{3.5}$$

where $\dot{m}$ is the mass flow, c_p is the specific thermal capacity at constant pressure, T_{in} and T_{out} refer to the temperature into and out of the system and Q is the rate of heat transfer [9]. The following assumptions for liquid are:

- When the fluid is an incompressible liquid and changes in kinetic energy, potential energy, and all types of work (including flow work) are negligible. In this case, flow work is usually small because the pressure changes are not significant.
- For an incompressible liquid where viscous dissipation is very small. Viscous dissipation is when mechanical energy is transformed into heat because of friction within the fluid. This effect becomes significant mainly when the fluid flows very fast or is very thick (highly viscous).

From [10] the First Law of Thermodynamics for a closed loop system can be described by

$$Q = W + \frac{dU}{dt}, \tag{3.6}$$

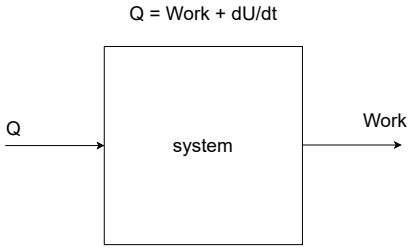


Figure 3.4: *First law of thermodynamics for a closed loop system, referred from [10].*

where Q is the heat transfer rate, W is the work transfer rate and $\frac{dU}{dt}$ is the change of rate of the internal thermal energy. This is illustrated in Figure 3.4. The equation for $\frac{dU}{dt}$ is,

$$\frac{dU}{dt} = mc_v \frac{dT}{dt}. \tag{3.7}$$

Under the assumption that the process is incompressible (such that under any pressure variation the V is constant), the specific heat capacities are equal $c_v = c_p$. Work can also be represented as heat exchanged to and from the system. Therefore Equation 3.6 becomes

$$Q = Heat + mc_p \frac{dT}{dt}. \tag{3.8}$$

3.5 Thermal System Component Failures

The thermal system has many components that can undergo different failure modes, leading to faults in the system. This section focuses on different faults that can affect the components in the system like pumps and temperature sensors.

3.5.1 Pump Failure Modes

Pumps serve to regulate flow of coolant through the cooling system and ensure proper circulation of the coolant to all areas that need to be cooled. A failure of a pump, can lead to multitudes of problems in the cooling system as smooth and consistent coolant flow is essential for appropriate cooling performance. The failures that can occur in a pump are listed as follows:

1. **Cavitation**: It is the formation and collapse of vapour bubbles in the liquid due to low pressure, causing vibrations, noisy operation and damage to components. It can be caused due to restrictions in inlet, high flow velocity and high fluid temperature. [11], [12]

2. **Vibration**: Excessive oscillatory movement in the pump which can be an indication of hydraulic or mechanical imbalances. This can be caused by misaligned or unbalanced impellers, worn bearings or cavitation. [11], [12]

3. **Low Flow**: Pump fails to deliver the required flow or is below the operational thresholds. This can be caused by air in the system, a degraded impeller or issues with the valve. [4]

4. **Overheating**: The temperature of the pump or the motor components can increase very rapidly, leading to material degradation. This can occur due to lack of lubrication, excessive load on the pump or friction. [4]

5. **Motor Overload**: The pump motor can have an electrical overload, causing overheating and, in extreme cases, potential failure of motor windings. This can be caused by increased system resistance or incorrect sizing of motor for the respective power requirements. [4]

6. **Excessive Noise**: A loud operational noise may occur due to mechanical problems or flow anomalies like cavitation. [11], [12]

3.5.2 Temperature Sensors

Any thermal system needs temperature sensors to monitor the temperature values across different points in the system. Different types of temperature sensors can be used depending on the requirements. These can include thermocouples, resistance temperature detectors and thermistors. The temperature values given by these sensors are used to decide where and what to cool in the system. Therefore any fault in these temperature sensors can cause issues with thermal system operation. The different types of faults that can occur in these sensors are enumerated as follows:

1. Open Circuit Faults: This can commonly occur in thermocouples due to mechanical stress or corrosion. It is characterised by a broken wire or loose connection in the sensor, leading to incorrect outputs. [13]

2. Short Circuit Faults: Failure of sensor insulation, or touching of sensor leads to each other, causing a false sensor reading. [13]

3. Moisture Ingress: Intrusion of water into the sensor body causing corrosion or shorting of the sensor terminals. [13]

4. Sensor Drift: The output reading of the sensor changes gradually over time due to aging, contamination or damage to the sensor.[14]

5. Thermoelectric Drift: Seebeck coefficient drift in thermocouples which causes problems in output reading of temperature values. [14]

6. Self Heating: Excess current in the sensor heats the body of the sensor, affecting the readings of the sensors.[15]

7. Improper Calibration: Incorrect setup of the temperature sensors leading to inaccurate measurements. [16]

4

Methodology: FMEA and Fault Injection

There are multiple failures that can occur on different components for the thermal system. A list of such failures was created by Volvo GTT through a FMEA process conducted on the physical thermal system. All possible thermal system failures are classified into four categories.This chapter outlines how the failures are converted into equivalent faults that can be emulated in the GSP simulation system. The effects of fault injection in the GSP simulation system are described.

4.1 Fault Classification

The classification of failures is based on previous work conducted by Volvo GTT:

1. Low risk: A failure with high probability of detection, low probability of occurrence.

2. High risk: A failure with moderate probability of detection, low probability of occurrence.

3. Possible Significant Characteristic (SC): A failure with moderate probability of detection, low probability of occurrence but a significant effect such as an unplanned stop.

4. Possible Critical Characteristic (CC): A failure with high probability of detection, low probability of occurrence but has a high risk of causing injury without warning.

Table 4.1 and Table 4.2 present various low and high risk faults that can occur in the thermal system. The possible SC and CC faults are presented in Table 4.3 and Table 4.4.

Table 4.1: *Low risk failures.*

Failed Part	Effects	Potential Cause
ESS pump	Unable to provide the required cooling for ESS	Loss of pump function (mechanical wear and tear/damage)
ESS Chiller pump	Partial loss of Cooling function	ESS Chiller pump unable to function at the required speed (Lower or Higher)
Radiator	Unable to provide the required cooling for ESS	Insufficient function due to clogging and/or leakage in the Radiator
Joint, pipe and hoses	Unable to provide the required cooling for ESS	Leakage of coolant in any of the joints, pipe and hoses or the components

Table 4.2: *High risk failures.*

Failed Part	Effect	Potential Cause
DCP_Clt	Unable to provide the required cooling for ESS	DCP_Clt malfunction. Leading to incorrect temperature reading
ESS heating, battery coolant heater	Too little heat while charging	Component fails or control fails

Table 4.3: *Possible SC failures.*

Failed Part	Effect	Potential Cause
MDS pump	Partial loss of cooling function	Low flow from MDS pump
MDS and ESS pumps	Partial loss of cooling function	Low flow from ESS and MDS pumps simultaneously
Decoupling Valve	Unable to provide the required cooling for ESS	Failure in coordination of the decoupling valves. First valve closed to battery branch. Second open to MDS branch. Due to dirt in system, component failure

Table 4.4: *Possible CC failures.*

Failed Part	Failure	Potential Cause
CAB Heater	Excess heating of CAB	Too much heat due to CAB Heater mechanical, electrical or control failure

4.2 Fault Injection In GSP Simulation

All of the failure modes that can occur in the system, shown in Section 2.2, could not be simulated due to limitations in the simulation system setup. The failures which could not be emulated and the reasoning for the same are listed below:

1. The leakage of the coolant from the system could not be simulated as the GSP simulation crashes if the coolant flow through the entire cooling system does not satisfy the minimum mass flow requirements.

2. Excess heating of the CAB could not be simulated, as the temperature of the CAB is not modelled in the GSP simulation.

3. Decoupling valve synchronisation was an issue for Volvo when there were multiple valves in the system but is not applicable to the current layout as there is only one valve now.

The failure modes can be simulated in the GSP simulation for different types of faults. These simulated faults are listed in Table 4.5 to demonstrate how failures in the actual system can be simulated in the GSP simulation.
All cases from Table 4.5 as well as the no fault case, are simulated for the conditions explained in Table 4.6. The faults in Table 4.5 were injected into the GSP simulation separately, using a MATLAB script file. This script is used tp change different variable values before running the simulation to emulate a particular failure. For example, to emulate the low flow failure mode for the MDS pump, the variable MDS_Pump_Degr is assigned a value of 0.01 to simulate low flow from the pump. For a normal flow this variable has a value of 1.0.
This process is repeated for all faults listed in the Table 4.5. The plots obtained from all the fault simulations are then compared with the no fault case. The no fault case is the batch simulation run without any changes to the simulation for the drive cycle Figure 2.5.

Table 4.5: *Fault emulation.*

Failure	Fault Name	Simulation Changes	Risk Level
ESS pump is unable to provide the required cooling	ESS_Pump_Degr	Degraded flow setup via reduced battery pump speed	Low
ESS chiller pump is unable to function at required speed	ESSChiller_Pump_ Degr	Limited pump speed via low gain value	Low
Radiator is clogged and unable to provide required cooling function	Rad_Degr	Limited heat rejection capacity of the radiator via low gain	Low
DCP_Clt has an offset	Decoup_Offset	Added a constant offset to the temperature signal	High
ESS heater is limited in heating power output	ESSHeater_Degr	Limited the power from the heater via low gain	High
Loss of cooling function due to low flow from MDS pump	MDS_Pump_Degr	Limited speed of MDS pump	Possible SC
Loss of cooling function due to low flow from MDS pump and ESS	MDS_Pump_Degr and ESS_Pump_Degr	Limited speed of MDS and ESS pumps	Possible SC

Table 4.6: *Parameters used in the fault injection simulation in Table 4.5.*

Ambient Temperature (°C)	-15, 0, 25, 35, 45
Initial Battery Temperature (°C)	25

4.3 Effects of Fault Injection

The simulated faults and their effects on the thermal system are described in this section. Not all of the faults listed in Table 4.5 resulted in severe system problems that need to be handled by a system diagnostic function.
The following faults were not considered for developing the diagnostic strategy:

1. ESS_Pump_Degr and ESSChiller_Pump_Degr: These faults did not lead to any significant system effects on the temperature of the coolant or the battery. This is possibly due to the MDS pump being powerful enough to drive the coolant flow through the ESS and MDS loops on its own.

2. ESSHeater_Degr: The battery heater being degraded did not affect the operation of the battery nor the temperature of the battery coolant to a significant extent. This is likely due to the battery being used extensively, which leads to a sufficient heat generation in itself. Thus, the challenge is to cool the system in most cases except in extremely low (freezing) temperatures. Therefore the heater degradation did not lead to severe effects in the simulations at the simulated battery temperature of 25°C.

3. MDS_Pump_Degr and ESS_Pump_Degr: This fault led to similar effects as in the case of MDS_Pump_Degr. These effects are described in the next part of this section.

The faults with the most severe system effects are :

1. MDS_Pump_Degr: MDS pump failure leading to low flow in the system.

2. Decoup_Offset: Temperature offset on the DCP_Clt sensor.

3. Rad_Degr: Radiator Degradation.

4.3.1 MDS Pump Fault

The flow from the MDS pump is degraded in this fault injection and this is achieved by reducing the rotating speed of the pump. In the simulation system, the pump is modelled such that the mass flow in the system is dependent on the incremental pressure difference before and after the pump and the speed of the pump. Limiting the speed of the pump reduces the flow from the pump proportionally. The fault in the system is generated by setting the speed to 1 percent of the actual speed. This limits the flow to a low value.
Figure 4.1a and Figure 4.1b illustrate the flow and speeds of the MDS pump in the no fault and low flow cases respectively. The speed for the no fault case is consistent at 6500 rpm while for the fault case, it has reduced to 65 rpm. This in turn reduces the flow fore the fault case. The flow in the no fault case is about 20 litres/min but almost 0.3 litres/min in the low flow fault case. The fault emulation works as it should by limiting the flow to almost 1 percent of normal flow.

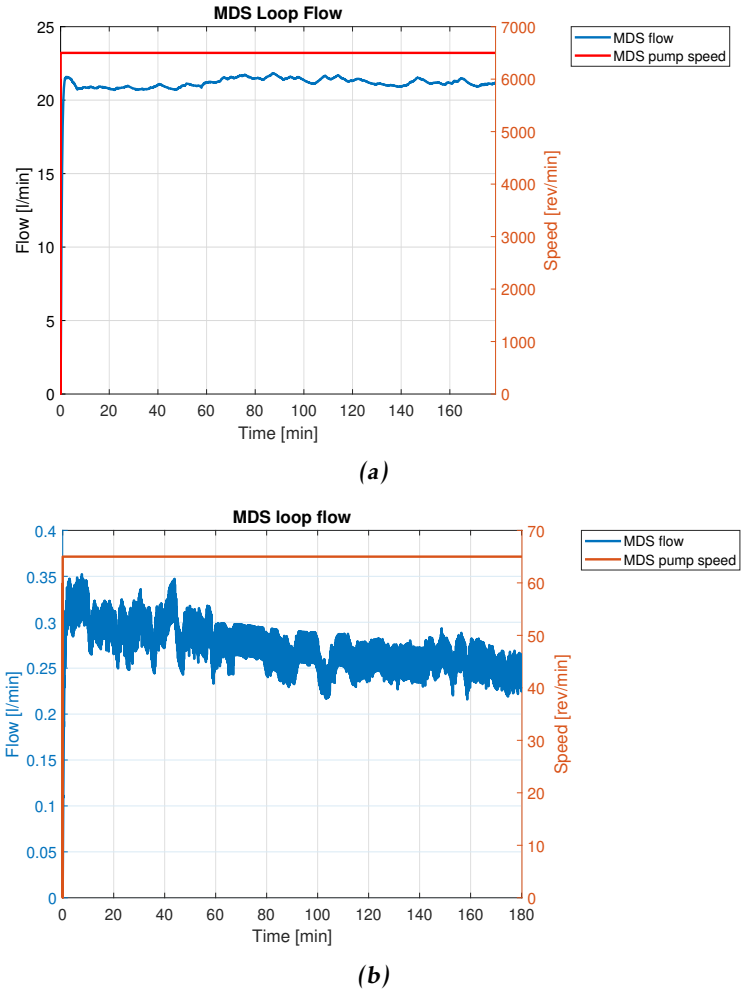
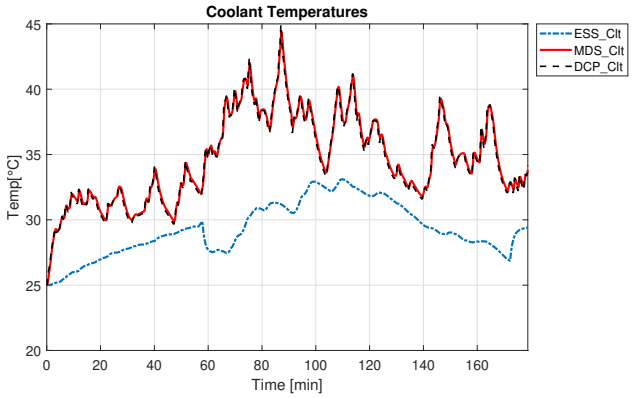


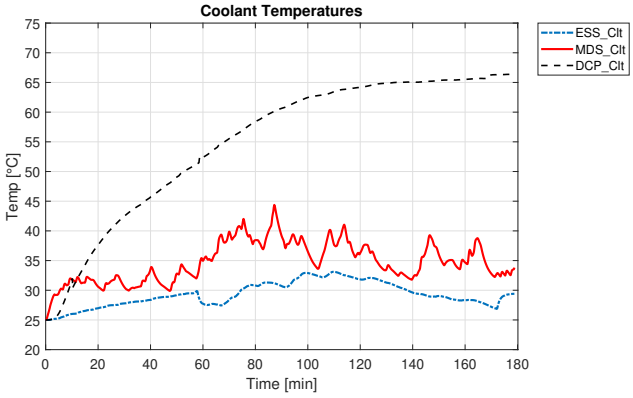
Figure 4.1: *MDS loop flow comparison. (a) No fault. (b) MDS pump degraded (reduced to 1% of normal flow).*

The coolant temperatures for the no fault case and the reduced pump speed case are shown in Figure 4.2a and Figure 4.2b, respectively.

Figure 4.2a shows DCP_Clt and MDS_Clt at almost the same temperature throughout the simulation cycle, while the ESS_Clt temperature has a lower value. Figure 4.2b shows the effect of reduced flow from the MDS pump. This reduced flow causes the DCP_Clt temperature to rise rapidly to a very high value, peaking at around 67 °C . This is in contrast with the peak DCP_Clt temperature in the no fault case, which peaks around 45 °C . The MDS_Clt temperature is similar to the DCP_Clt temperature for the no fault case and significantly lower than the DCP_clt temperature for the fault case (around 45 °C for both cases).
The ESS_Clt temperature is between 25 to 30 °C . For both the no fault and low flow case, the decoupling valve is closed and the system is in decoupled state for the entire simulation time. The ambient temperature for both figures is 25 °C. As the ambient temperature increases, the average DCP_Clt temperature also increases to a higher value and vice versa for a lower ambient temperature.



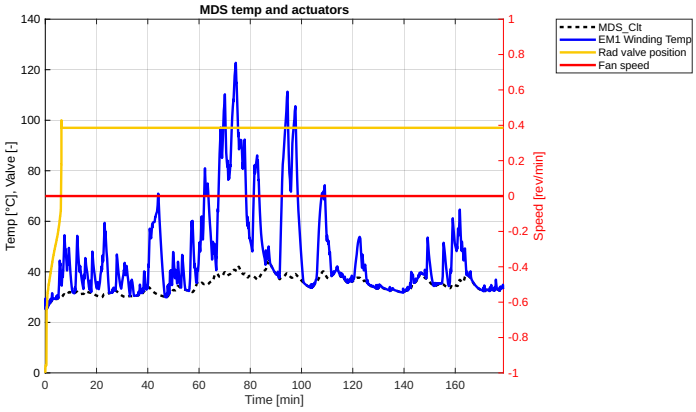
(a)



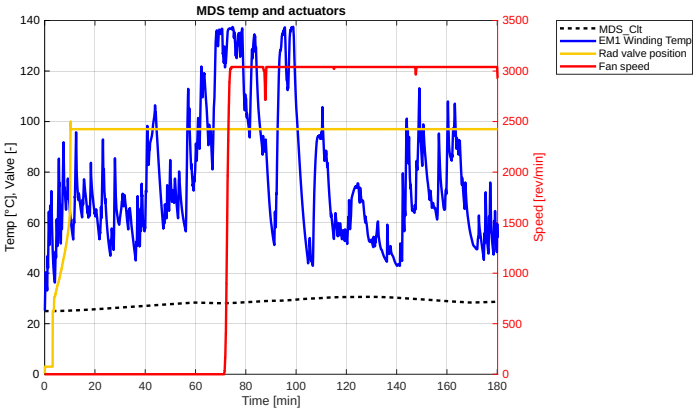
(b)

Figure 4.2: *Coolant temperatures comparison, when decoupling valve is closed. (a) No fault. (b) MDS pump degraded (reduced to 1% of normal flow).*

A low flow in the MDS loop causes a problem with the cooling of the motors in the vehicle. This ineffective cooling leads to high temperature in motor windings and the fan in the thermal system has to operate in case of the fault to cool the windings. The fan speed is zero in Figure 4.3a throughout the simulation time and at 3000 rpm in Figure 4.3b at around 70 minutes. In Figure 4.3b, the peak temperature of the electric motor 1 winding (EM1 winding temp) is around 130 °C and the average temperature is high (around 100 °C). In Figure 4.3a the average temperature of the winding is lower, at around 85 °C , and the peak temperature is lower. The average as well as the peak temperature of the motor windings in the no fault case is lower than the fault case.



(a)



(b)

Figure 4.3: *MDS Loop flow comparison. (a) No fault. (b) MDS pump degraded (reduced to 1% of normal flow).*

Table 4.7 shows no fault and MDS degraded flow of the average, peak DCP_Clt and electric windings temperatures with different ambient temperatures. As can be seen from the trend, the DCP_Clt and motor winding temperatures increase as the ambient temperature is increased. This shows that for environments with higher ambient temperatures, a MDS degraded flow can cause significant overheating in electric motor windings and heat the DCP_Clt. The fan will need to be operated for a longer time to reduce high temperature, leading to inefficient thermal system operation. In a worst case scenario this fault can lead to melting the motor which is a severe safety risk.

MDS degraded flow needs to be detected as soon as it occurs to reduce the damage it may cause to the thermal system.

Table 4.7: *Comparison between average and peak of DCP_Clt and electric winding coolant temperature for different ambient temperatures.*

Ambient Temperature (°C)	**No fault (°C)**	**MDS degraded flow (°C)**
Average DCP_Clt Temperature		
-15	24.79	37.33
25	34.61	55.26
35	46.23	63.82
45	54.13	64.25
Peak DCP_Clt Temperature		
-15	30.64	52.17
25	44.95	66.45
35	57.12	73.44
45	58.19	71.84
Average Electric Winding Coolant Temperature		
-15	30.30	64.01
25	46.35	77.09
35	57.12	79.09
45	66.63	86.14
Peak Electric Winding Coolant Temperature		
-15	86.87	136.16
25	122.73	137.49
35	134.06	137.59
45	134.90	138.37

4.3.2 Decoupled offset

This sensor fault introduces an offset on the DCP_Clt sensor. The fault is simulated by adding a constant value offset to the DCP_Clt sensor signal in the simulation. The radiator valve in the thermal system, presented in Figure 2.1, opens and closes based on a setpoint decided based on DCP_Clt temperature. This fault will cause an issue with the opening and closing of the radiator valve.
Depending on the offset type (negative or positive), it can affect when the thermal system starts to heat or cool the system. A positive offset on the DCP_Clt sensor will cause the VHPCU to begin cooling the coolant earlier or heat it later compared to no fault case. For a negative offset the VHPCU will start cooling the coolant later or heat it earlier than in the no fault case.

To illustrate the effect of a negative offset on DCP_Clt sensor, a -10°C offset on the sensor has been simulated, as presented in Figure 4.4. The effect is increasing the MDS_Clt temperature (when the valve is closed) compared to no fault case. During the simulation the decoupling valve opens after about 60 minutes, which is when the ESS_Clt temperature reaches 30°C. This leads to a rise in the ESS_Clt temperature, due to mixing of the ESS_Clt and MDS_Clt. This is problematic since the batteries will be heated up, causing issues with their operation and reduced cycle life.

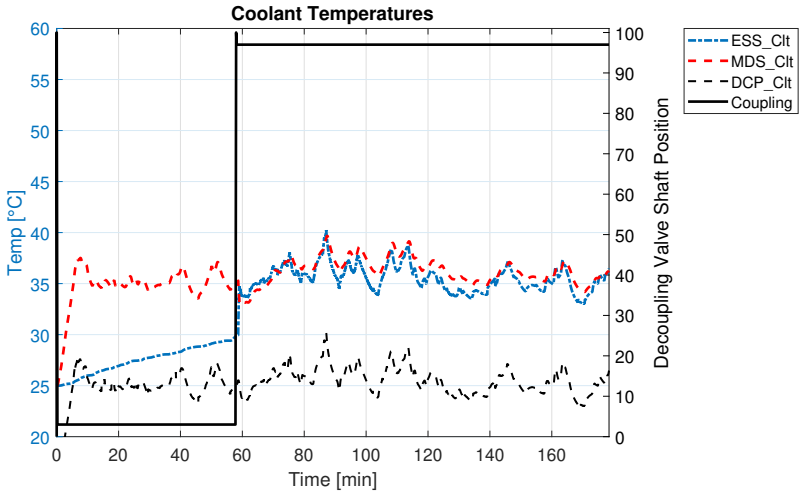


Figure 4.4: *-10 °C offset on DCP_Clt, ambient temperature 25 °C.*

An offset of +10 °C on DCP_Clt sensor leads to decoupling of the ESS and MDS loops from the start, as seen in Figure 4.5. The +10 °C causes the temperature setpoint of the decoupling valve to change such that the valve can never be coupled and the ESS and MDS loops are always separated. This leads to an inefficient heating operation as in cases when battery heating maybe required, the residual heat from the MDS coolant cannot be used to heat the ESS coolant when the 2 loops are separate.

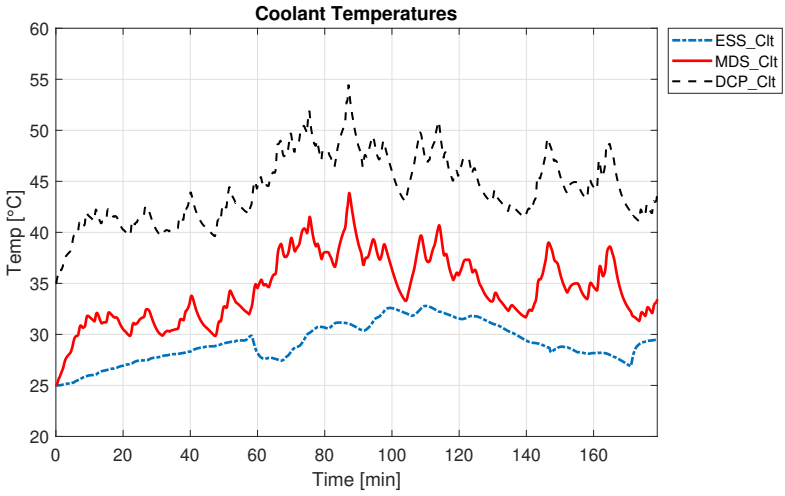


Figure 4.5: *10 °C offset on DCP_Clt, ambient temperature 25 °C.*

A summary of the effects of the offset on the DCP_Clt sensor is presented in the Table 4.8. For an increasing ambient temperature, the negative offset on the sensor, causes an increase in average MDS_Clt as well as ESS_Clt temperatures compared to their respective no fault cases. The peak MDS_Clt temperature value is also higher for this negative offset. This is mainly seen in average and peak temperatures of the ESS_Clt for ambient temperatures of -15 and 25 °C . These effects are reduced for the ambient temperatures of 35 and 45 °C as for these higher temperatures the ESS and MDS loops are decoupled according to the pre-existing control strategy. This fault affects how the thermal system operates directly as it affects the setpoint of operation for the radiator and decoupling valves. It leads to an inefficient operation of the thermal system as well as problems with the control strategy for appropriate thermal system operation.

Table 4.8: *Comparison between average and peak of MDS_Clt and ESS_Clt for different ambient temperatures.*

Ambient Temperature (°C)	**No fault (°C)**	**Decoupled offset -10°C (°C)**	**Decoupled offset 10°C (°C)**
Average MDS_Clt Temperature			
-15	25.86	33.61	17.49
25	34.97	36.26	34.94
35	46.54	48.63	44.17
45	54.58	59.15	49.44
Peak MDS_Clt Temperature			
-15	31.27	31.27	24.72
25	45.19	45.19	45.19
35	57.79	57.79	48.23
45	58.50	58.50	52.63
Average ESS_Clt Temperature			
-15	26.92	31.59	26.48
25	28.30	32.88	29.19
35	28.48	29.50	29.52
45	29.22	30.23	30.69
Peak ESS_Clt Temperature			
-15	30.02	36.30	30.13
25	31.39	40.21	32.79
35	31.84	33.25	33.29
45	32.84	34.20	35.00

4.3.3 Radiator Degradation

The radiator degradation fault is when the radiator is not able to provide adequate heat rejection. In GSP, this is achieved by reducing the heat rejection capacity of the radiator. In the physical system this fault is akin to the blockage of radiator vents with dust or other impurities.
When the radiator is degraded it has a significant impact on the temperature of the coolants in the thermal system. The DCP_Clt and MDS_Clt temperatures increased to a very high value due to insufficient heat dissipation via the radiator. The decoupling valve sets to decoupled state, so the coolant between the ESS and MDS loop is not mixed. This safeguards the batteries from being exposed to a hot ESS_Clt. The fan starts operating for the majority of the drive cycle. These aforementioned effects can be seen in Figure 4.6.

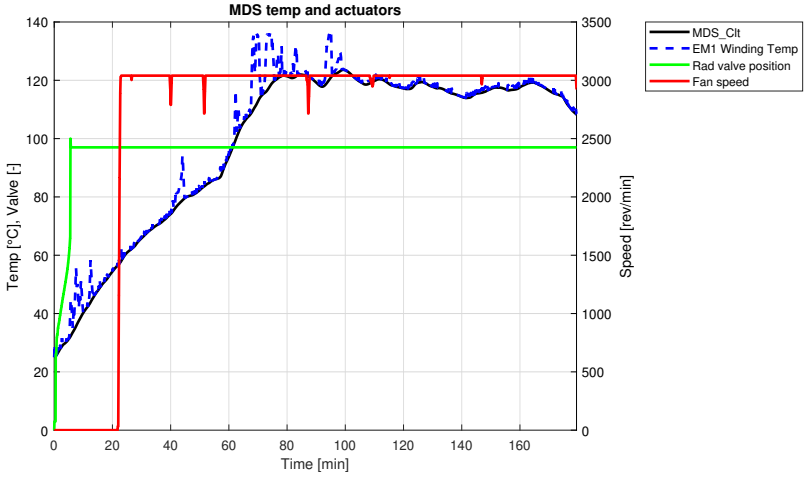


Figure 4.6: *Radiator degraded, ambient temperature 25°C. Decoupling valve is closed, meaning ESS and MDS loop are flowing separated.*

Radiator degradation causes the thermal system to be overloaded, unable to provide the necessary cooling to the vehicle. Table 4.9 presents the trend of the average and peak DCP_Clt and electric motor winding temperatures with different ambient temperatures. The temperatures of the DCP_Clt and motor winding increase as the ambient temperature is increased. In the radiator degraded case the average DCP_Clt temperature values are significantly higher than in the no fault case. The average electric winding temperature is much higher in the radiator degraded case compared to no fault case.

Table 4.9: *Comparison between average and peak of DCP_Clt and electric motor winding coolant temperature for different ambient temperatures.*

Ambient Temperature (°C)	**No fault (°C)**	**Radiator degraded (°C)**
Average DCP_Clt Temperature		
-15	24.79	78.56
25	34.61	99.96
35	46.23	107.45
45	54.13	112.95
Peak DCP_Clt Temperature		
-15	30.64	115.56
25	44.95	125.20
35	57.12	128.63
45	58.19	130.93
Average Electric Winding Coolant Temperature		
-15	30.30	78.71
25	46.35	100.46
35	57.12	107.82
45	66.63	113.17
Peak Electric Winding Coolant Temperature		
-15	86.87	115.03
25	122.73	136.66
35	134.06	137.36
45	134.90	137.95

5

Methodology: Design of a System Diagnostic Function

The main focus of the diagnostic strategy is on the specific faults that had the most severe system effects as mentioned in the previous chapter.
The radiator degradation and MDS pump failure faults can and should only be detected if the temperature sensors in the system do not have an offset error. Offset error in this case means that the temperature reported is either higher or lower than the actual temperature of the coolant.

In the system diagnostic algorithm, the first step is to diagnose if there is an offset on the temperature sensors or not. Once this has been confirmed, further steps can be taken to assess if a fault exists in the operation of the radiator or the flow of coolant through the system.

5.1 Fault Detection Logic

As discussed previously, a temperature offset can affect the radiator valve operation because the radiator valve opens and closes on the basis of the temperature value of the DCP_Clt sensor. Hence, a too low temperature value can cause the radiator valve to open more than necessary and vice versa. Additionally the decouple valve opens and closes based on the temperature targets set by the ESS_Clt and MDS_Clt sensors. Therefore, an offset on these sensors leads to improper operation of the decouple valve. This further causes problems with the cooling function in the system. This section investigates how such a temperature offset can be detected. The following insights on how the system behaves form the basis of the diagnostic model showcased in the next section.

To build a fault detection logic, the system's effect is studied via a comparison between available temperature sensors in Figure 2.1. The sensors that are compared are:

- MDS_Clt and the DCP_Clt.
- ESS_Clt and the DCP_Clt.

The fault logic detection is based on the assumption that the coolant temperature increase over a coolant pump is small. Therefore it is possible to compare MDS_Clt and DCP_Clt sensors when the decoupling valve is closed. This is due to the fact that the coolant flows directly through both sensors, with only the MDS pump in between. This is also illustrated in Figure 2.2, by examining the MDS Loop and tracing the flow from DCP_Clt to MDS_Clt sensor. Figure 5.1 illustrates that |MDS_Clt - DCP_Clt| is small when the decoupling valve is closed (0 to 3 %, in decoupled state) and under no fault case.

For the case when the decoupling valve is set to open (97 to 100%, in coupled state), the coolant will flow through the DCP_Clt and ESS_Clt, with only the ESS pump in between. Therefore it is possible to compare the temperature sensors ESS_Clt and DCP_Clt. Figure 5.2 illustrates that |ESS_Clt - DCP_Clt| is small when the decoupling valve is open and under no fault case. Both Figure 5.1 and Figure 5.2 are obtained through simulations at an ambient temperature set to -15 °C, but similar results were achieved for the ambient temperatures of 25, 35, 45 °C.

The same logic as in the no fault case will still hold for an offset on the DCP_Clt sensor. This means that the calculation |MDS_Clt - DCP_Clt| is done when the decoupling valve is closed and |ESS_Clt - DCP_Clt| is done for the open decoupling valve. The contrast between the no fault case and sensor fault case is that the absolute difference will be close to the added offset value for the fault case (instead of close to zero). This effect can be seen in the simulated case with an -10°C on DCP_Clt sensor. When the decoupling valve is set to closed, the value of |MDS_Clt - DCP_Clt| shows an offset of approximately 10 °C, as illustrated in Figure 5.3.

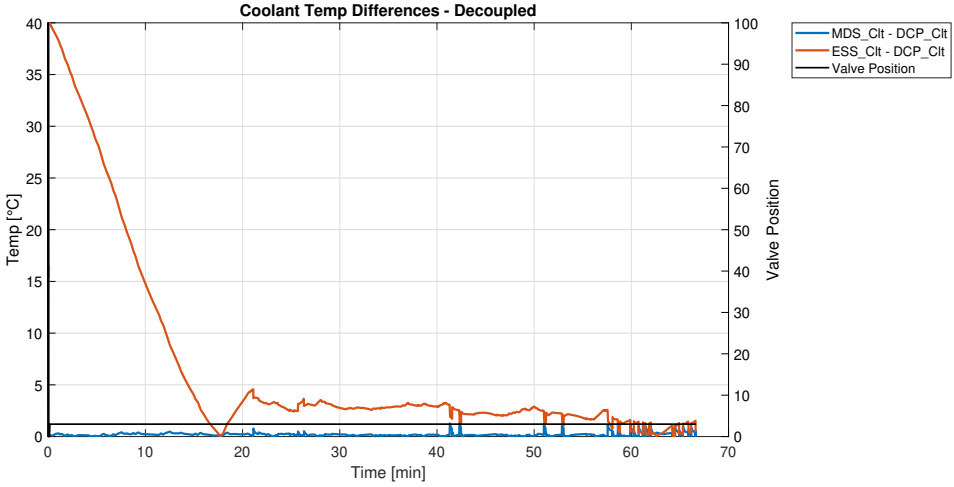


Figure 5.1: *Absolute difference in coolant temperatures for the decoupled state (decoupling valve is closed) in the no fault case. Since the decoupling valve is closed, the relevant signal is |MDS_Clt - DCP_Clt|, which remains close to zero throughout this period.*

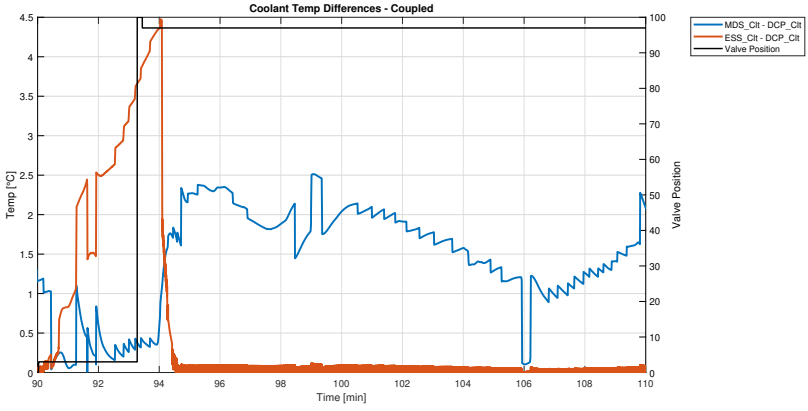


Figure 5.2: *Absolute difference between coolant temperatures under no-fault conditions. The key signal to observe is |ESS_Clt - DCP_Clt| when the decoupling valve is open. Note the initial spike for the signal |ESS_Clt - DCP_Clt| when the decoupling valve transitions from open to closed.*

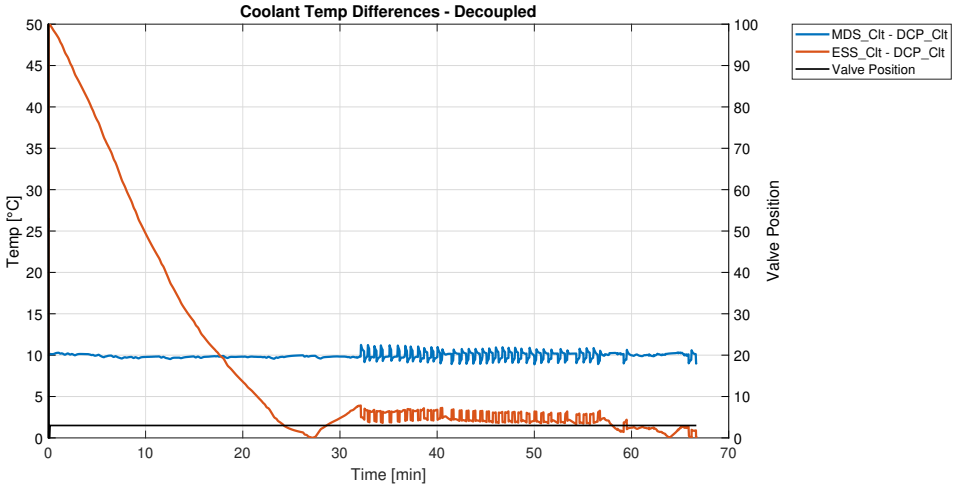


Figure 5.3: *Absolute difference in coolant temperatures for the decoupled state (decoupling valve is closed) for a -10 °C offset on the DCP_Clt. Particularly look at the signal |MDS_Clt - DCP_Clt|, which is close to a 10°C offset.*

When the decoupling valve is set to open, the value of |ESS_Clt - DCP_Clt| shows an offset of approximately 10 °C, as shown in Figure 5.4.

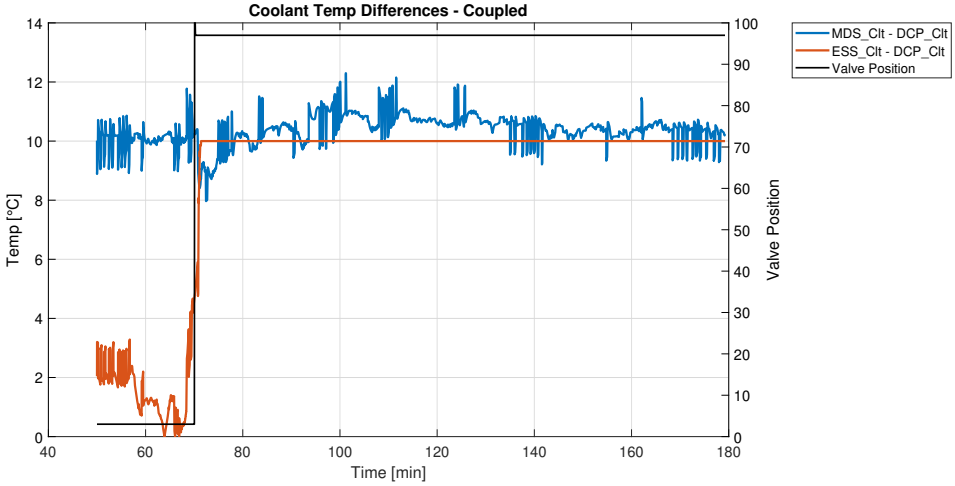


Figure 5.4: *Absolute difference in coolant temperatures for the coupled state (decoupling valve is open) for a -10 °C offset on the DCP_Clt. The important point is that |ESS_Clt - DCP_Clt| is close to the expected 10°C offset.*

5.1.1 Detection of Other Faults

As seen in Figure 4.3b, for the low flow in the MDS pump, the decoupling valve is closed and there is a large temperature difference between DCP_Clt and MDS_Clt temperatures. This means that according to the logic of looking at temperature differences between the various temperature sensors, a fault will be flagged in both the case of the offset on DCP_Clt sensor and low flow in MDS pump. However, further logic can be added to avoid this as the temperature difference seen in Figure 4.3b is far more than the temperature difference in Figure 4.5. Hence a logic based on adjusting the threshold above which the fault is declared can be used to differentiate the two faults.

5.2 Diagnostic Model

The diagnostic model is shown in Figure 5.5. This model is divided into multiple subsystems and is made to detect the offset in the three temperature sensors, decoupling coolant temperature sensor, ESS coolant temperature sensor and the MDS coolant temperature sensor.

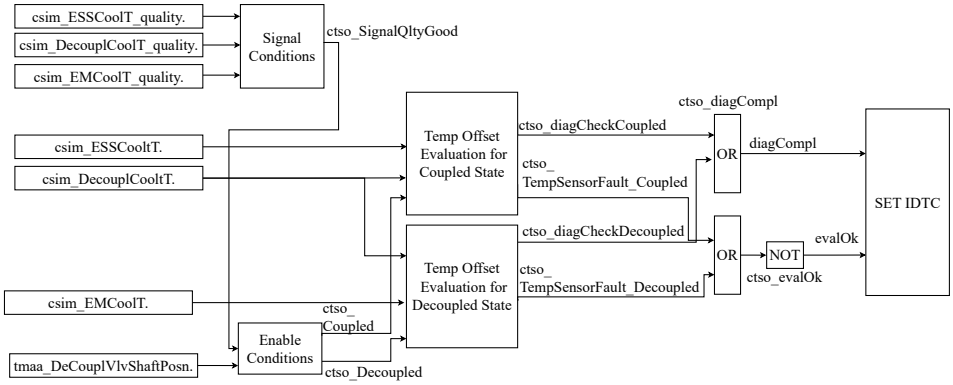


Figure 5.5: *Diagnostic model overview for temperature offset fault.*

The layout is divided into signal conditions, enable conditions, evaluation and finally the diagnostic test code flag. The parameters used in the following functions are enumerated below and their assigned values are listed in Table 5.1.

- ctso_DCPL_VLV_SHAFT_MAX: Maximum value for the decouple valve shaft position.
- ctso_DCPL_VLV_SHAFT_MIN: Minimum value for the decouple valve shaft position.
- ctso_DCPL_VLV_SHAFT_CPLD: Value of shaft position above which the decouple valve is considered to be in coupled state.
- ctso_DCPL_VLV_SHAFT_DCPLD: Value of shaft position below which the decouple valve is considered to be in decoupled state.
- ctso_CPL_TO_DCPL_EVAL_TIME: Time delay to allow for decoupled to coupled state transition.
- ctso_DCPL_TO_CPL_EVAL_TIME: Time delay to allow for coupled to decoupled state transition.
- ctso_TEMP_OFFSET_EVAL_TIME: Time delay to set the fault only if the temperature difference persists for a set amount of time.
- ctso_TEMP_OFFSET_EVAL_LIM: Temperature difference threshold for fault setting.

Table 5.1: *Variable definition for diagnostic model.*

Variable	Value	Unit
ctso_DCPL_VLV_SHAFT_MAX	100	%
ctso_DCPL_VLV_SHAFT_MIN	0	%
ctso_DCPL_VLV_SHAFT_CPLD	97	%
ctso_DCPL_VLV_SHAFT_DCPLD	3	%
ctso_CPL_TO_DCPL_EVAL_TIME	20	s
ctso_DCPL_TO_CPL_EVAL_TIME	20	s
ctso_TEMP_OFFSET_EVAL_TIME	30	s
ctso_TEMP_OFFSET_EVAL_LIM	4	°C

5.2.1 Signal Conditions

The signal conditions presented in Figure 5.6 focus on the quality of the signals used for diagnosis.

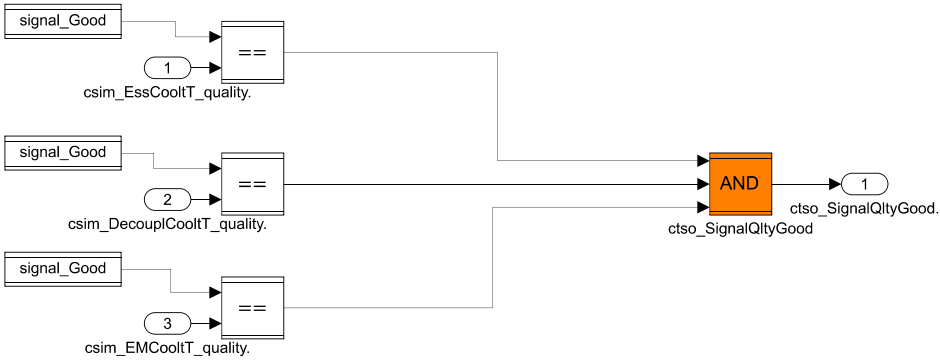


Figure 5.6: *Signal conditions block.*

If all three input signals in Figure 5.6 satisfy appropriate quality conditions, then the output of this block will be true, and hence the fault diagnosis can proceed to the next step. This true output is given to the outport of the signal condition block to be used later in other conditional blocks.

5.2.2 Enable Conditions

The enable condition block seen in Figure 5.7 is the pre-check block which tells the diagnostic model when to operate and look for the fault conditions.
The initial checks in this block verify if the decoupled valve is in the coupled or the decoupled state. This is checked by checking the shaft position of the valve. The conditional checks are summarised as follows:

1. If the shaft position is below the ctso_DCPL_VLV_SHAFT_MAX value and more than the ctso_DCPL_VLV_SHAFT_CPLD value, then the valve is considered to be in the coupled state.

2. If the shaft position is above the ctso_DCPL_VLV_SHAFT_MIN value and less than the ctso_DCPL_VLV_SHAFT_DCPLD value, then the valve is considered to be in the decoupled state.

The shaft position is checked to ensure that the coupled and decoupled outputs are only triggered when it is absolutely certain that the valve is in one of these two states. The signal condition is a pre-check to ensure no issue with the signals while the enabling conditions are being checked.

From Figure 5.2, each transition from decoupled to coupled state (that is, when the decoupling valve changes from open to closed) is followed by a settling period before the difference |ESS_Clt - DCP_Clt| becomes small. During this settling time, |ESS_Clt - DCP_Clt| show a high peak, which can lead to mis-triggering of the threshold for the fault conditions in the evaluation block. Similar effect can be seen for the |MDS_Clt - DCP_Clt| when transition from coupled to decoupled state. Thus to avoid this false triggering, time delays
(ctso_CPL_TO_DCPL_EVAL_TIME and ctso_DCPL_TO_CPL_EVAL_TIME) are included before setting the outputs of this block as true. This ensures that enough delay is considered to avoid enable condition checking during the transition time from the decoupled to coupled state and vice versa.

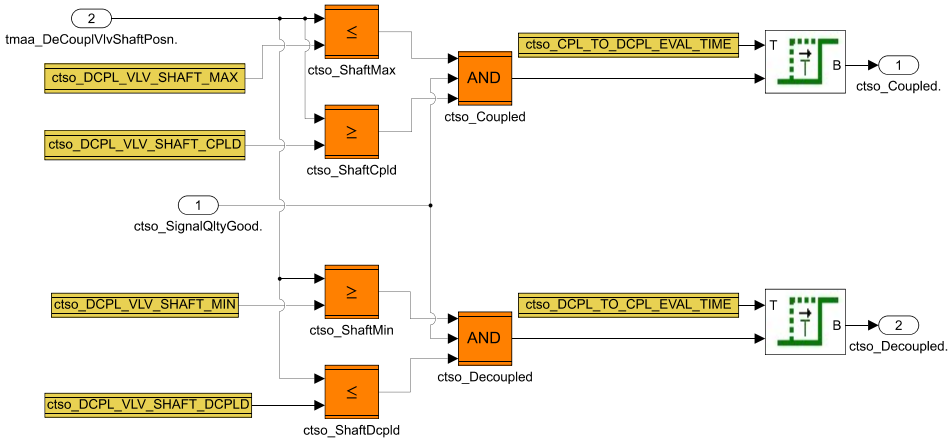


Figure 5.7: *Enable condition block.*

5.2.3 Coupled Case Fault Diagnostic

This block is presented in Figure 5.8 and focuses on the condition when the loops are coupled and the decoupled valve is open. The block gets the coupled check (ctso_Coupled) input from the evaluation block as well as the DCP_Clt (csim_DecouplCooltT) and ESS_Clt (csim_ESSCooltT) temperatures. The absolute difference between these temperatures is calculated. If it exceeds a threshold value (ctso_TEMP_OFFSET_EVAL_LIM) and the valve is still in the coupled state, the fault output (ctso_TempSensorFaultCoupled) is set to be true.

A further diagnostic evaluation is performed to determine if and when the fault should be declared. Two samples of the conditional signal are checked to ensure if a fault is present. If the coupled state has lasted for a particular amount of time delay (ctso_TEMP_OFFSET_EVAL_TIME) and the fault has been present for multiple samples (through the duration of the time delay), the diagnostic evaluation will be true and the fault can be reported.

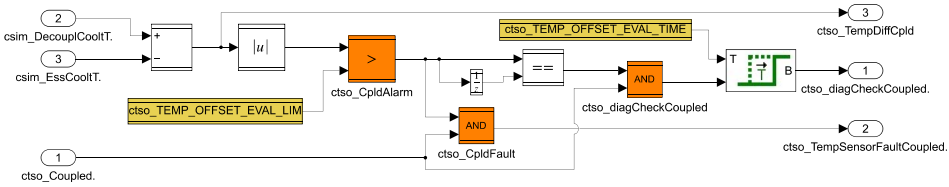


Figure 5.8: *Evaluation block for coupled state.*

5.2.4 Decoupled Case Fault Diagnostic

The same logic as in the coupled case (see Subsection 5.2.3) will be applied here. However, the key difference is that it compares the signals csim_DecouplCoolt and csim_EMCoolt, and uses csto_Decoupled instead of ctso_Coupled as the input. The label for the corresponding logic block is updated for the decoupled case. The logic is presented in Figure 5.9.

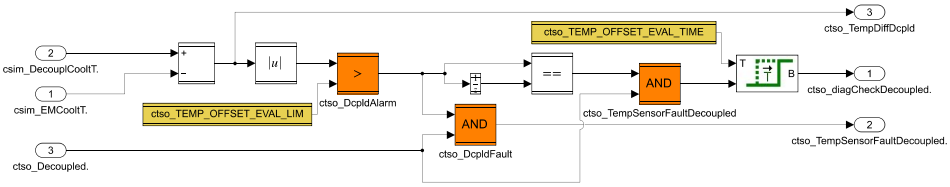


Figure 5.9: *Evaluation block for decoupled state.*

5.2.5 Diagnostic Evaluation

The final block in the model is the diagnostic evaluation block presented in Figure 5.5. This block is used to finally report if a fault has occurred once the diagnostic evaluation is completed.

The evalOk flag in this block signifies whether there is a fault or not. If this flag is 0, the fault has occurred and if it is 1 the fault has not occurred. This flag is reported only when the diagCompl flag is true in the block. As seen in Figure 5.5, the diagnostic completed check can be triggered by any one of the decoupled or coupled evaluation blocks. Even if one of the offsets lead to triggering of an error, the evaluation check will fail and the diagnostic test will report a fault in the system.

6

Diagnostic Function Test

A testing strategy was developed to ensure that identified cases of failure for the diagnostic function are accounted for. This includes running the diagnostic function and checking the operation of all condition blocks presented in Section 5.2. The testing strategy is performed on a test bench setup as seen in Figure 6.1.

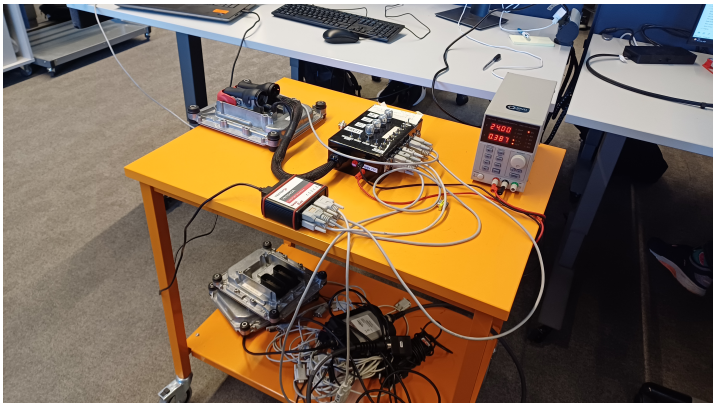


Figure 6.1: *Test bench setup.*

The tests are validated by observing the boolean values of the condition blocks described in the diagnostic model. The values for the parameters in the model for the test are the same as defined in Table 5.1. The condition based signals and their corresponding logic are presented in Table 6.1, with 1 representing true and 0 representing false. The following flags ctso_diagCompl, ctso_evalOk and Diagnostic Test Code are used for the testing and are described as follows:

1. ctso_diagCompl: This is a diagnostic flag that if set to 1, enables reporting of current status (no fault or fault) of the diagnostic test code (IDTC) to the VHPCU. ctso_diagCompl will be set to 1 if any one of ctso_diagComplCoupled and ctso_diagComplDecoupled is 1. Every time there is a change in the values of these two inputs, the ctso_diagCompl will be set to 0.

2. ctso_evalOk: This diagnostic flag reports the actual occurrence of a fault. The evaluation is decided jointly by TempSensorFaultCoupled and TempSensorFaultDecoupled. If one of these two values is 0, the flag is set to 0. If the evaluation is ok and no fault has occurred, the flag is set to 1 and for a fault case, the flag is set to 0.

3. Diagnostic Test Code: This is the final result from the diagnostic function. It is triggered (set to 1) when the ctso_evalOk flag is set to 0. It is an indicator of failure of the diagnostic test. It serves as an indicator of fault occurrence to the VHPCU.

Table 6.1: *Evaluation Flags.*

Flag & Description	True (1)	False (0)
ctso_SignalQltyGood Signal Quality Evaluation	All three temperature signals are available	Not all temperature signals are available
ctso_Coupled State of decoupling valve	Coupled state	Not coupled state
ctso_Decoupled State of decoupling valve	Decoupled state	Not decoupled state
ctso_CpldFault Temperature difference |DCP_Clt - ESS_Clt|	> 4 °C	< 4 °C
ctso_DcpldFault Temperature difference |DCP_Clt - MDS_Clt|	> 4 °C	< 4 °C
ctso_diagCheckCoupled ctso_CpldAlarm consistency across samples	Same value across multiple samples, for evaluation time of 20 s	Different values across multiple samples
ctso_TempSensorFaultCouple Set fault flag, depending on	ctso_CpldFault = 1 AND ctso_Coupled = 1	ctso_CpldFault = 0 OR ctso_Coupled = 0
ctso_diagCheckDecoupled ctso_DcpldAlarm consistency across samples	Same value across multiple samples, for evaluation time of 20 s	Different values across multiple samples
ctso_TempSensorFaultDecoupled Set fault flag, depending on	ctso_DcpldFault = 1 AND ctso_Decoupled = 1	ctso_DcpldFault = 0 OR ctso_Decoupled = 0

Before initiating the tests, a summary of all test cases are provided that are used to validate the diagnostic logic. These include scenarios with signal quality failure, coupled and decoupled configurations under both fault and no fault conditions, and transitions between coupled and decoupled states. An overview of these tests are presented in Table 6.2.

Table 6.2: *Summary of tests for the diagnostic function.*

Test	**Test evaluation (after evaluation time)**	**Explanation**
Test 0: Signal Quality Failure	ctso_diagComp=0	Not reporting anything to the IDTC
1: Coupled no fault	ctso_Coupled = 1 ctso_CpldFault = 0 csto_evalOk = 1 csto_diagCompl = 1	Coupled state and no fault Reporting no fault
2: Coupled fault	ctso_Coupled = 1 ctso_CpldFault = 1 csto_evalOk = 0 csto_diagCompl = 1	Coupled state and fault Reporting fault
3: Decoupled no fault	ctso_Decoupled = 1 ctso_DcpldFault = 0 csto_evalOk = 1 csto_diagCompl = 1	Decoupled state and no fault Reporting no fault
4: Decoupled fault	ctso_Decoupled = 1 ctso_DcpldFault = 0 csto_evalOk = 0 csto_diagCompl = 1	Decoupled state and fault Reporting fault
5: Coupled to decoupled transition, with intial coupled fault	csto_evalOk = 0 csto_diagCompl = 1	Test transition phase from coupled to decoupled state. Report fault
6: Decoupled to coupled transition, with inital decoupled fault	csto_evalOk = 0 csto_diagCompl = 1	Test transition phase from decoupled to coupled. Report fault

6.1 Test Methodology For Diagnostic Function

The tests focused on validating every sub block of the model as well as testing all possible scenarios observed from the fault injection in GSP simulation presented in Table 4.6. The values for the time delay variables are as mentioned in Table 5.1.

1. **Test 0, Signal Quality Failure**: Check the signal conditions block (see Subsection 5.2.1).

   (a) Check signal quality of the signals from the temperature sensors.

   (b) Set ctso_SignalQltyGood to 1 if the signal quality is good for all three signals (see Subsection 5.2.1), otherwise set it to 0.

   (c) For the subsequent conditions to be evaluated, ctso_SignalQlty needs to be 1.

   Figure 6.2 shows that ctso_SignalQlty is set to false for the entire period and hence it forces all subsequent conditions to be 0.

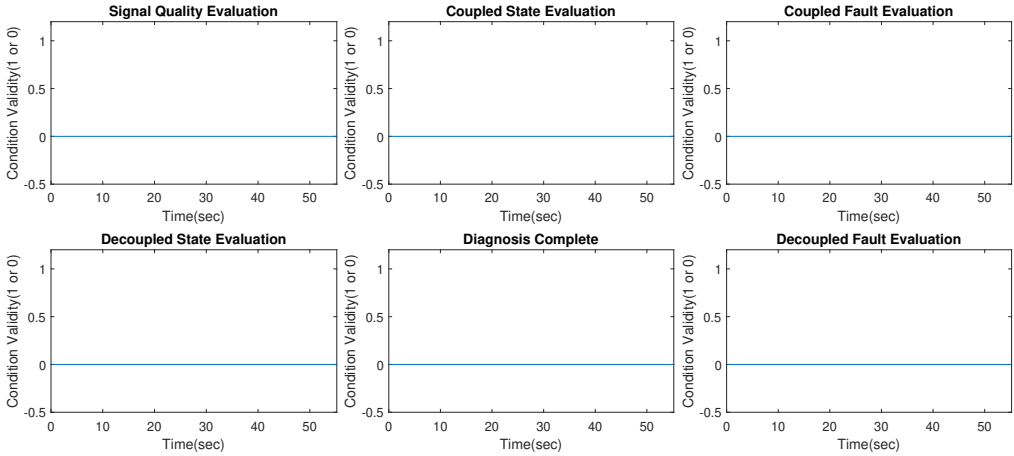


Figure 6.2: *Signal Quality is not good.*

2. **Test 1, Coupled no fault**: Check the performance of the diagnostic when the decoupling valve is open (coupled state) and there is no fault regarding the temperature sensors.

   (a) Check if decoupled shaft position is in between 97 and 100 %. If the position condition is 1, set ctso_Coupled block to 1 after ctso_CPL_TO_DCPL_EVAL_TIME (see Subsection 5.2.2).

   (b) Check the temperature difference between DCP_Clt (csim_DecouplCooltT) and ESS_Clt (csim_EssCooltT) is below the ctso_TEMP_OFFSET_EVAL_LIM. If the value is lower than the threshold, set ctso_CpldFault to 0 (see Subsection 5.2.3).

   (c) ctso_evalOk remains 1 throughout and ctso_diagCompl becomes 1 after a set delay time of ctso_TEMP_OFFSET_EVAL_TIME (see Subsection 5.2.3).

   The results for these tests are shown in Figure 6.3. The ctso_CpldFault condition is 0 throughout as the temperature difference between DCP_Clt and ESS_Clt is zero.

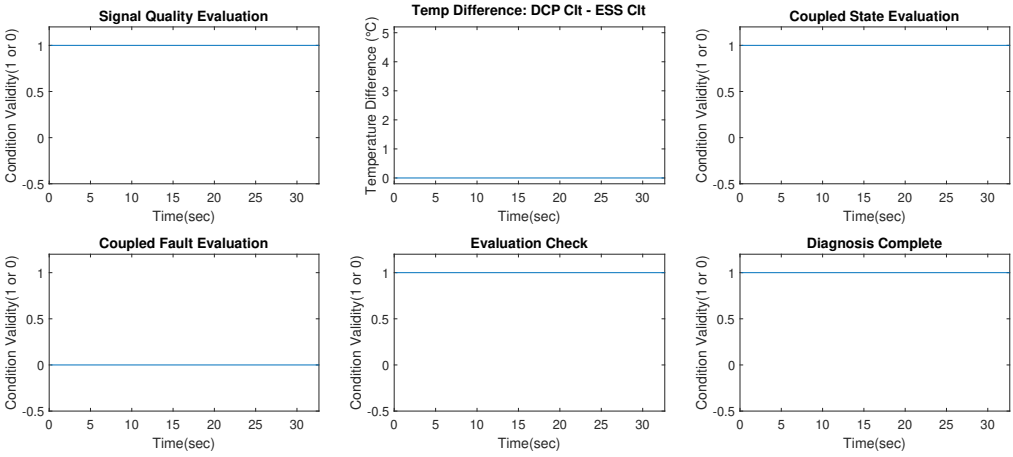


Figure 6.3: *Coupled no fault.*

3. **Test 2, Coupled Fault**: Check the performance of the diagnostic when the decoupling valve is open (coupled state) and there is a fault regarding the temperature sensors.

   (a) Check if decoupled shaft position is in between 97 and 100 %. If the position condition is 1, set ctso_Coupled block to 1 after ctso_CPL_TO_DCPL_EVAL_TIME (see Subsection 5.2.2).

   (b) Check the temperature difference between DCP_Clt and ESS_Clt is below the ctso_TEMP_OFFSET_EVAL_LIM. If the value is greater than the threshold, set ctso_CpldFault to 1 (see Subsection 5.2.3).

   (c) ctso_evalOk is 1 initially and sets to 0 for the duration of the fault and ctso_diagCompl becomes 1 after a set delay time of ctso_TEMP_OFFSET_EVAL_TIME (see Subsection 5.2.3).

   The result for this test are shown in Figure 6.4. The ctso_CpldFault becomes 1 for the duration when the temperature difference between DCP_Clt and ESS_Clt is more than 4°C. It resets to 0 when the difference becomes less than or equal to 4°C.

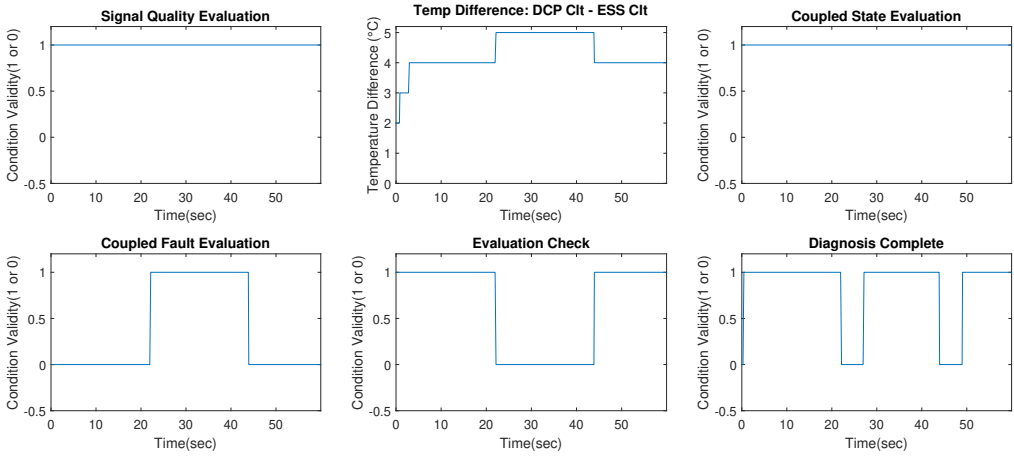


Figure 6.4: *Coupled fault.*

4. **Test 3, Decoupled no fault**: Check the performance of the diagnostic when the decoupling valve is closed (decoupled state) and there is no fault regarding the temperature sensors.

   (a) Check if the decoupled valve shaft position is between 0 and 3 %. If the position condition is 1, set the ctso_Decoupled block to 1 after ctso_DCPL_TO_CPL_EVAL_TIME (see Subsection 5.2.2).

   (b) Check if the temperature difference between DCP_Clt and MDS_Clt (csim_EMCooltT) is below the ctso_TEMP_OFFSET_EVAL_LIM. If the value is lesser than the threshold, set ctso_DcpldFault to 0 (see Subsection 5.2.4).

   (c) The output of ctso_diagCheckDecoupled becomes 1 after a set delay time of ctso_TEMP_OFFSET_EVAL_TIME and the ctso_TempSensorFaultDecoupled sets to 0 (see Subsection 5.2.4).

   The results for this test are shown in Figure 6.5. The ctso_DcpldFault is always 0 as the temperature difference between DCP_Clt and MDS_Clt never goes above the threshold of 4°C.

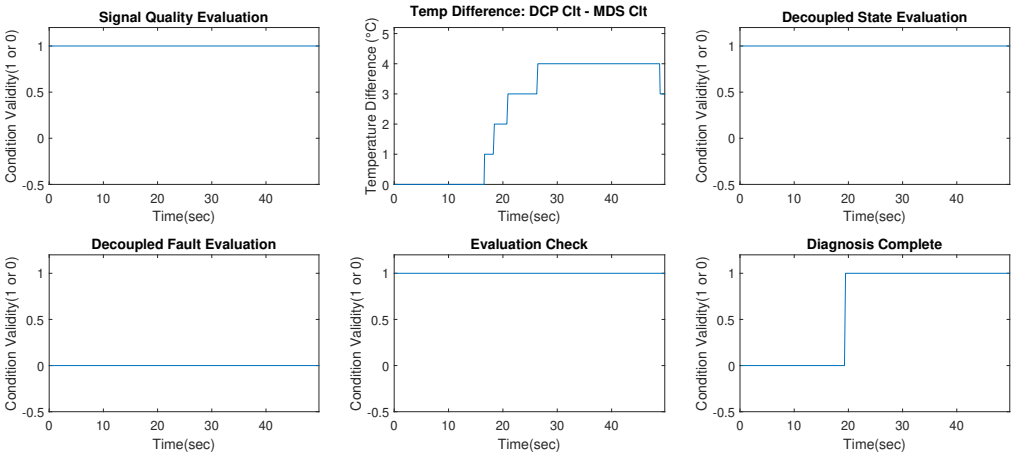


Figure 6.5: *Decoupled No Fault.*

5. **Test 4, Decoupled fault**: Check the performance of the diagnostic when the decoupling valve is closed (decoupled state) and there is a fault regarding the temperature sensors.

   (a) Check if the decoupled valve shaft position is between 0 and 3 %. If the position condition is 1, set the ctso_Decoupled block to 1 after ctso_DCPL_TO_CPL_EVAL_TIME (see Subsection 5.2.2).

   (b) Check if temperature difference between DCP_Clt and MDS_Clt is above ctso_TEMP_OFFSET_EVAL_LIM. If the value is greater than the threshold, set ctso_DcpldFault to 1 (see Subsection 5.2.4).

   (c) The output of ctso_diagCheckDecoupled becomes true after a set delay time of ctso_TEMP_OFFSET_EVAL_TIME and the ctso_TempSensorFaultDecoupled sets to true (see Subsection 5.2.4).

   The results of this test are shown in Figure 6.6. The ctso_DcpldFault becomes true for the duration when the temperature difference between DCP_Clt and MDS_Clt goes above the threshold of 4 °C.

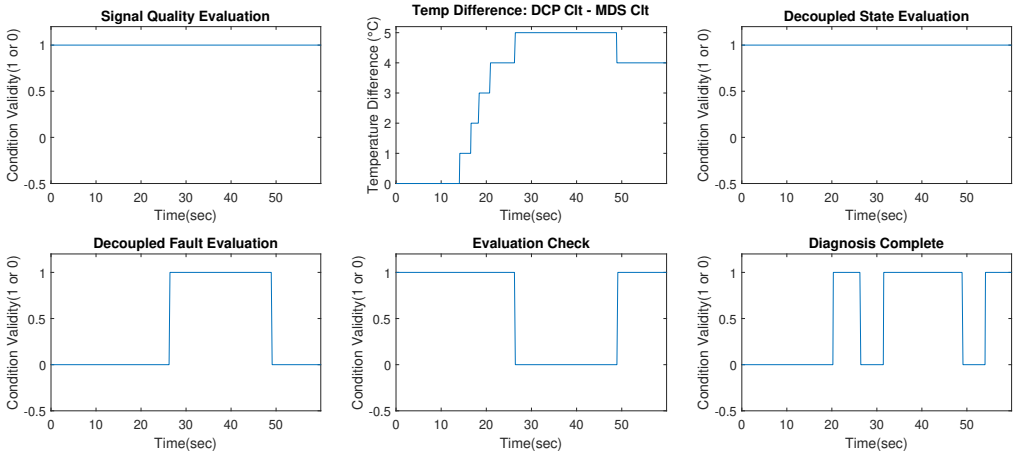


Figure 6.6: *Decoupled Fault.*

6. **Test 5, Coupled to Decoupled Transition, with Initial Coupled Fault**: Evaluate the performance of the diagnostic function during a transition phase from coupled to decoupled.

   (a) At the start of the test, the decoupling valve is in coupled state and the temperature difference between DCP_Clt and ESS_Clt is greater than the threshold. This sets the ctso_Coupled and ctso_CpldAlarm to 1 (see Subsection 5.2.3).

   (b) This means that there is a coupled fault in the diagnostic function and thus ctso_CpldFault is true. In the next stage, the decoupled valve changes its state from coupled to decoupled. This makes the ctso_Coupled input to the evaluation block false and thus ctso_CpldFault is set to false subsequently (see Subsection 5.2.3).

Figure 6.7 presents the results of this test. When the transition takes place from coupled to decoupled state, ctso_CpldFault sets to false even though the temperature difference between DCP_Clt and ESS_Clt is greater than the threshold. This shows the success of the function in handling the coupled to decoupled transition.

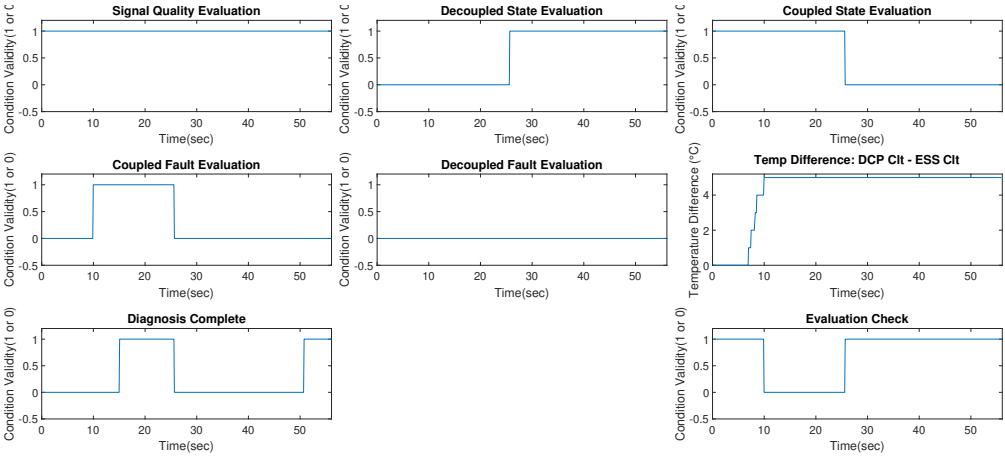


Figure 6.7: *Coupled to decoupled transition.*

7. **Test 6, Decoupled to Coupled Transition, with Initial Decoupled Fault**: Evaluate the performance of the diagnostic function during a transition phase from coupled to decoupled.

   (a) At the start of the test, the decoupling valve is in coupled state and the temperature difference between DCP_Clt and MDS_Clt is greater than the threshold. This sets the ctso_DcpldFault and ctso_DcpldAlarm sets to 1 (see Subsection 5.2.4).

   (b) This means that there is a coupled fault in the diagnostic function and thus ctso_DcpldFault is true. In the next stage, the decoupled valve changes its state from coupled to decoupled. This makes the ctso_Decoupled input to the evaluation block false and thus ctso_DcpldFault is set to false subsequently (see Subsection 5.2.4).

   Figure 6.8 presents the results of this test. When the transition takes place from decoupled to coupled state, ctso_DcpldFault sets to false even though the temperature difference between DCP_Clt and ESS_Clt is greater than the threshold. This shows the success of the function in handling the decoupled to coupled transition.

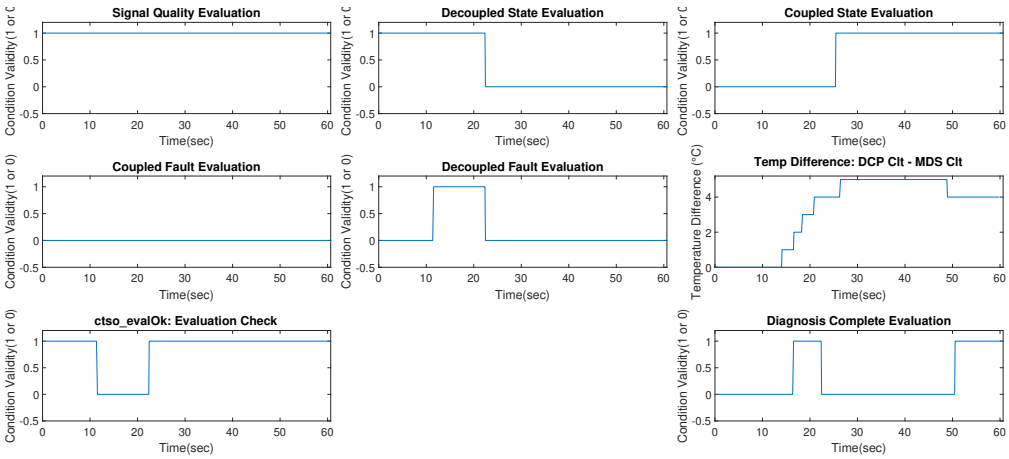


Figure 6.8: *Decoupled to Coupled Transition.*

7

Vehicle Test Results

The fault injection test and verification of the diagnostic function have been carried out on a truck at Volvo GTT. This process included setting up of the vehicle under test as well as creating a vehicle code package to upload to the VHPCU.

7.1 Vehicle Test Setup Standstill

This test is necessary to confirm if the diagnostic function can work in an actual vehicle without triggering unnecessary faults. If any function logic developed for the vehicle triggers a fault or an alarm when it is not needed, this leads to reduced productivity as well increased costs for customers. To prevent this vehicle testing is done to ensure the fault sets in very specific cases and only when absolutely necessary. The vehicle used for testing was at standstill in the testing area at Volvo GTT. The setup can be seen in Figure 7.1.

(a) ***(b)***

Figure 7.1: *(a) Electric truck used for testing. (b) Test setup for electric truck.*

The test vehicle was an electric truck with the latest thermal system installed as shown in Figure 2.1. A representation of the connections used during the testing phase are presented in Figure 7.2.

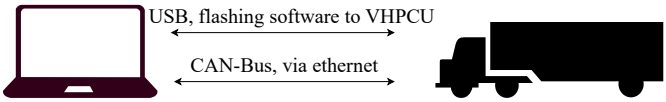


Figure 7.2: *Test setup for the electric truck.*

The USB connection to the vehicle was to flash the software to the vehicle from the laptop. The CAN bus connection was through Ethernet for the purpose of two way communication between the laptop and the vehicle. This communication network allows for the collection of data during the tests.
When the truck was switched on, the decoupling valve was in the decoupled state initially. Therefore the first test was carried out in the decoupled state. Further details related to the testing methodology and subsequent results are listed in the following sections.

7.1.1 Test 1: Decoupled Valve initially in Decoupled State

Figure 7.3 shows the results of this test for the duration of the test. This test was focused on checking if the diagnostic function could detect the occurrence of a decoupled fault. This means that the function should trigger a fault only when the temperature difference between the DCP_Clt and MDS_Clt is above the threshold and the valve is in decoupled condition. A summary of the results is presented below:

1. The decoupling valve is in decoupled state. Therefore ctso_Decoupled is true throughout as the valve state never changes to coupled.
2. The temperature difference between the DCP_clt and MDS_Clt temperatures exceeds the threshold of 4 °C at 120 seconds and at 520 seconds. This sets ctso_DcpldFault condition to 1.
3. The ctso_evalOk flag is 1 initially as no fault occurs at the start. After each fault occurrence, it sets to 0, showing that the evaluation check has failed.
4. The ctso_diagCompl flag is set at 1 initially and the fault occurrences at around 120 and 520 seconds, reset it to zero. This means that once the fault occurs, the diagnosis is completed after a brief evaluation time delay.
5. The diagnostic test code plot shows the point in time when the VHPCU receives the message that a fault has occurred.

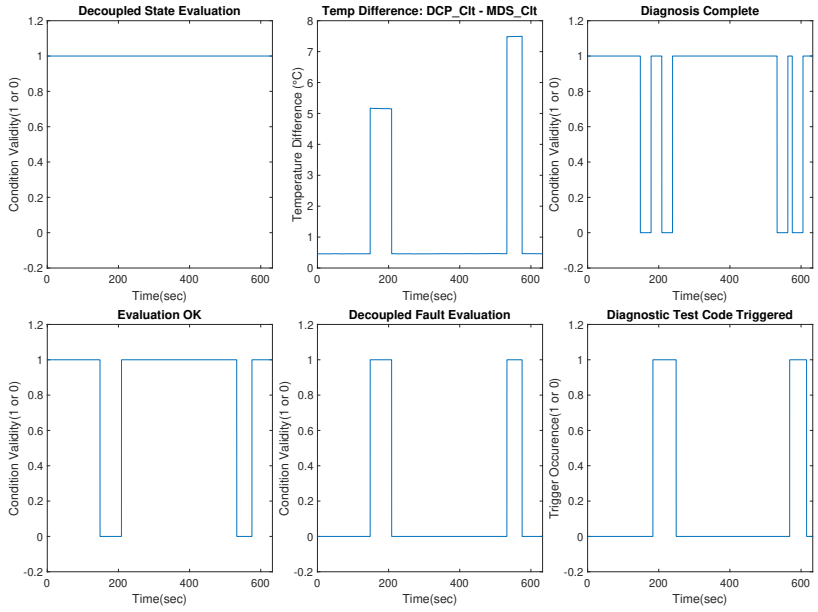


Figure 7.3: *Vehicle Test 1.*

7.1.2 Test 2: Decoupled Valve initially in Coupled State

This test focused on the detection of a coupled fault by the diagnostic fault. It also tested if the function could handle a sudden transition from the coupled to decoupled state. This was to test if the diagnostic function is able to differentiate between these states and not trigger a fault unnecessarily. For example if the temperature difference between ESS_Clt and DCP_Clt is above the threshold and valve is in coupled state initially, the fault should be triggered. But if during this time period, there is a sudden transition of the valve to the decoupled state, the fault alarm should be deactivated. Figure 7.4 shows the results of this test. A summary is presented below:

1. The ctso_Coupled and ctso_Decoupled states are during the test as can be seen in their respective plots.

2. The temperature difference between DCP_Clt and ESS_Clt is consistently above the fault threshold of 4 °C except for the brief period from 215 to 300 seconds. In spite of this, the ctso_CpldFault sets to 0 when the valve state is changed from coupled to decoupled. Additionally, ctso_evalOk sets to 0 for the time period of the valve being in decoupled state.

3. The function is able to distinguish between the 2 states and only trigger faults that are appropriate for that state.

4. The ctso_CpldFault corresponding to an excessive temperature difference between DCP_Clt and ESS_Clt is declared only when the decoupling valve is in coupled state.

5. All other conditions work as described in the earlier test. They are able to detect and report the fault properly. The transitions from coupled to decoupled state and vice versa also do not cause issues with fault detection. This proves the robustness of our control logic.

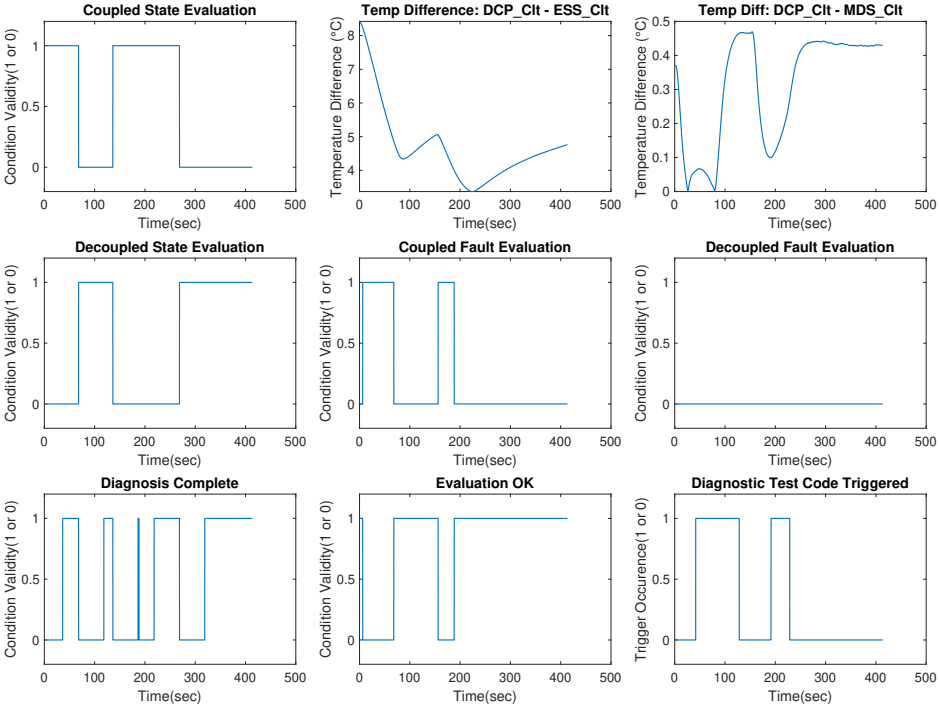


Figure 7.4: *Vehicle Test 2.*

7.2 Test Results for Vehicle In Driving Mode

Due to time and availability constraints with the vehicle, the tests could not be carried out for the vehicle in driving mode. Volvo GTT had earlier logged data for the vehicle in driving mode (only for no fault case). The logged data was fed into the diagnostic function to test that the logic worked as intended and no fault was reported.

7.2.1 Simulation and Vehicle Result Comparison

The drive cycle based on logged data recorded by Volvo GTT is presented in Figure 7.5. The vehicle was operating in driving mode and the ambient temperature was 25 °C.

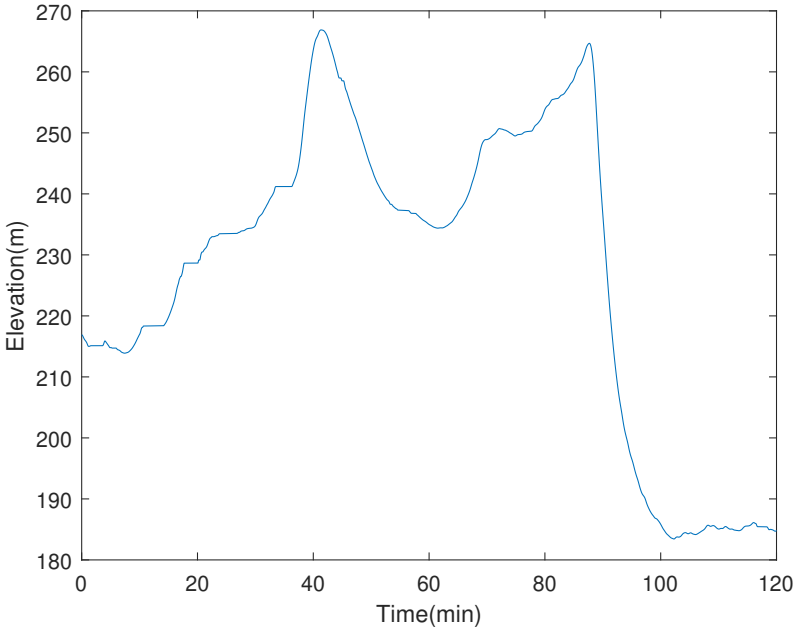


Figure 7.5: *Elevation change during the drive cycle*

Figure 7.6 shows the plots for temperature differences for the actual vehicle. The decoupling valve is open and hence the system is in decoupled state. The temperature difference between the DCP_ Clt and MDS_Clt is negligible while the difference between DCP_Clt and ESS_Clt is relatively high and varies throughout the drive cycle.
Figure 7.7 shows the coolant temperature differences and decoupling valve position when the same drive cycle is simulated in GSP. As seen from the figures, the simulation shows similar results regarding temperature difference calculations and decoupling valve states.

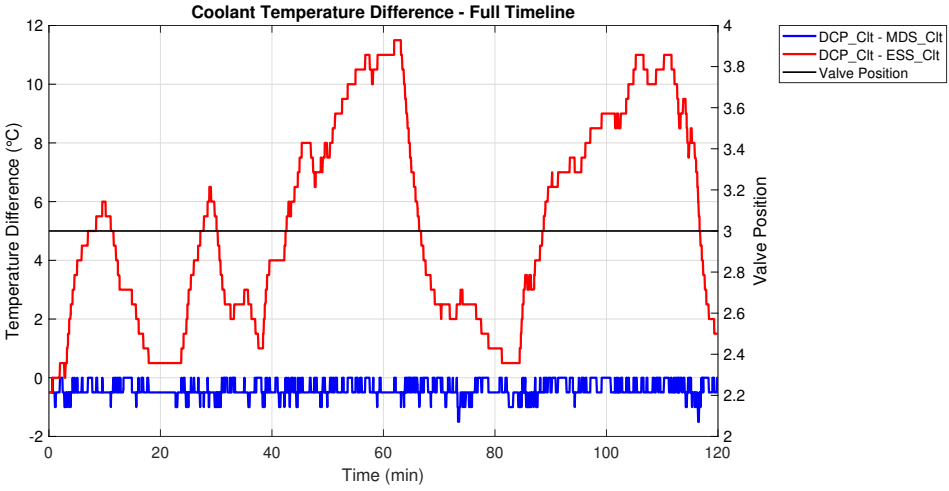


Figure 7.6: *Coolant temperature difference for vehicle in driving mode at an ambient temperature of 25 °C. The decoupling valve is closed.*

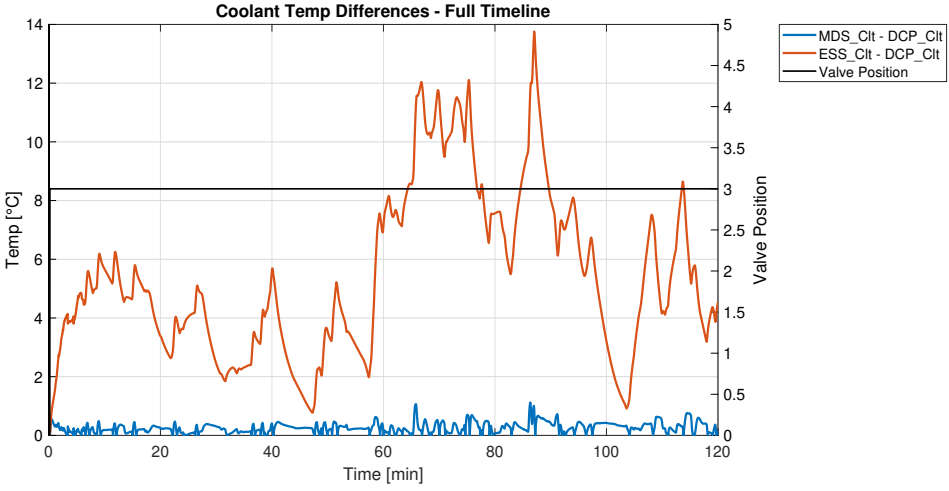


Figure 7.7: *Coolant temperature difference for simulation at an ambient temperature of 25 °C. The decoupling valve is closed.*

7.2.2 Test Results

The diagnostic function is tested again using the logged data of temperature values for DCP_Clt, MDS_Clt, ESS_Clt sensors and the decoupling valve position. This data is fed to the function and the results of this test are shown in Figure 7.8. Since the temperature difference between the DCP_Clt and MDS_Clt sensor is almost negligible (varies between 0 to -1.5 °C) and the decoupling valve is open (decoupled state), a fault should not be set in the function. The Figure 7.8 shows this with the ctso_Decoupled signal and ctso_evalOk signals being set throughout the test. The temperature difference between DCP_Clt and ESS_Clt sensor is high but this does not trigger a fault as the valve is in decoupled state and not in coupled state. Hence this test proves that the function works as it should and does not trigger any unnecessary fault when the vehicle is in driving mode.

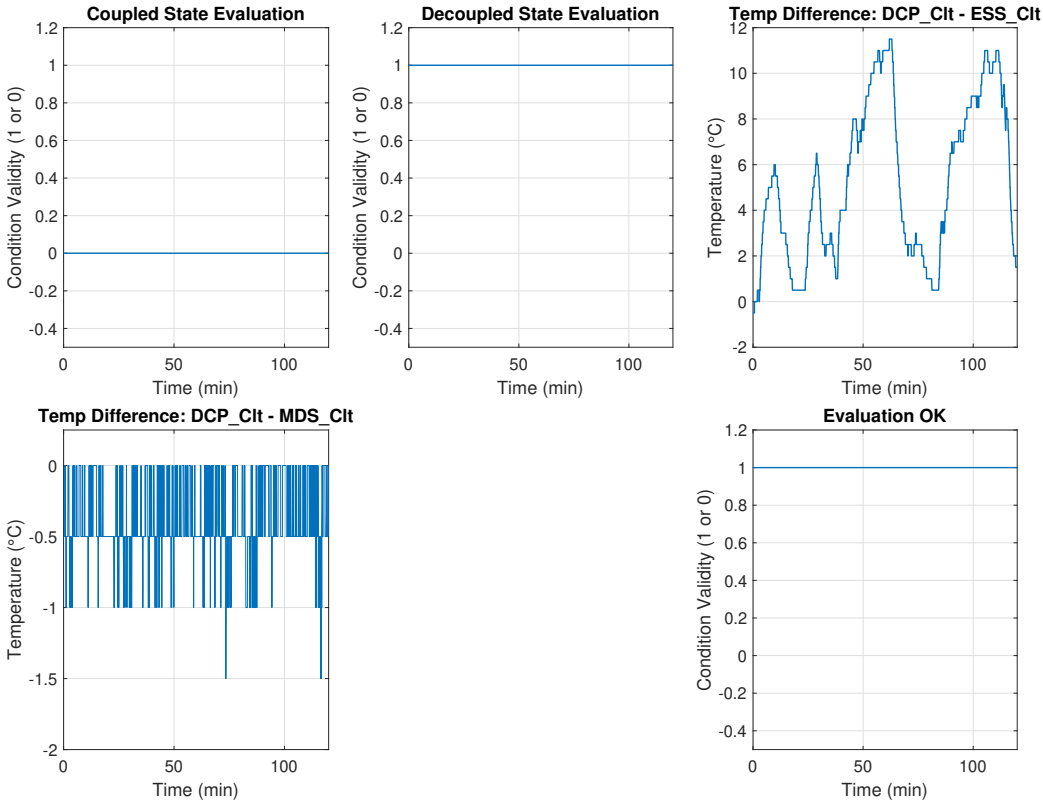


Figure 7.8: *Diagnostic function results for logged data, when vehicle is in driving mode (no fault case).*

8

Methodology: Temperature Estimation

The current diagnostic and control logic is based on different temperature set-points that are calculated by comparing different temperature sensor readings across the thermal system. The temperature sensors are only accessible via CAN bus for a limited number of locations in the system. This limits the ability of the VHPCU to monitor temperature changes throughout the system.

A place where a VHPCU accessible temperature sensor does not currently exist but would be beneficial to have, is after the ESS chiller. This would enable the system to check for any issues with the temperature of the coolant before it enters the batteries. It would allow the system to identify if there are any faults with the operation of the chiller or the heater. To test this hypothesis, a model to estimate this temperature after the ESS chiller is described in this section.

8.1 Introduction to The Estimation Model

The ESS loop consists of the chiller pump, heater and chiller. The loop has one temperature sensor that is read by the VHPCU, which is the ESS_Clt sensor.
Depending on what type of thermal mode is being used, either the chiller or the heater will be operated. In that case the temperature at the output of the loop will either be higher than or lower than the ESS_Clt temperature. Therefore, the estimation model will need to include a model of temperature changes due to the operation of the chiller and heater. The temperature change due to heat transfer in the model is created by referring the theory defined in Section 3.4. Additionally information about the mass flow of coolant in the ESS loop is taken from the actual data of the battery and chiller pumps. This data is taken from Volvo GTT's flow simulation results. On the basis of these inputs, the estimation model has been built and tested under various conditions to validate its efficacy

and accuracy.

8.2 Final Estimation Model

The estimation model is presented in Figure 8.1. It has a total of 6 input and 5 output signals. It has 2 sub blocks, the mass flow calculation block and the temperature estimation block.

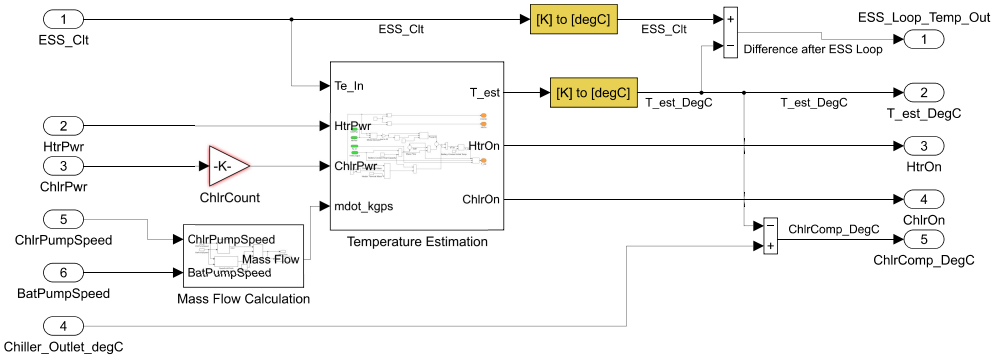


Figure 8.1: *Estimation model for ESS loop.*

The inputs and outputs of the model are described as follows:

1. ESS_Clt: Temperature value before the ESS loop, as described in Section 2.1.
2. HtrPwr: Power consumed by heater in kW.
3. ChlrPwr: Power consumed by chiller in kW.
4. ChlrPumpSpeed: Speed of the chiller pump in rotations per minute (RPM).
5. BatPumpSpeed: Speed of the battery in rotations per minute (RPM).
6. Chiller_outlet_degC: Temperature of the coolant after leaving the ESS loop, obtained from the GSP simulation.
7. T_est_DegC: Estimated temperature value of the coolant after leaving ESS loop.
8. ESS_Loop_Temp_Out: Temperature difference between ESS_Clt and T_est_DegC.
9. HtrOn: Flag indicating if the heater is switched on during the running of the estimator.
10. ChlrOn: Flag indicating if the chiller is switched on during the running of the estimator.
11. ChlrComp_DegC: Temperature difference between the Chiller_outlet_degC input value and the T_est_DegC estimated value.

8.2.1 Mass Flow Calculation

This block is shown Figure 8.2. The design logic for it is explained as follows:

1. The mass flow calculation block takes in the ChlrPumpSpeed and BatPumpSpeed and outputs the corresponding mass flow value as presented in Figure 8.2. This is obtained by using a lookup table which has designated mass flow values for a specific combination of the chiller pump and battery pump speed.

2. The speeds are first checked to see if they are within the bounds of the operating regions of the pump. This is done in the SpeedLimitCheck block, as presented in Figure 8.3, which outputs the SpeedWithinBounds boolean flag. This flag is 1 when the speed is within the limits and 0 otherwise.

3. The output is decided with a switch which outputs the mass flow from the lookup table if the speed is within bounds and no mass flow if the speed is out of bounds. In case the speed is out of bounds, it means that the pumps are either not operating properly or not at all. For this case the mass flow is assumed to be zero.

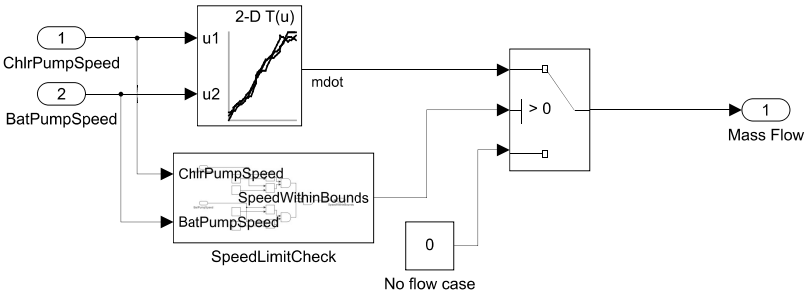


Figure 8.2: *Mass flow calculation.*

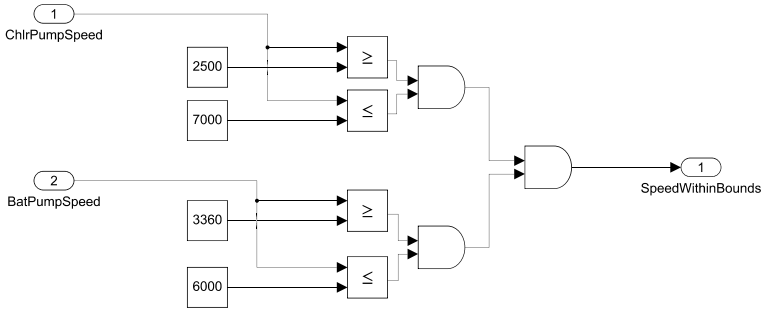


Figure 8.3: *Pump speed check.*

8.2.2 Temperature Estimate Calculation

Figure 8.4 presents the estimation model which has T_est, HtrOn and ChlrOn signals as the outputs.

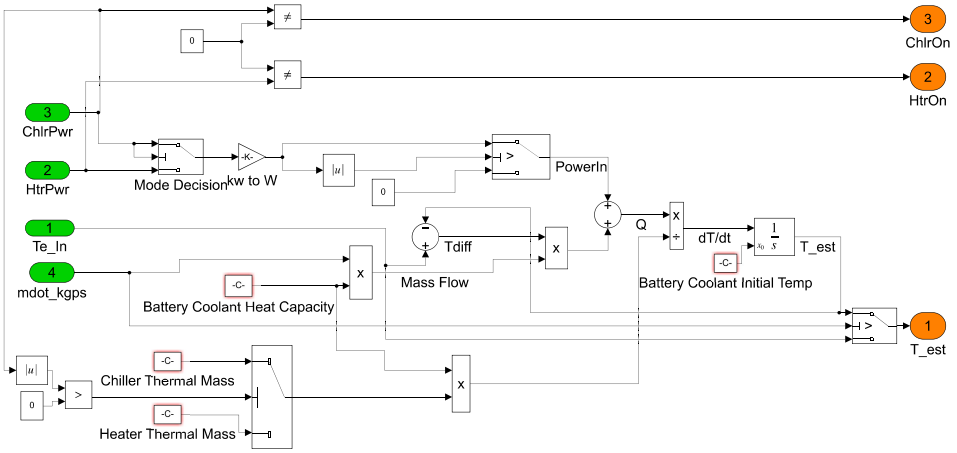


Figure 8.4: *Main estimation model.*

The calculation of the temperature estimate can be done by combining Equation 3.5 and Equation 3.8. This leads to the Equation 8.1

$$\frac{dT_{out}}{dt} = \frac{\dot{m}c_p(T_{out} - T_{in}) + Heat}{mc_p} \tag{8.1}$$

where the T_{out} is the temperature estimate to be calculated, T_{in} is the initial temperature, $\dot{m}$ is the mass flow of the coolant, c_p is the specific heat capacity of the coolant, $Heat$ is the power consumption of a component and m is the thermal mass of the same component. To calculate T_est, Equation 8.1 can be rearranged

$$T\_est = \int \frac{\dot{m}c_p(T\_est - Te\_in) + Heat}{mc_p} \tag{8.2}$$

where Te_in is the same as ESS_Clt and, $(T\_est - Te\_in)$ is Tdiff and $Heat$ is PowerIn from Figure 8.4.
The model is described as follows:

1. The inputs ChlrPwr and HtrPwr are used to decide which mode the ESS loop is operating in. This means that if the ChlrPower is non zero, then that is taken as the $Heat$ in Equation 8.1. Otherwise the heater power is taken as the heat if the HtrPwr is non zero.

2. ChlrPwr and HtrPwr are also used to decide the ChlrOn and HtrOn boolean flags. If the ChlrPwr is non zero, the ChlrOn is 1 and if HtrPwr is non zero, HtrOn is 1. In normal operation only one of the chiller and heater will operate within the ESS loop.

3. If the ChlrPwr is non zero, the thermal mass of the chiller is used as m in Equation 8.1. Otherwise the heater thermal mass is used.
4. The Battery Coolant Initial Temp constant corresponds to the initial temperature of the ESS coolant to the model, which is 298.15 $Kelvin$. This corresponds to Te_in.
5. The constant Battery Coolant Heat Capacity corresponds to the specific heat capacity of the ESS coolant, with a value of 3460 $J/kg*K$. This value is used as c_p.
6. After selection of the values as mentioned above, the model first checks if the mass flow input is greater than zero. If this condition is true, it calculates T_est according to Equation 8.2. If the condition is false the Te_{in} is directly given to T_{est}. This is because a zero value of mass flow means the chiller pump does not operate at all and a temperature estimation is not needed for this case as the ESS loop is inactive.

8.3 Estimation Model Result

After creating the model, multiple tests were run on the estimation model. This was done to check the efficacy of the model and validate its results. The simulated data from GSP simulation at different battery and ambient temperatures were taken to test the operation of the model. The different ambient and battery temperatures used were:

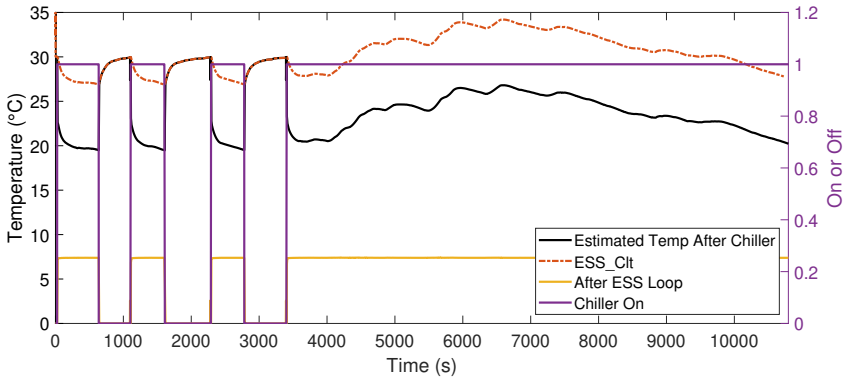
- Ambient temperature of 35 °C and initial battery temperature of 30 °C for the chiller operation in Test 1 and Test 2.
- Ambient temperature of -15 °C and initial battery temperature of 11 °C for the heater operation in Test 3 and Test 4.

8.3.1 Test 1: Estimator Performance with Chiller

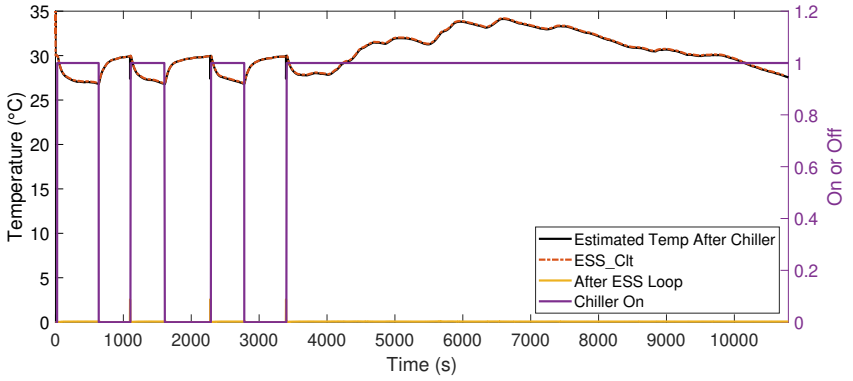
When the chiller operates in the ESS loop, the temperature after the chiller, will be lower than the ESS_Clt temperature. The estimator should reflect this change in temperature, if it works properly. To test this the estimator was provided with the required inputs and the temperature before and after the chiller were plotted. The results are presented in Figure 8.5a. The chiller turns on and off multiple times during the duration of the test. The signal 'After ESS Loop' shows the temperature difference between ESS_Clt and estimated temperature after the chiller. When the chiller is on, 'After ESS Loop' has a value of 7°C and when it is off, it is zero. Therefore, the estimator is able to account for the effect of the chiller. The chiller in this case operating with its full cooling capacity.

Figure 8.5b shows the operation of the estimator when the chiller power supplied is degraded. In this case, the cooling capacity of the chiller is limited to 1 % (assuming the chiller stops working) of its full capacity. The effect of this is that the

temperature before and after the chiller is the same. This is because the chiller power consumption term in Equation 8.2, becomes almost zero. The estimation model is able to show the effect of a fault on the system, which makes it potentially useful for fault detection.



(a)



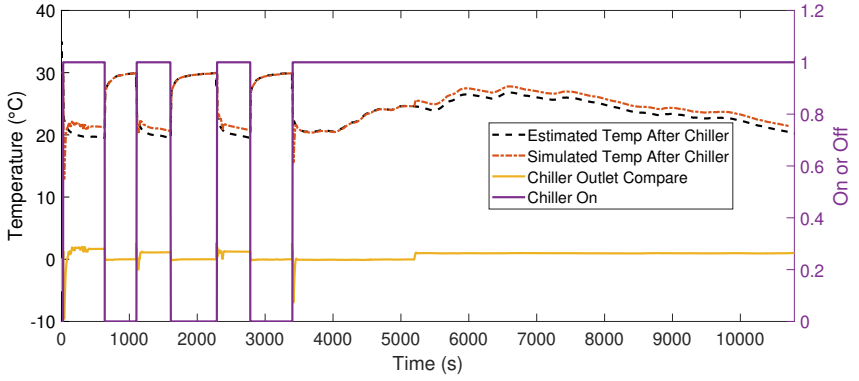
(b)

Figure 8.5: *Performance of the estimator. (a) No fault. (b) Chiller effect degraded (1% of normal power).*

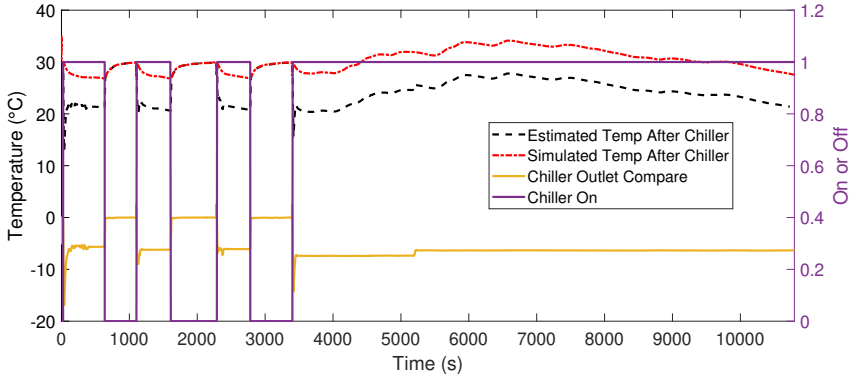
8.3.2 Test 2: Estimator and Simulation Comparison for Chiller

As mentioned in Section 8.1, the temperature after the chiller is not accessible by the VHPCU and hence it cannot be used to validate the temperature estimator. The GSP simulation has the value of this temperature and this value can be used to compare with the value obtained by the estimator. This test was performed to compare the estimator performance for the chiller in a no fault and degraded chiller case.

Figure 8.6a presents the results when there is no fault in the chiller and it operates with the full cooling capacity. The estimated temperature value closely follows the simulated temperature value with slight differences. This difference is calculated as the signal 'Chiller Outlet Compare', which has a maximum value 3°C and average of around 1°C. This shows that the estimator is quite accurate. The lack of actual data from the physical temperature sensor leads to this being the only way of validating the estimator.
Figure 8.6b illustrates a case where the chiller has been degraded to 1 % of its cooling capacity. This degraded power of the chiller is applied in the simulation environment . The estimator has the chiller operating at full power. The effect of the fault can be seen when the simulated temperature rises to a higher value as compared to the estimator. Therefore, the degradation of the chiller in the simulated environment can be detected by comparing it against the estimated temperature value.



(a)



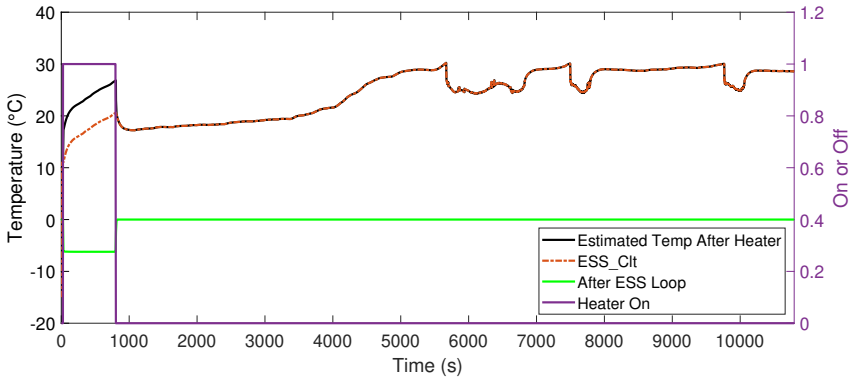
(b)

Figure 8.6: *Estimator versus simulated temperature comparison. (a) No fault. (b) Chiller effect degraded (1% of normal power).*

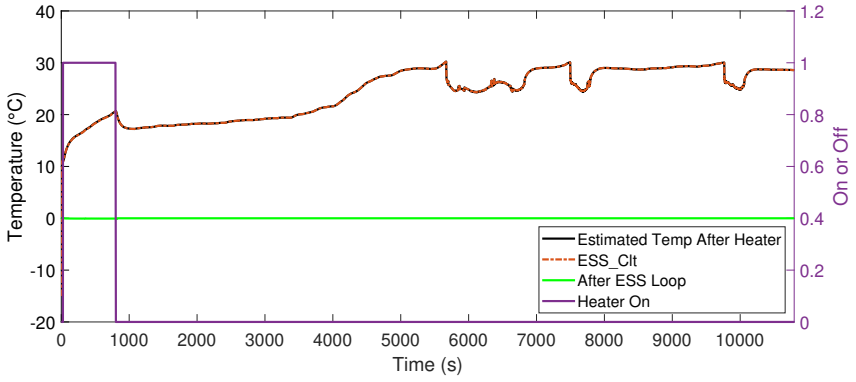
8.3.3 Test 3: Estimator Performance with Heater

The estimator is also setup to estimate the temperature after operation of the heater. When the heater is active the estimated coolant temperature after chiller should increase compared to the ESS_Clt. This is tested and the results are as seen in Figure 8.7a. The heater heats up the coolant and the temperature after the heater rises. The 'After ESS Loop' signal in this case, shows the temperature difference between the ESS_Clt temperature and after heater temperature. It is negative as the temperature after the heater is subtracted from the ESS_Clt temperature. The difference is about 6°C when the heater is turned on.

For the duration when the heater is off, there is no difference between the two temperature values. When the heater power is degraded (set to 1% of its original power) the difference between the estimated coolant temperature after chiller and ESS_Clt is negligible. This effect is illustrated in Figure 8.7b.



(a)



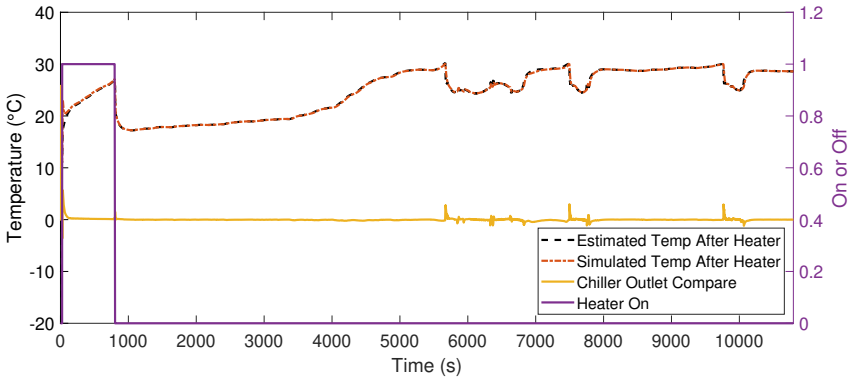
(b)

Figure 8.7: *Estimated temperature after chiller compared to ESS_Clt. (a) No fault. (b) ESS heater effect degraded (1% of normal power).*

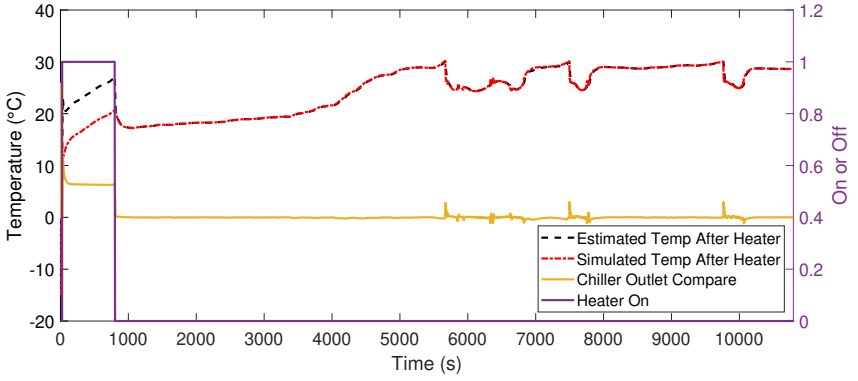
8.3.4 Test 4: Estimator and Simulation Comparison for Heater

The estimator was again tested for the heater operation against the simulated temperature value after the heater. For the no fault case, Figure 8.8a, shows that the estimator accurately estimates the temperature after the heater in comparison to the simulated value.

When the heater is degraded to 1 % of its actual heating capacity in the simulated environment, and compared against the estimated temperature, there is a difference between the temperatures. This difference can be seen in Figure 8.8b, when the heater is on and has a value of around 7°C. After the heater is switched off, the simulated temperature becomes equal to the estimated temperature value.



(a)



(b)

Figure 8.8: *Comparison of temperature estimation and with simulation data. (a) No fault. (b) ESS Heater effect degraded (1% of normal power).*

9

Conclusion and Discussion

The thesis work has focused on answering various research questions set forth earlier:

1. **Potential faults and failures in the thermal system:**

   A detailed investigation of the already available FMEA ensure that different types of failures of thermal system components were catalogued and studied. The components investigated included pumps, heaters, chillers as well as temperature sensors. The complete list of components investigated is presented in Table 4.1 to Table 4.3. This list of failures were used to simulate different types of faults in the GSP simulation system.

   Some failure modes could not be simulated in GSP due to limitations of the simulation setup, for more details refer to Section 4.2. These included leakage in pipes, out of sync issues with decouple valves and unavailability of the CAB Heater in the simulation system.

2. **Effect of injection of faults on the thermal system**

   The fault injection that was carried out in the GSP simulation led to different results depending on the type of fault injected. The faults that led to the most severe system effects were low flow from MDS pump, offset in the DCP_Clt sensor and reduced radiator heat rejection capacity.

   These effects are discussed in detail in Section 4.3. An example of one effect was low flow in the MDS loop causing problems with overheating of the electric motor windings. This proved that the simulation setup could be used as a reasonable way to measure the effects of these faults in the actual physical system.

3. **System level diagnostic function based on effects of fault injection**

   The effects seen from the fault injection seemed to form a valid basis for the diagnostic function design. The only way to validate this was to build and test a function on the basis of insights from the fault injection results.

   As such a particular fault, DCP_Clt sensor offset, was chosen to confirm the hypothesis. This fault was selected as temperature measurements decide the control operation for the entire thermal system and hence the temperature sensors need to be accurate. Additionally any other fault in the thermal system affected the temperature directly and if the temperature sensors indicated an offset. There would be no way to differentiate between the various faults.

   The simulated fault injection showed the effect to be mainly dependent on the operation of the decoupling valve. As described in Section 5.1, the temperature differences between DCP_Clt and MDS_Clt for the decoupled state and DCP_Clt and ESS_Clt for the coupled state were selected to form the basis of the diagnostic function.

4. **Performance of the diagnostic function on the VHPCU**

   The function was tested on the VHPCU test bench to validate its efficacy. The testing method and results obtained are presented in chapter 6.

   One insight from these tests was that the function initially struggled to recognize the change in valve state as a valid reason to reset a fault status. For example, if a temperature offset fault had occurred when the valve was in the decoupled state, the function continued to show this fault even when the valve changed its state to coupled. This was then corrected by adding an additional condition in form of ctso_CpldAlarm and ctso_DcpldAlarm. This addition was tested with the two tests presented in chapter 6, test number item 6 and item 7.

   The success of these tests proved that the results from the fault injection in the GSP simulation system can be used to create a diagnostic function that works in the VHPCU diagnostic system. The test proved that this function can detect the temperature offset fault when implemented on the VHPCU, triggering only when all fault conditions were met completely.

5. **Vehicle Tests**

   The final stage of the diagnostic function was to test it in an electric truck. This was done on a truck in the Volvo GTT testing area as described in Section 7.1.

   The diagnostic function was tested in both states of the decoupling valve as

described in Subsection 7.1.1 and Subsection 7.1.2. This was done to test all possible combinations of fault occurrence in the vehicle. The tests were successful and the temperature sensor offset faults were detected by the function. Additionally the function was able to clearly differentiate between the coupled and decoupled state, preventing an unnecessary declaration of fault occurrence.

One point of interest was that at standstill, the ESS_Clt and DCP_Clt temperatures differed by a value far larger than initially anticipated. Even when the decoupling valve was in the coupled state, the temperature difference between these two sensors was quite high and it took some time for it to settle down to a lower value. This meant that the fault in the coupled state was set almost immediately due to such a high temperature difference. This can be dealt with by changing the temperature offset for the coupled state or by increasing the time delay to set the fault in coupled state. This was not done during the testing due to limited availability of the vehicle.

The final test was carried by using the logged data of an vehicle as mentioned in Section 7.2. This test showed that since the logged data did not have a fault on the temperature sensors, the function did not trigger a fault alarm. Data with a temperature sensor fault case on a vehicle in driving mode was unavailable and hence this particular aspect of the function test could not be tested. From the comparison of the GSP simulation and actual data it can be seen that the simulation can be used to create a diagnostic function. The simulation had almost exactly same results for the coolant temperature difference calculations as calculated from the logged vehicle data.

6. **Addition of temperature estimation to improve data availability in VHCPU**

   As discussed in chapter 8, the unavailability of temperature sensor after chiller on the CAN bus is an issue for the current diagnostic and control system. To solve this problem, a detailed temperature estimation system was built by modelling the ESS loop. The model includes real data from the pumps used in the system as well as real values for the coolant parameters. This makes the estimation a close approximation of the vehicle system.

   Good model accuracy is indicated by results presented in Section 8.3 where the temperature estimation almost equals to the temperature value obtained by the GSP simulation. Additionally the tests showed that for a fault present in the simulated environment, a comparison with the estimated temperature values enables detection of the fault for the chiller and the heater. Thus, the estimator could be used to diagnose faults in these components if developed further. It serves as a feasible improvement opportunity if tailored to the existing diagnostic and control system.

10

Future Work

This study has covered important aspects of developing thermal system diagnostics for electric trucks, but there are still several areas worth exploring further. Building on the foundation laid here, future work can help overcome current limitations and expand the system's capabilities. Some key areas for future development are outlined below:

1. Additional investigations regarding different faults (other than the ones investigated in this report) injected into the system. There could be more faults that have adverse effects on various components of the system that have not been investigated here. Similar diagnostic functions could be built to handle them.

2. The faults left out of the investigation due to limitations of the simulation system, like pipe leakage, could be investigated directly on the physical components. These investigations could allow for more detailed diagnostic functions that take into account the system as a whole.

3. In the current investigation, for the the fault with the MDS pump flow, a very severe fault occurrence was considered. In the future, a less severe fault such as speed limitation by only 50 % instead of 1 % can also be studied to understand effects of different speed limitations on the pump flow.

4. Due to lack of multiple sensors such as flow and pressure, the diagnostic function focused solely on the logic based approach. If sensors are added to the vehicle at a later stage, an equation based diagnostic approach might be possible. This could be done by modelling the various components in the thermal system mathematically with flow and pressure equations.

5. Another approach for the diagnostic function could be a data based approach. This would involve accessing logged temperature sensor data from

vehicle in driving mode to gauge system behaviour in different conditions and accordingly build a mathematical model of the system. One such approach could be Kalman filter based to create a correlation between temperature sensors and the coolant flow through the system.

6. The diagnostic function can be improved by adding a logic to take care of the MDS low flow fault case. Currently the diagnostic function does not differentiate between these two faults but in the future a logic based on electric winding temperature changes can be added.

7. The estimation function can be modified and changed to be used directly in the diagnostic system. The estimation will have to be simplified to consume less memory from the VHPCU without sacrificing accuracy.

Bibliography

[1] Mattias Nyberg and Erik Frisk. *Model Based Diagnosis of Technical Processes*. Department of Electrical Engineering (ISY), Linköping University, 2020.

[2] Bérangère Suiphon, Zineb Simeu-Abazi, and Eric Gascard. "Implementation of a fault diagnosis method for timed discrete-event systems". In: *Proceedings of 2013 International Conference on Industrial Engineering and Systems Management (IESM)*. Proceedings of 2013 International Conference on Industrial Engineering and Systems Management (IESM). 2013. DOI: 10.1109/IESM.2013.6763142.

[3] Ali Behravan et al. "Fault injection framework for fault diagnosis based on machine learning in heating and demand-controlled ventilation systems". In: *2017 IEEE 4th International Conference on Knowledge-Based Engineering and Innovation (KBEI)*. 2017, pp. 0273–0279. DOI: 10.1109/KBEI.2017.8324986.

[4] Steli Loznen, Constantin Bolintineanu, and Jan Swart. *Electrical Product Compliance and Safety Engineering*. Artech House, 2017. ISBN: 978-1-63081-011-5.

[5] Robin E. McDermott, Raymond J. Mikulak, and Michael R. Beauregard. *The basics of FMEA*. 2nd ed. New York: CRC Press/Productivity Press, 2009. ISBN: 9781563273773.

[6] Mogens Blanke et al. *Diagnosis and Fault-Tolerant Control*. Springer Berlin Heidelberg, 2016. ISBN: 978-3-662-47942-1. DOI: 10.1007/978-3-662-47943-8.

[7] Steven X. Ding. *Model-based fault diagnosis techniques: design schemes, algorithms, and tools*. Springer, 2008. ISBN: 9783540763031.

[8] Zhiwei Gao, Carlo Cecati, and Steven X. Ding. "A survey of fault diagnosis and fault-tolerant techniques—part i: Fault diagnosis with model-based and signal-based approaches". In: *IEEE Transactions on Industrial Electronics* 62 (2015). ISSN: 1557-9948. DOI: 10.1109/TIE.2015.2417501.

[9] Theodore L. Bergman et al. *Introduction to Heat Transfer*. 6th ed. Hoboken, NJ: John Wiley & Sons, 2011. ISBN: 978-0-470-50196-2.

[10] John H. Lienhard. *A heat transfer textbook*. Fifth Edition. Mineola, New York: Dover Publications, 2019. 771 pp. ISBN: 9780486837352.

[11] H. Sun, S. Yuan, and Y. Luo. "Characterization of cavitation and seal damage during pump operation by vibration and motor current signal spectra". In: *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy* (2018). DOI: 10.1177/0957650918769761.

[12] C. Stephen. "A review of cavitation effects on vibration in centrifugal pump". In: *Fluid Mechanics Research International Journal* (2018). DOI: 10.15406/fmrij.2018.02.00028.

[13] E. Holmberg. *Industrial Temperature Measurement: Techniques and Applications*. Wiley-Blackwell, 2007. ISBN: 978-0470511407.

[14] S. P. Panchangam and V. N. A. Naikan. "Reliability analysis of temperature sensor system". In: *International Journal of Reliability and Quality Assurance* 20.2 (2013). DOI: 10.1142/S0218539313500034.

[15] Thi Huynh. *Fundamentals of Thermal Sensors*. Springer, 2015. ISBN: 9781493925803.

[16] W. Sun, Y. Zhang, and Z. Wang. "Temperature Sensor Calibration and Compensation Methods for Precision Applications". In: *Sensors* 16.4 (2016), p. 485. DOI: 10.3390/s16040485.