ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

SCHOOL OF ENGINEERING AND ARCHITECTURE

DEPARTMENT of
ELECTRICAL, ELECTRONIC, AND INFORMATION ENGINEERING
"Guglielmo Marconi"
DEI

MASTER'S DEGREE in
ADVANCED AUTOMOTIVE ELECTRONIC ENGINEERING

MASTER THESIS
in
Automotive Applied Controls

Hardware-in-the-loop setup and test of an air spring control system

CANDIDATE

Leonardo Chiacchiararelli

SUPERVISOR

Professor Nicola Mimmo

HOST COMPANY SUPERVISOR

Roberto Rabbeni, Ph.D.

Academic Year
2020/2021

Session
I

Index

Figures and tables

1. Introduction

The production process of modern cars is becoming increasingly complex and more and more consisting in the production of the electronic components. To regulate the production process of these components manufacturers rely on the so called V-Model.

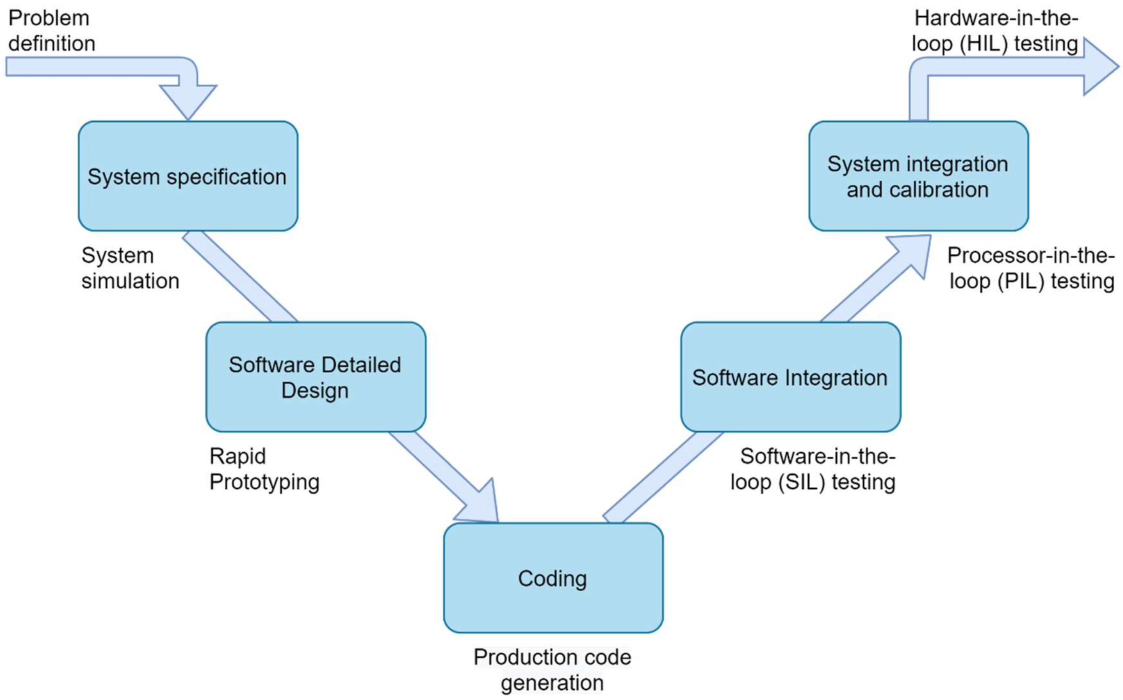


Figure 1.1 V-Model diagram.

Figure 1.1 shows the V-Model diagram; it is constituted by two main parts. The descendent part of the diagram regards the design phase, while the ascending part is about the verification and validation phase of the product.

In the ascending part, Software-in-the-loop (SIL), Processor-in-the-loop (PIL), Hardware-in-the-Loop (HIL) tests are performed on the simulation of the plant that the Electronic Control Units (ECUs) have to deal with to save time, to reduce costs, and harm risks.

HIL tests are the last step of the validation phase before the component can be integrated with the rest of the vehicle. HIL tests, based on their goal, are divided in four categories based on the selection of the component to be tested and the kind of simulator exploited for the simulation. The HW under test can be one ECU or a more complex system composed by one or more ECUs and all the required interface components. In the first case the tests are referred to as Component-HIL tests whereas in the latter they are referred to as System-HIL tests. The plant simulation can be implemented as a dynamic or static model called Dynamic-HIL and Behavioural-HIL respectively.

In Figure 1.2 a generic HIL test bench setup is shown. The test is structured as follows:

- The hardware under test is connected, thanks to its designed IO interfaces, to a Real-Time (RT) vehicle simulator. The RT capability is essential to test the timings of the ECU response
- The tests are performed on the closed loop system composed by the HW under test (HUT) and the RT simulator. The object collecting all the data useful to run the tests (e.g. Input patterns, expected outputs, …), is defined test case.
- The closed loop system outputs are checked. More in details, the behaviour of the controlled plant as well as the correct use of the I/O channels are verified.
- If the ECU does not pass the test, a bug-scan procedure is performed to solve the issue. Then the test is performed again

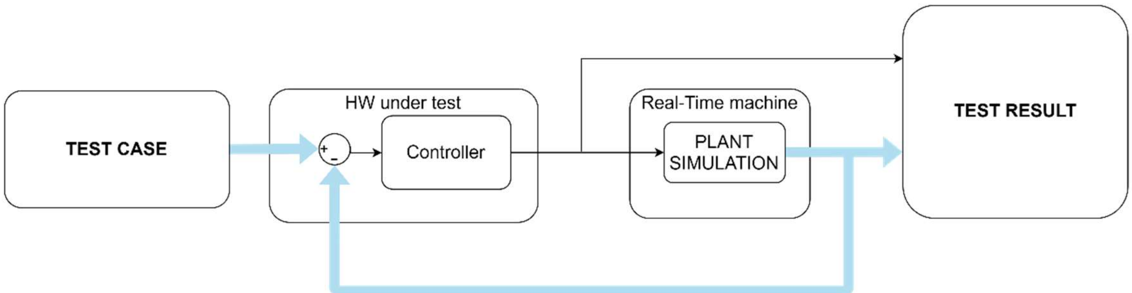


Figure 1.2 HIL tests setup.

This work focuses on the use of the HIL technique to test the ECU responsible for the air spring system control.

State of art

Some Maserati vehicles are equipped with an air spring system which is controlled by a dedicated ECU, called Air Spring Control Module (ASCM). This ECU was produced and tested by the air spring system vendor, so that, inside the Maserati labs, a HIL bench dedicated to this ECU was not needed. A HIL bench dedicated to a similar system was designed to validate the software of an ECU deployed in cars produced by Chrysler, which belongs to the same Maserati group. This purpose of this bench was to validate some specific logics in the software hence the plant mathematical model was unnecessary. The bench adopted for this purpose was a Component-Behavioural-HIL bench.

In the platform currently under development in Maserati, the air spring control function has been transferred from the ASCM to a new, cross-functional ECU called Vehicle Domain Control Module (VDCM). This ECU, built internally in Maserati, is responsible for the vehicle dynamics control systems, comprehending the air spring system.

Motivations

This work was motivated by the need to test, through a HIL scheme, the VDCM functionalities affecting the air spring control system (both HW and SW). To reach this goal three options have been considered. The first focused on tests like those proposed by Chrysler which were not suitable to verify the dynamics of the ECU, since the HIL bench was not designed accordingly. An alternative strategy was to build a HIL bench hosting the actual air spring system and the ECU, so to have a System-Behavioural-HIL bench. Unfortunately, also this strategy has been proved to be unfeasible. Indeed, due to safety reasons it was not possible to set up a bench hosting a pressurized air system inside Maserati labs. The last investigated option was constituted by the design and the implementation of a high-fidelity simulator of the air spring plant, as described in the following section.

Contributions

This work focuses on the set-up of a Component-Dynamic-HIL designed to test the air spring control function of the VDCM.

The first step focused on the identification of the mathematical model of the plant. The air spring system is composed by the springs and a pneumatic circuit used to regulate the mass of air flowing inside and outside the springs. Once known the circuit structure, the model has been defined in a two-step procedure: first, the list of all the input and output signals required to interface the ECU and the plant has been defined. Then, the equations that describe the dynamics of each component were selected among classical fluid dynamics and mechanics theories. The pneumatic elements of the circuit have been modelled using thermodynamic and fluid dynamic equations whereas the springs also required a mechanical description. The overall plant model was obtained as composition of all the equations describing each component of the system. To make the simulator as accurate as possible, the physical parameters of the plant (e.g. chamber volumes, nominal pressures, ...) have been obtained by field tests and through confidential company notes.

Once the set of equations characterizing the input-output behaviour of the system was fully determined, the implementation phase took place. The RT simulators used in Maserati are systems designed by dSpace. The hardware that runs the simulations and hosts the I/O connections to the HUT is constituted by SCALEXIO machines. To properly set-up the simulation environment, a PC is connected to the SCALEXIO machines. This PC runs a software suite, offered by dSpace to support the tester. The suite is mainly composed by three tools, used in this work: ConfigurationDesk is the tool that the tester uses to configure the I/O connections between SCALEXIO and the HUT, ControlDesk helps the tester to manually send signals to the HUT and monitor its responses, AutomationDesk is used to automate the test runs. The suite supports Simulink as the tool to be used to implement dynamical models. Each equation has been implemented using Simulink, then the equations have been grouped in blocks. Each block models a component of the pneumatic circuit. As a result, the implementation is modular, in order to make it reusable for future projects. All the blocks have been integrated to form one model, so that the input-output behaviour of the system could finally be simulated.

Each block has been qualitatively validated. The goal of this validation process was to verify that each block, provided with specifically designed input patterns, responded as expected, compared to the known qualitative behaviour of the corresponding pneumatic circuit component.

The whole simulation model has been quantitively validated, this was possible since data acquisitions regarding the whole air spring system were available (there were not measurements regarding the single components of the system). The simulation model has been provided with the same inputs as the real system, then its outputs have been compared with the data acquisition measurements. In this process some parameters of the model have been tuned, to obtain a simulation as close as possible to the actual system's behaviour.

Once the model was validated, the VDCM could finally be connected to the RT simulator. To make this connection possible, the needed wiring harness has been assembled. Due to safety reasons, before enabling the air spring control software unit, the VDCM expects that some external signals assume some specific values. These signals regard the vehicle state, that is required to be in non-extreme driving conditions. Before the testing phase, all the mentioned signals have been set to the required values.

The test phase started with the analysis of the VDCM requirements. These requirements are collected in a document, used as a basis to build the test cases. The document specified the required output responses of the VDCM with given input vectors and external conditions. Each test case has been defined starting from these input-output associations.

Once clearly defined, the test cases have been implemented in AutomationDesk, the tool used to run the tests. The tool has been used to automatically change the inputs and the parameters internal to the Simulink model, in order to test the VDCM behaviour in different conditions of the air spring system. The tool has been also used to evaluate the test results and collect them in an automatically generated test report.

Thesis organisation

In Chapter 2 the mathematical model is presented, together with its implementation and validation.

In Chapter 3 the test cases design is presented, together with their submission and result interpretation.

In the last chapter conclusions and future steps are discussed.

2. Dynamical model of the Air Supply Unit system

In this chapter the Maserati air suspension system will be presented, together with the dynamical model used to simulate its behaviour during the Hardware-In-the-Loop tests.

The goal of the system is to regulate and maintain, according to the driver's requests, the vehicle's ride height, meaning the height of the sprang body of the vehicle. It is defined sprang body the portion of the vehicle that sits on the suspensions. The ride height is function of the suspension length. The suspensions change their length thanks to the air mass moving inside or outside their variable volume chamber. The air moves inside a pneumatic circuit, composed by the suspensions, a reservoir, a compressor, the pipes that link them together and the valves that regulate the airflow.

All the components present in the system are linked by sealed junctions thanks to which the fluid is free to move from one component to another resulting in pressure, volume, and temperature variation.

The system under investigation has three operative modes: compressor mode, reservoir mode and boost mode.

In compressor mode the air is forced inside the spring chambers by the compressor.

In reservoir mode the air moves inside the suspension chambers from a higher-pressure chamber (i.e. the reservoir).

The boost mode is useful when the air pressure in the reservoir relies in a certain threshold: higher than the atmospheric pressure but lower than the suspension's nominal pressure. In a boost mode levelling the air is forced inside the springs by the compressor, but its intake manifold is connected to the reservoir instead of the external environment. This will let the compressed air be at a higher pressure than in the compressor only mode, resulting in a faster levelling.

The way in which the system is wanted to work most of the time is the reservoir mode, because it is noise-free and faster compared to the other two, since the air in the reservoir is more pressurized than the one coming from the compressor.

In Figure 2.1 the pneumatic circuit is represented. In the figure the valves used to control the flow are shown. The valves are bidirectional, on-off, and electrically controlled by the ECU. The valves are grouped in a block, constituted by a chamber that puts in connection all the valves. It embeds an electric connection between the ECU and the valves. It also hosts a pressure sensor.

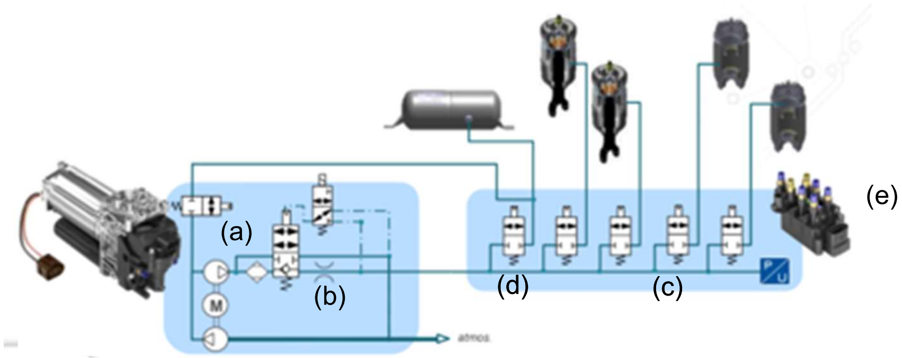


Figure 2.1 High level representation of the pneumatic circuit. (a) boost valve. (b) compressor/exhaust valves. (c) spring valves. (d) reservoir valve. (e) valve block.

2.1 Mathematical representation of the system

This section presents the mathematical model of the system described in Section 2.

The equations discussed rely on fluid dynamics, thermodynamics and mechanical dynamics.

The dynamical model is based on the representation of each block of the system separately, then the blocks are linked together through an interface quantity. In the model the interface quantity is the time varying air mass that moves from one component to another.

The outputs of the model are: the heights of each corner (i.e. the height of the sprang body of the vehicle above each wheel), and the pressure of the central valve block. The corner heights are used for feedback by the ECU. On the other hand, the pressure of the central valve block is exploited by the ECU to check plausibility, to avoid dangerous situations, and to identify the current lifting mode (boost, reservoir, compressor only).

The inputs of the model are the valve switch signals. These are ECU's outputs signals, they are used to regulate the airflow. An additional input signal is an extra vertical force, applied at the top of the spring. This input is used to model the weight variations of the vehicle's sprang body.

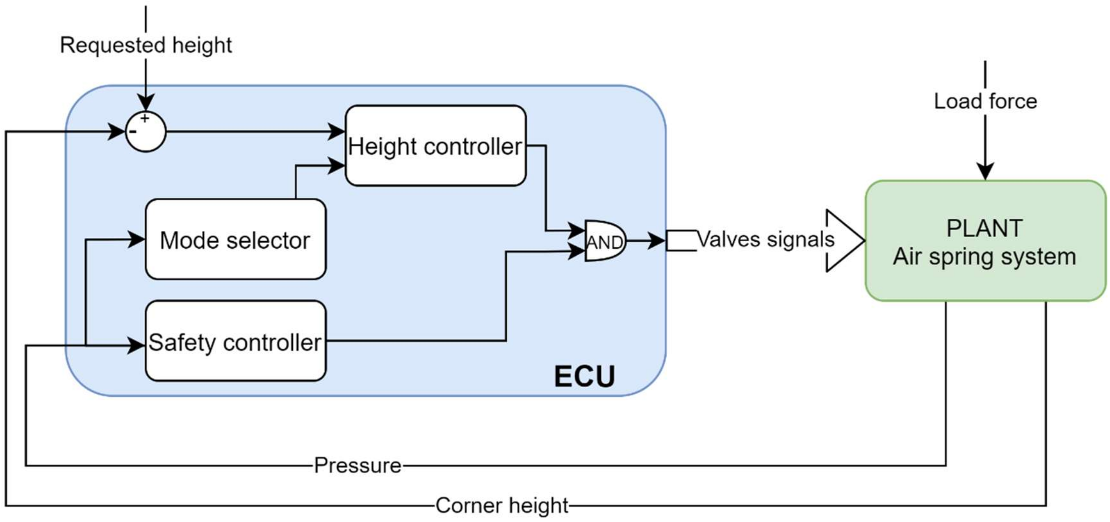


Figure 2.2 Controller-Plant-like scheme of the ECU-air spring system. The height controller sets the valve signals according to the feedback control of the ride height and the selected mode. If the safety controller judges not safe the situation, its output is a zero, so the valve signals will be all zeros, because of the AND gate.

2.1.1 **Valve**

All the valves of the system are on-off valves. They are modelled as two-phase systems. When they are closed their cross-section area $\mathrm{A}_{\mathrm{valve}}$ is equal to zero, so there is no airflow between the components that they link When they are opened $A_{valve} = S_{valve}^{nominal}$ and the air is free to flow.

The valve is modelled as a fixed diameter orifice that, depending on the pressure drop (Δ_p) insisting on them, will produce a different airflow. The airflow depends also on the upstream temperature (T_u). All the valves in the model are two-way valves.

The equation used is the following [1]:

$$\dot{m}_v = A_{valve} C_m C_q \frac{P_u}{\sqrt{T_u}} sign(\Delta_P) \tag{1}$$

$$C_m = \sqrt{\frac{2k}{R(k-1)} \left(\left(\frac{P_d}{P_u}\right)^{2/k} - \left(\frac{P_d}{P_u}\right)^{\frac{k+1}{k}} \right)^{1/2}} \tag{2}$$

With P_u being the upstream side pressure and P_d the downstream side pressure

C_q is a coefficient whose value is determined comparing the theoretical data from the equation dynamics and the actual behaviour of the valve.

The constant k is the *polytropic constant,* which in this case is $k = \frac{C_p}{C_v} = 1.4$. The constant assumes this value since the process is isentropic and the studied fluid is air; C_p and C_v are, respectively, the heat capacity at constant pressure and the heat capacity at constant volume.

Inputs of the valve blocks are: P_d, P_u, T_u, A_{valve}. This block's output is $\dot{m}_v$.

2.1.2 **Valve Block**

Most of the valves are grouped in one block (Figure 2.1 (e)). The block consists in a chamber, the valves have one side facing the chamber, the other facing the component they are linked to. The block also provides to the ECU the electrical interface to the valves. Inside the chamber there is a pressure sensor, whose output is read by the ECU.

The block is modelled as a fixed volume chamber, the following equations are used:

$$\dot{P}_b = \frac{kRT_b}{V_b} \dot{m}_b \tag{3}$$

Equation (3) is used according to [1]. P_b is the pressure inside the valve block chamber, V_b is the volume of the chamber and T_b the temperature of the fluid. R is the gas constant and $\dot{m}_b$ is the airflow entering or leaving the chamber.

$$\dot{m}_b = \dot{\mathrm{m}}_{\mathrm{in}} - \dot{\mathrm{m}}_{\mathrm{out}} \quad (4)$$

Where m_{in} is the incoming airflow and m_{out} is the exiting airflow.

This block's input is $\dot{m}_b$, its outputs is P_b.

2.1.3 **Reservoir**

This part of the system is modelled as a fixed volume chamber.

The equation used is, according to [1]:

$$\dot{P}_r = \frac{kRT_r}{V_r}\dot{m}_r \quad (5)$$

Where:

$$\dot{m}_r = \dot{m}_{in} - \dot{m}_{out} \quad (6)$$

The differential equation to evaluate the temperature is:

$$\dot{T}_r = \frac{RT_r^2}{PV}\left(\dot{m}_{in}(k-1) - \dot{m}_{out}(1-k)\right) \quad (7)$$

Equation (7) is derived from the mass-energy balance equation:

$$m_r C_v \dot{T}_r = \dot{m}_{in} C_p T_{in} - \dot{m}_{out} C_p T_r - \dot{m}_r C_v T_r \qquad (8)$$

Where:

$$m_r = \frac{P_r V_r}{R T_r} \qquad (9)$$

Where T_{in} is the temperature of the incoming air flow, supposed $T_{in} = T_r$

This block's input is $\dot{m}_r$, its outputs are P_r, T_r.

2.1.4 **Compressor**

The compressor must be described in its three different working modes: reservoir filling, boost levelling, compressor levelling.

The biggest difference will be between the second mode and the other two: this is the only case in which the compressor pressurizes a mass of air starting not from the nominal ambient pressure but a higher pressure.

To model this component the following equation has been used:

$$\dot{m}_c = \frac{W_c}{\frac{k}{k-1} R T_1 \left(\left(\frac{P_2}{P_1} \right)^{\frac{k-1}{k}} - 1 \right)} \qquad (10)$$

Where W_c is the power consumption of the electrical machine, it is a constant defined by the compressor vendor. The parameters are, referred to the generic-chamber-filling procedure: initial (P_1) and final (P_2) pressures, initial temperature (T_1). To set these parameters the basic situation has been addressed: in the reservoir filling phase the final pressure is the nominal reservoir pressure while P_1 is the atmospheric pressure, the temperature is the ambient temperature. In the boost levelling phase, the power consumption of the compressor is the same but, in this case, the starting pressure is the current reservoir pressure, lower than a threshold under which the boost function is activated.

To better understand the boost function and the behaviour of the cited equation, the plot of the $\dot{m}$ behaviour when varying the starting pressure is shown in, since it shows that to an higher starting pressure corresponds an higher $\dot{m}$.

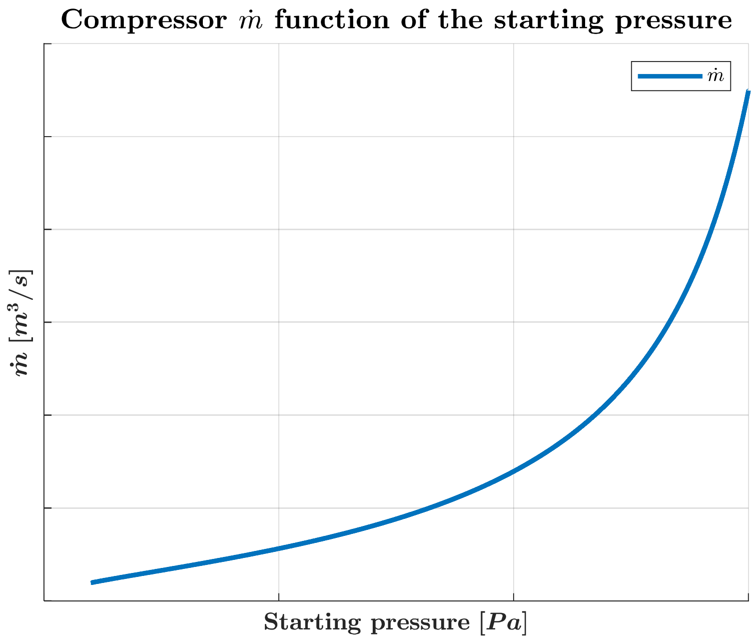


Figure 2.3 Airflow from the compressor, function of the starting pressure, according to eqn (10)

The picture highlights the advantage of the boost function, since it shows that to an higher starting pressure corresponds an higher $\dot{m}$.

During a compressor levelling the airflow the compressor is the same as in a reservoir filling phase, according to the vendor.

This block's input is P_1, that can assume the values P_r, P_{atm}. This block's output is $\dot{m}_c$

2.1.5 **Spring**

The spring is modelled as a variable volume chamber, the pressurized air inside the volume generates a force, that lifts the vehicle. The spring volume is evaluated as $V_s = A_{eff}(s) \times s$ where s is its length and A_{eff} is its base area, that varies in function of s. The function describing the s dependency of A_{eff} was found starting from experimental data given by the spring vendor.

The equations used to model the spring are the following [1] [2] [2]:

$$\begin{cases} \dot{P_s} = \dfrac{kRT_s}{V_s}\dot{m} - \dfrac{kP_s}{V_s}\dot{V_s} \\ \dot{V_s} = \dot{A}_{eff}(s)s + \dot{s}A_{eff}(s) \\ \ddot{s} = \dfrac{F_s}{m_{eq}} - g - \dfrac{\dot{s}\beta}{m_{eq}} + \dfrac{F_{extra}}{m_{eq}} \\ \dot{F_s} = \dot{A_{eff}}(s)P_s + \dot{P_s}A_{eff}(s) \end{cases} \tag{11}$$

Where P_s is the pressure inside the spring chamber, F_s is the force generated by the pressure inside the chamber, β is the damping coefficient of the rubber forming the chamber's walls, m_{eq} is the equivalent mass of the sprung body evaluated at the top of the spring as: $m_{eq} = m_s \times \lambda$. With λ being a coefficient used to take care of the kinematic of the suspension and m_s the vehicle sprang body mass. F_{extra} is an additional input to model an extra vertical force applied to the top of the chamber.

When a $\dot{m}$ variation occurs a certain mass of air leaves or enters the chamber. The mass variation results in a pressure variation, that results in a force variation, so that the force balance at the top of the spring leaves the equilibrium condition (Figure 2.4). When the equilibrium is lost the top of the chamber is free to move, varying the volume of the spring.

The force associated with the weight of the vehicle will be the same from one equilibrium condition to another, so will be the pressure.

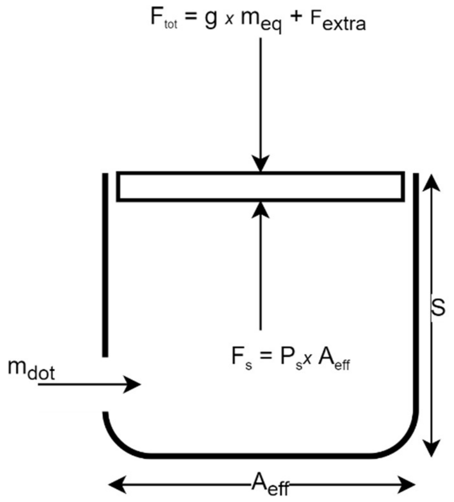


Figure 2.4 Force balance at the air spring top wall

The variation of the mass inside the spring will reflect on first approximation exclusively on a volume change (Figure 2.5.d) In second approximation the effect of the s dependency of A_{eff} has to be taken in account: since $F_s = A_{eff} \times P_s$, a variation of A_{eff} will correspond to a variation in the pressure, to make their product constant (Figure 2.5.c .e .f). Figure 2.5 shows, during a lowering manoeuvre, the evolution of the main quantities characterizing the spring dynamics according to (11). The exhaust valve and the spring valves are open together, in this way the air leaves the springs. Since the air mass inside the chamber is decreasing, the spring volume decreases and the car is lowered. The spike in Figure 2.5.a before the actual levelling phase, is due to the air flowing initially from the spring to the valve block, later also the exhaust valve is opened so that the levelling phase can take place. While both the valves are open the pressure initially decreases, since air is flowing outside the chamber, so does the force. Since the force is lower than the nominal value, the sprang body starts lowering, so the chamber's volume decreases. When the two valves are closed the body with his inertia keeps lowering, increasing temporarily the pressure. After a settling time the force reaches its nominal value and the lowering phase ends.

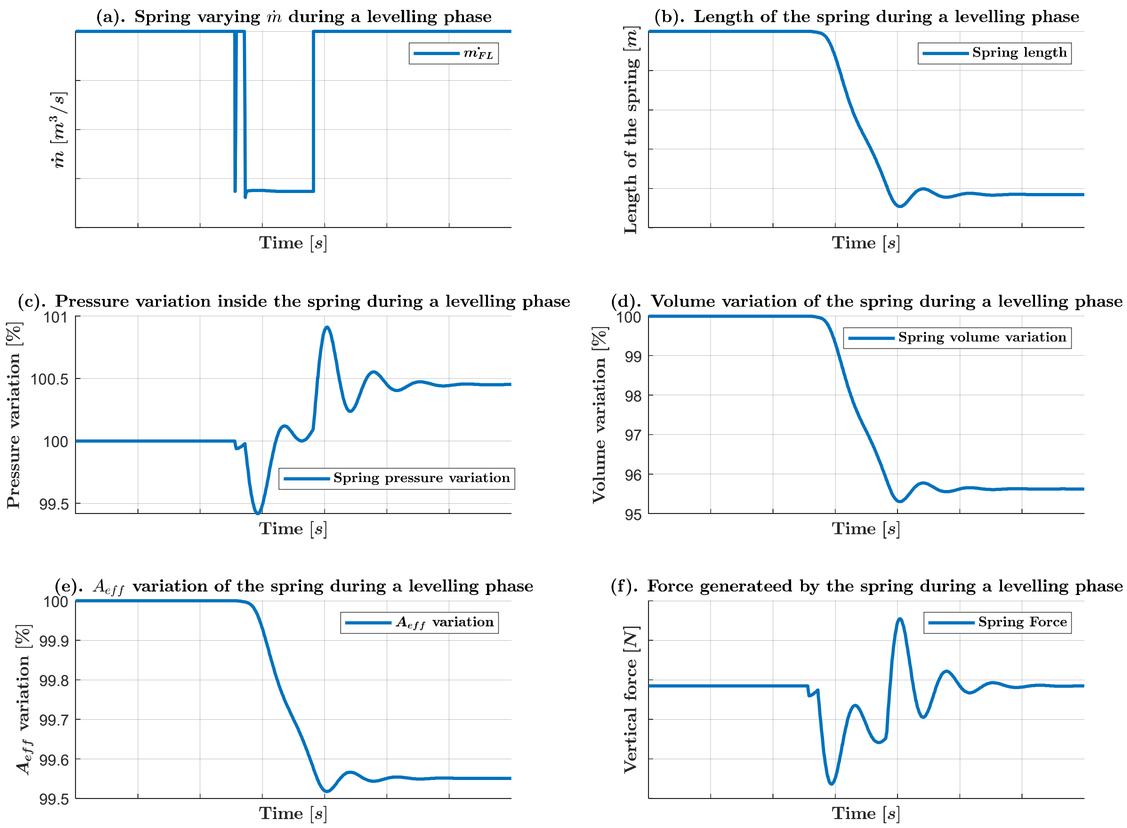


Figure 2.5 Internal spring variables during a levelling phase. The phase shown consists in a lowering manoeuvre: the exhaust valve and the suspension valves are opened together. (a) $\dot{m}$ variations, negative because air leaves the spring. (b) length of the spring decreases, the car is lowered. (c) Percentage pressure variation inside the spring during the lowering phase. (d) Percentage volume variation during the lowering phase. (e) Percentage A_{eff} variation during the lowering phase. (f) Force variation during the lowering phase

2.1.6 **Model integration**

The mathematical representation of the whole system is constituted by the integration of the blocks described before.

The complete set of equations is shown below:

$$
\begin{cases}
\dot{m}_{XY} = A_{valve_{XY}} C_m C_q \dfrac{P_u}{\sqrt{T_u}} sign(\Delta_P) \\
\dot{m}_c = \dfrac{W_c}{\dfrac{k}{k-1} R T_1 \left(\left(\dfrac{P_2}{P_1} \right)^{\frac{k-1}{k}} - 1 \right)} \\
\dot{P}_b = \dfrac{kRT_b}{V_b} \dot{m}_b \\
\dot{P}_r = \dfrac{kRT_r}{V_r} \dot{m}_r \\
\dot{T}_r = \dfrac{RT_r^2}{PV} \left(\dot{m}_{in}(k-1) - \dot{m}_{out}(1-k) \right) \\
\dot{P}_{s_{XY}} = \dfrac{kRT_{s_{XY}}}{V_{s_{XY}}} \dot{m}_{s_{XY}} - \dfrac{kP_{s_{XY}}}{V_{s_{XY}}} \dot{V}_{s_{XY}} \\
\dot{V}_{s_{\mathrm{XY}}} = \dot{A}_{eff_{\mathrm{XY}}}(s) s_{\mathrm{XY}} + \dot{s}_{XY} A_{eff_{\mathrm{XY}}}(s) \\
\ddot{s}_{XY} = \dfrac{F_{s_{XY}}}{m_{eq}} - g - \dfrac{\dot{s}_{\mathrm{XY}} \beta}{m_{eq}} + \dfrac{F_{extra_{\mathrm{XY}}}}{m_{eq}} \\
\dot{F}_{s_{\mathrm{XY}}} = \dot{A}_{eff_{\mathrm{XY}}}(s) P_{s_{\mathrm{XY}}} + \dot{P}_{s_{\mathrm{XY}}} A_{eff_{\mathrm{XY}}}(s)
\end{cases}
\tag{12}
$$

Where m_{XY} represents the generic valve, while s_{XY} the generic spring. The control inputs of the system are the areas of each valve $A_{valv\ \ XY}$ and the extra weight on the car F_{extra}. The outputs are the spring elongations s_{XY}.

To clarify the structure, a block scheme representing the set of equations and their integration is shown in Figure 2.6. The inputs are represented as arrows from the top of the picture, while the outputs are represented by the arrow pointing the bottom of the picture. The interface quantity is the $\dot{m}$, determined by the valves inside the block valve and dispatched to the other blocks.

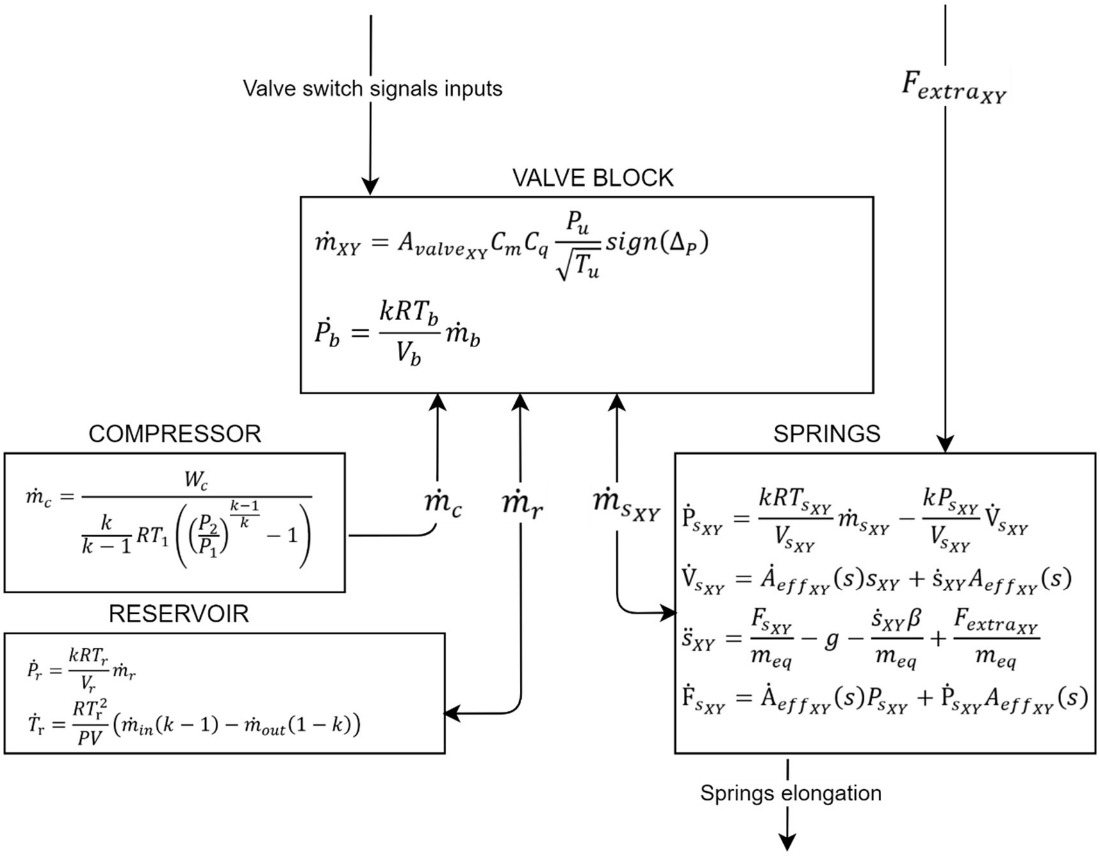


Figure 2.6 Block scheme of the integrated mathematical model. The inputs of the system are the arrows from the top of the figure, the outputs the arrow towards the bottom of the figure

2.2 **Model implementation**

The mathematical model has been implemented in a MATLAB environment. Simulink has been used to build the blocks of the differential equations. This choice is mainly because the Simulink model can be readily integrated with the dSpace suite, in order to make the model run on the SCALEXIO machine, the Real-Time machine used in Maserati for HIL tests.

2.2.1 **Valve**

In Figure 2.7 the implementation of the valve dynamics is depicted. Inputs as pressures, temperatures and the valve surface are on the left side of the block. The control input is the “Exhaust_valve_signal”, which sets the orifice cross-section area to zero (closed valve) or a constant area value (open valve).

On the right of the block the output $\dot{m}$ is shown.

The figure shows the exhaust valve, so the upstream pressure is the valve block pressure and the downstream pressure is the atmospheric pressure. The temperature of the valve block is upstream, while the atmospheric temperature is downstream.

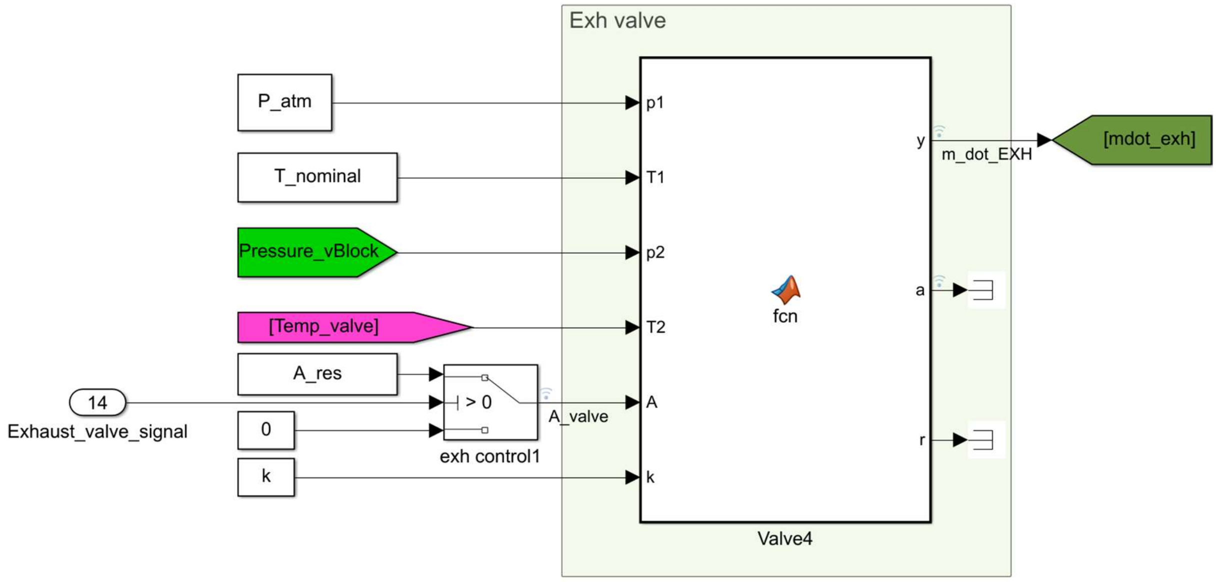


Figure 2.7 Simulink implementation of the valve dynamics, exhaust valve

In Figure 2.8 (a) plot of a simulation of the valve block alone is shown. The plot (a) represents the inputs of upstream and downstream pressures. The upstream pressure sweeps from a maximum to a minimum value (different than zero), while the downstream pressure is kept constant. Higher is the pressure ratio, higher is the airflow. When the upstream pressure is lower than the downstream one, the airflow changes sign and becomes negative.

The valve orifice area (Figure 2.8 (b)) has been set to three values per each input pressure ratio. Two of the three values have been chosen in order to be one the double of the other, the third one is zero. When the valve is closed ($A_{valve} = 0$) there is no airflow. When the valve is open, a bigger aperture corresponds to a higher airflow level.

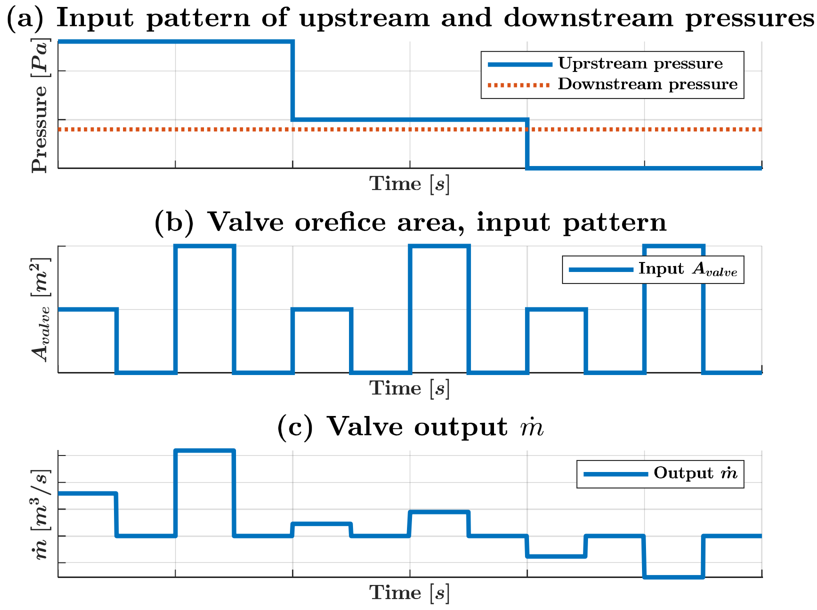


Figure 2.8 Simulation of the valve block. Input patterns: (a) upstream and downstream pressure (zoom on the zone of interest in the y axe) (b) orifice area (values: 0, 1x, 2x). Output: (c) $\dot{m}$ flowing through the valve (negative in the last part). The test shown represents the valve behaviour with, as input: three pressure ratios, three orifice area values per each pressure ratio.

2.2.2 **Valve block**

In Figure 2.9, the high level implementation of the equation (3) is depicted.

In the $\dot{P}$ block one input is the algebraic sum of all the airflows from and to each valve: having picked a sign rule the sign of the sum will determine a rising or lowering pressure. Here the interface quantity is the airflow $\dot{m}$ that flows through all the different valves, letting pressure, volume and temperature change in the block.

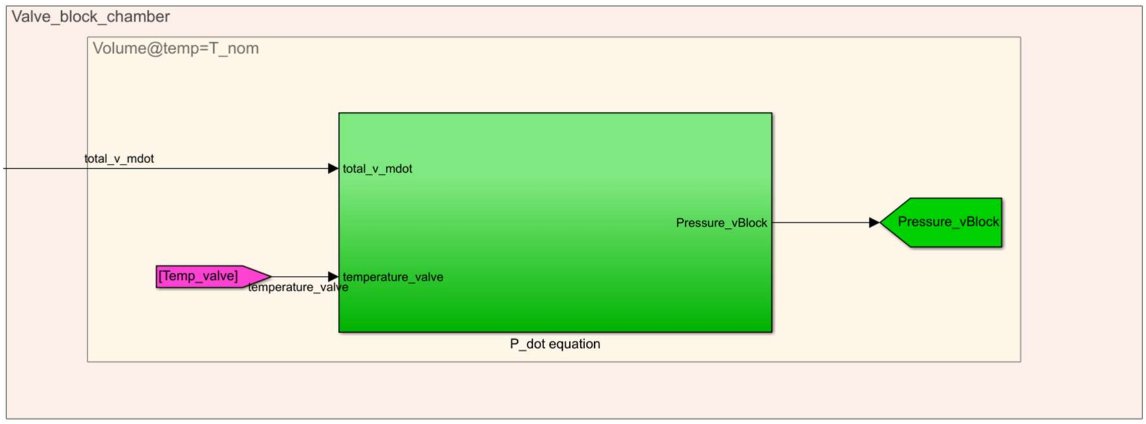


Figure 2.9 Simulink high level implementation of the valve block chamber dynamic equations

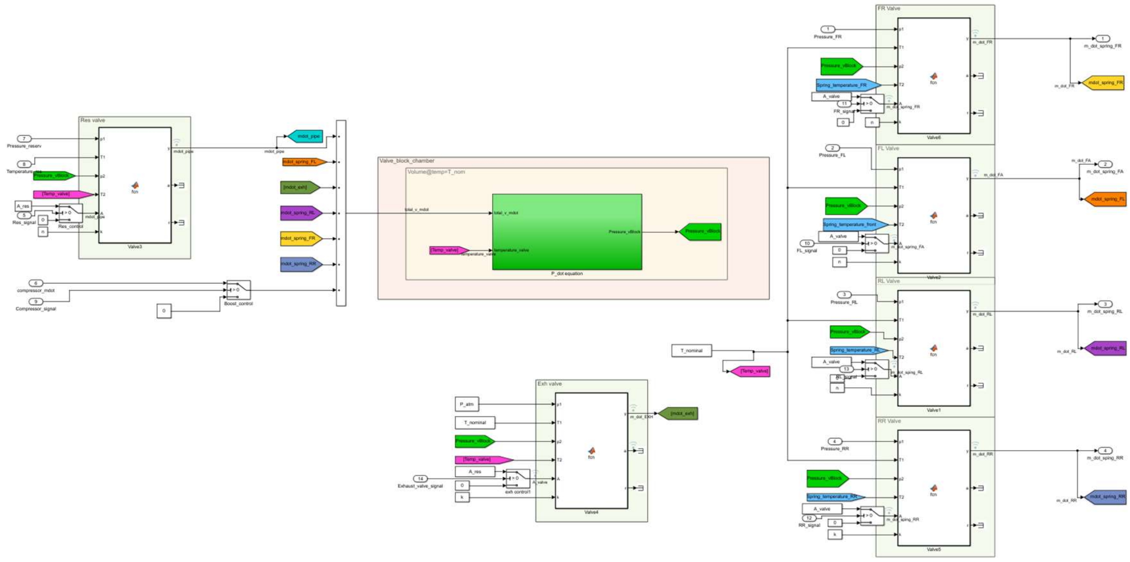


Figure 2.10 The whole valve block subsystem implementation

In Figure 2.10 the whole subsystem modelling the valve block is depicted. As the real block, the modelled one has an input interface to let each valve be driven. As output, it has all the individual airflows characterizing the system.

2.2.3 **Reservoir**

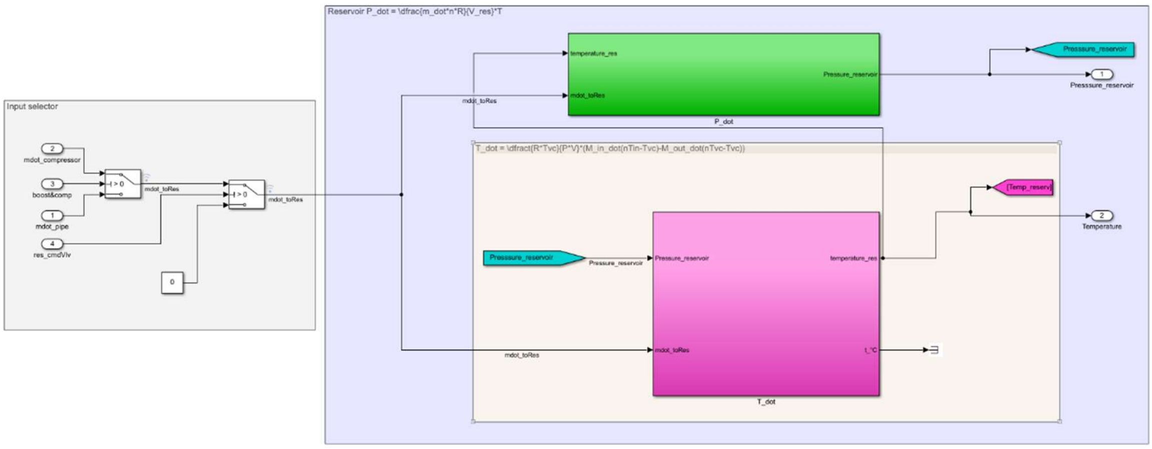


Figure 2.11 High level implementation of the reservoir subsystem

In Figure 2.11 the implementation of the valve block subsystem is depicted. The equations used are the same of the valve block chamber, one regarding the pressure dynamics and one regarding the temperature dynamics.

This subsystem is characterized by an input selector, this because depending on the ECU requests the reservoir will be emptied to rise the car or filled to bring it to the nominal working pressure.

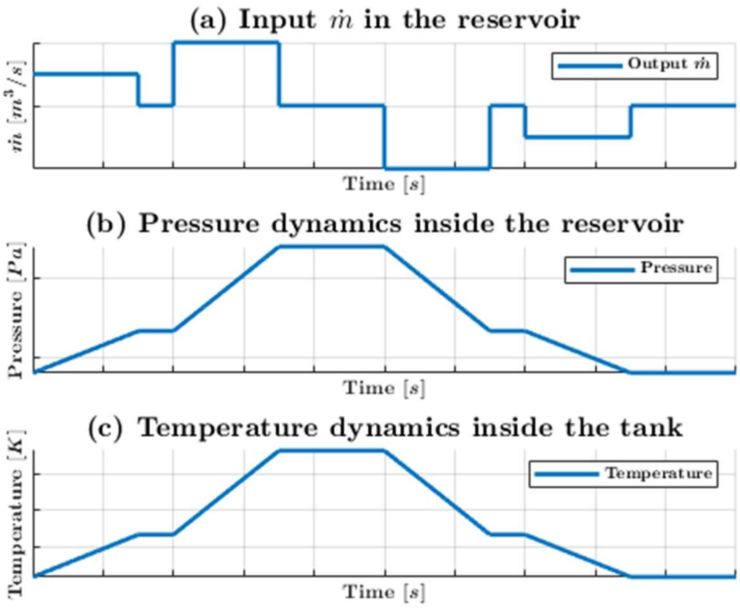


Figure 2.12 Simulation isolated reservoir, (a) input signal $\dot{m}$ (b) pressure evolution according to the input given (c) temperature evolution according to the given input. The plot represents the reservoir behaviour with, as input: two different $\dot{m}$ positive values and two values of the same magnitude but opposite sign.

In Figure 2.12 the behaviour of the reservoir block is plotted. The input signal is $\dot{m}$ varying its value among a set of positive and negative values, other than zero. Both the temperature and the pressure dynamics in (b) and (c) show a rising behaviour when the airflow is positive and a descending behaviour when the airflow is negative. The air mass is added and removed in the same amount; since the process is isentropic, both pressure and temperature in the final instant are the same as at the beginning of the simulation.

2.2.4 **Compressor**

In Figure 2.13 the high-level implementation of the system is depicted.

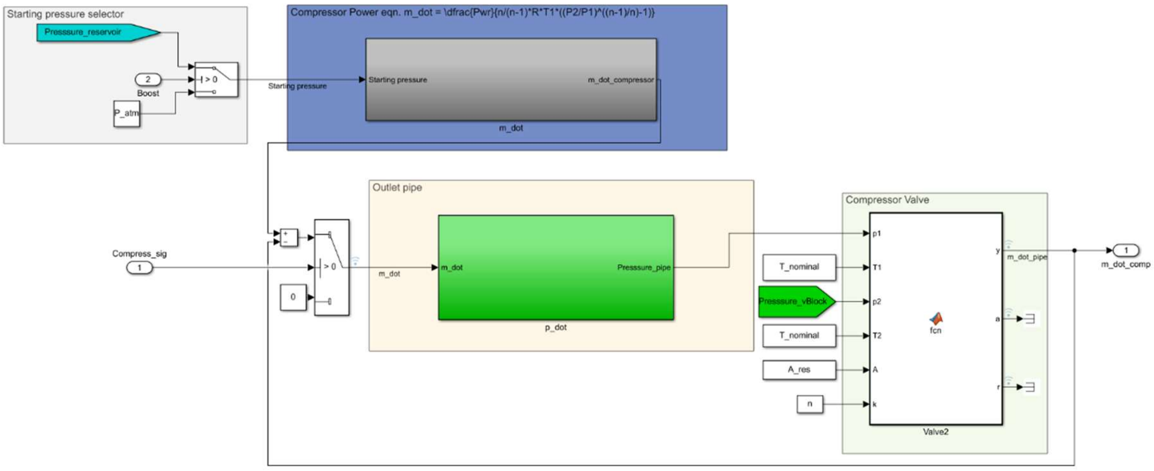


Figure 2.13 High-level implementation of the compressor subsystem

The compressor subsystem is built putting together equation (10), a pipe block and a valve block. The pipe is modelled as a fixed volume chamber, so the equation (3) is used, the valve block is the same as the other valves described. These two additional blocks are added to better characterize the real system and to simplify the implementation, having a valve at the interface level as in the other subsystems.

2.2.5 **Spring**

In Figure 2.14 the high-level implementation of the spring block is depicted. The block has two inputs: the airflow $\dot{m}$ and the extra vertical force *Extra_weight.* The latter is an input of the "s_dot_dot" block, the block implementing the third equation in (11). It corresponds to F_{extra} in the equation, that is an input of the whole system.

The outputs are grouped in the grey square in the bottom-right part of the figure.

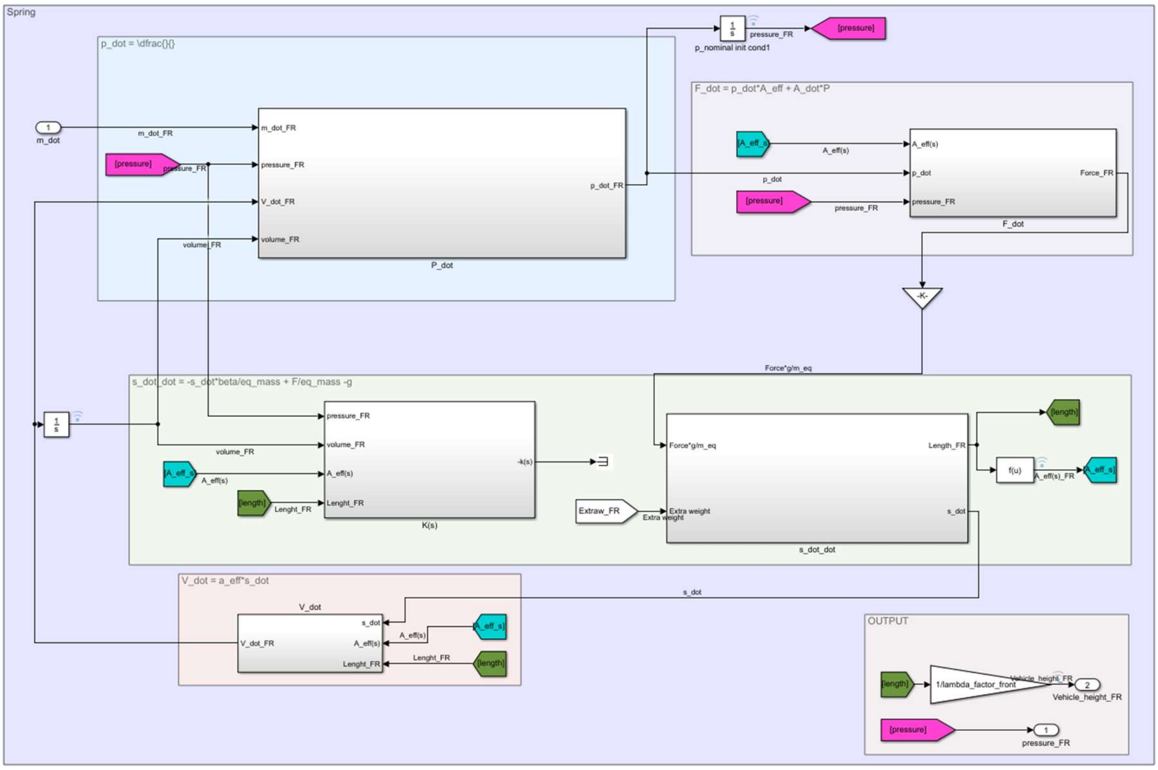


Figure 2.14 High-level implementation of the spring dynamics

The vehicle height is evaluated as: $\mathrm{h_{veh}} = \frac{\mathrm{s}}{\lambda_\mathrm{f}}$. Where s is the spring elongation and the coefficient λ_f is used to model the suspension scheme: a δ variation in the suspension length, will correspond to a $\frac{\delta}{\lambda_\mathrm{f}}$ variation of the vehicle height, since the suspension is not perfectly vertical.

An additional block is present in the figure, the one that evaluates the spring coefficient $k(s)$ [3]. In the activities covered in this document this value will never be used, however in the future this model may be integrated with a model that simulates the whole vehicle dynamics, in that case this coefficient will be useful to characterize the spring dynamics.

2.2.6 **Integration**

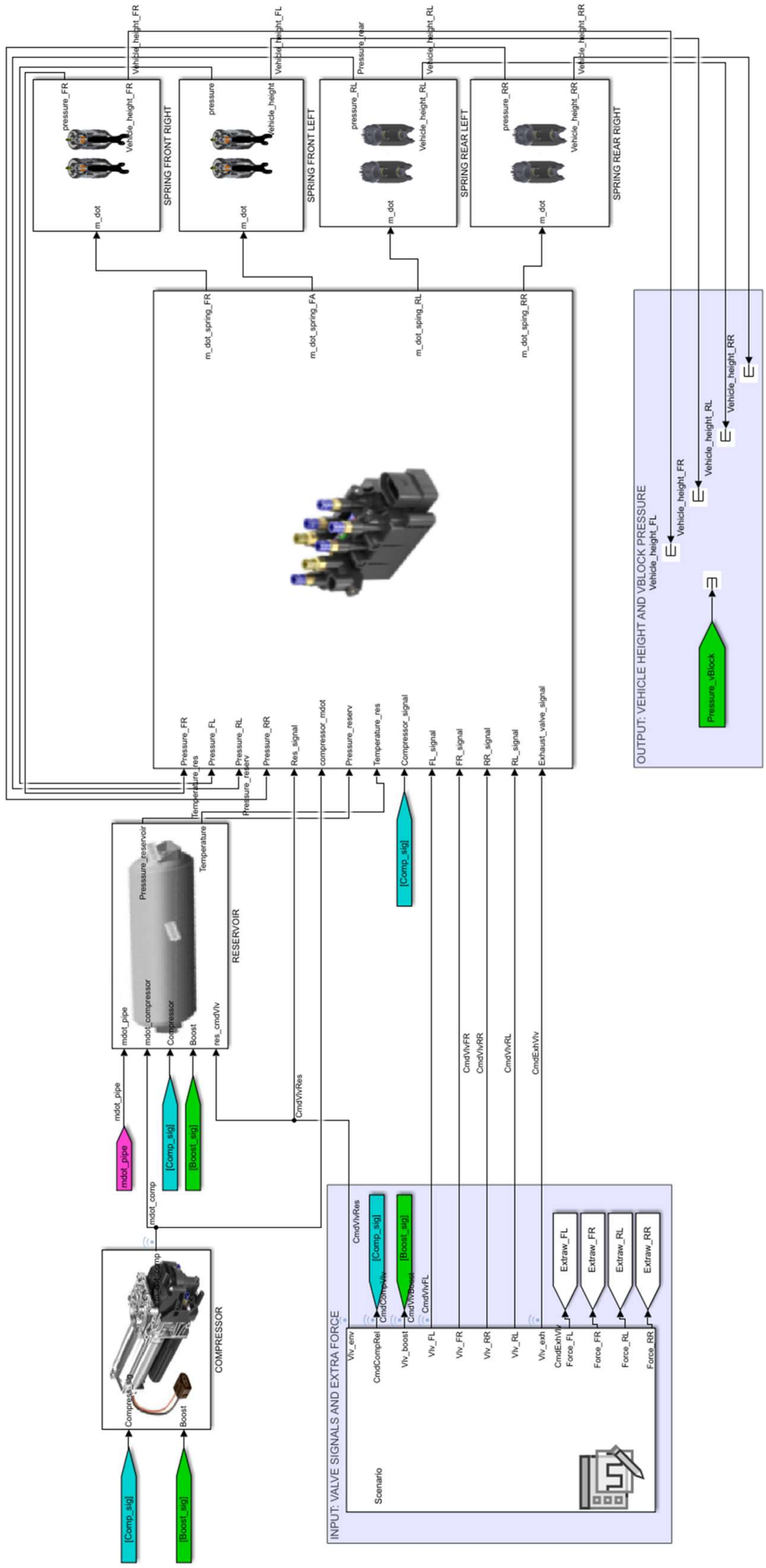


Figure 2.15 High-level implementation of the whole model

The chambers present in the model are linked together thanks to the valve blocks. Each valve block, given the thermodynamic quantities characterizing the two linked chambers, will return as output the $\dot{m}$ that flows between the two. Each chamber block, given the entering or exiting airflow, evaluates the thermodynamic transformations having place inside the chamber.

In Figure 2.15 the high-level architecture of the whole system is depicted. In the left light blue square the inputs of the model are highlighted, while in the light blue square in the bottom there are the outputs that the ECU will be fed with during the test phase.

The pressure of the chamber of the valve block is an input for the ECU, this because the only pressure sensor present in the whole system is located there in the actual system.

2.3 Model validation

The model has been validated in two phases.

In the first phase the blocks, isolated were fed with various input patterns, so that their outputs could be analysed. This phase produced a qualitative validation and all the blocks passed this phase, since their behaviour was coherent with the starting equations.

In the second phase, data acquisitions of the real system were available, so the integrated model's output have been compared with this data. In this phase the model has been validated on a quantitative basis.

An ad hoc data collection was not possible. Some external factors that can affect the system behaviour are unknown.

To bring the simulations output and the experimental data as close as possible, some initial parameters have been changed/set accordingly (e.g. the nominal pressure of the springs: higher if extra weight as people and a full fuel tank is present on the car).

In this section the comparisons between the simulation results and the experimental data will be shown. Each levelling mode will be analysed, for each one of them the valve block pressure signals and the vehicle height signals will be compared.

2.3.1 **Reservoir only levelling**

In this mode the car is lifted thanks to the highly pressurized air flowing from the reservoir to the springs.

In Figure 2.16 the simulated valve block pressure and the measured one are compared. The plotted pressure signals are relative to a rising manoeuvre. The mismatch highlighted in the comparison is considered to be negligible for this work purposes: the ECU needs this signal to decide which is the correct levelling mode and to ensure the occupants safety; numerically the mismatch is low enough to guarantee a meaningful HIL test of the ECU functionalities.

This plot shows the following situation:

- at t_0 the block is at the residual exhaust pressure;
- at t_1 the reservoir valve is open and the block reaches the reservoir pressure
- at t_2 the reservoir and both the rear spring valves are open, so the car's back axle starts to rise and the pressure inside the block has an average value between the reservoir and the spring
- at t_3 the rear spring valves close and the front spring valves open so the front axle starts rising and the pressure has an average value between the reservoir and the spring
- at t_4 only one rear spring valve is open
- at t_5 only the reservoir valve is open
- at t_6 only the exhaust valve is open and the valve block reaches again the built in residual pressure.

The simulation curve lands on the constructor-declared residual pressure, while in the acquisition signal that pressure is reached after a longer settling time.

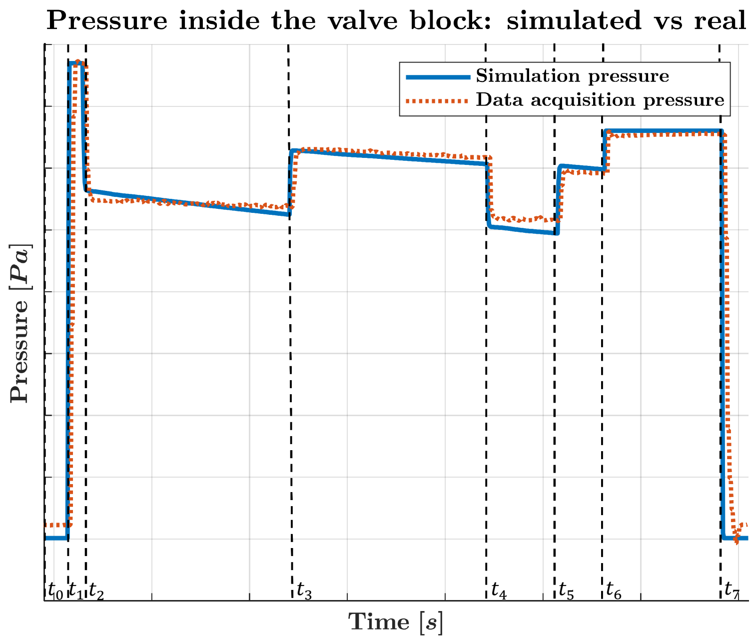


Figure 2.16 Simulation vs Real data comparison. Pressure inside the valve block during a rising manoeuvre

In Figure 2.17 the valve block pressure signals during two consecutive lowering manoeuvres are compared.

During this phase of the work some simulation outputs delivered from the springs vendor were available. The spring vendor has an in-house-made simulation model, that (they use to) is used to dimension some parts of the whole system and to design a part of the controller installed on the ECU. Since the outputs of the model object of this work and the vendor's simulations show a similar behaviour, even if the curves in Figure 2.17 have different transients, this part of the model has been considered valid.

As a result, during the HIL tests the ECU's behaviour was not affected by the difference between the simulation output and the acquisition data.

The situation described by the plot is the following:

- at t_0 the valve block is at the residual pressure
- at t_1 only the front spring valves are open, so the valve block is at the front spring nominal pressure
- at t_2 also the exhaust valve is open so that the front axle stars to lower and the pressure inside the valve block is at an average level between the atmospheric pressure and the one inside the front springs

- at t_3 the rear spring valves are opened together with the exhaust valve and the rear axle starts lowering
- at t_4 the same operation as t_2 and t_3 is performed again twice
- at t_5 only one rear suspension valve is open together with the exhaust valve
- at t_6 only the exhaust valve is open and the valve block reaches the residual pressure.

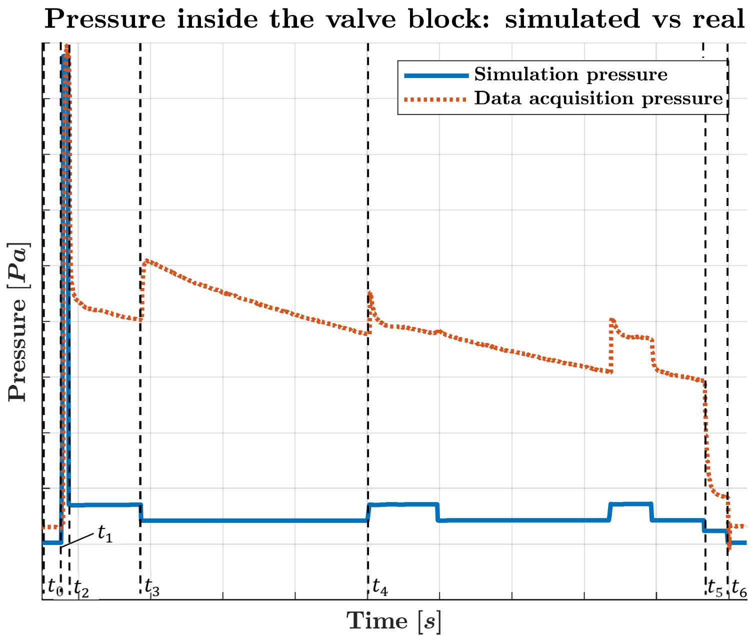


Figure 2.17 Simulation vs Real data comparison. Pressure inside the valve block during two consecutive lowering manoeuvres

One possible explanation for the strange behaviour of the measured signal is that the mounting position of the sensor let there be a measuring error: the airflow hitting the sensor during a lowering manoeuvre might deform the sensor so to read a wrong higher pressure.

In Figure 2.18 the comparison between the simulation output and the measured data regarding the front left corner's height is shown. The simulated and collected data present a low mismatch, less than a millimetre, which is the ECU smallest measured value. Hence the simulation output is considered valid.

In Figure 2.19 the comparison between the simulation output "vehicle height Rear Right" and the real data corresponding to the same quantity is shown. The mismatch is higher than in the front case but still around the millimetre, so the simulation data is considered acceptable.

It is worth mentioning that the compared shown signals regard just one corner per axle. This because in the simulation the car is on a perfectly plane surface with the weight symmetrically balanced on the two sides, as in the measurement situation.

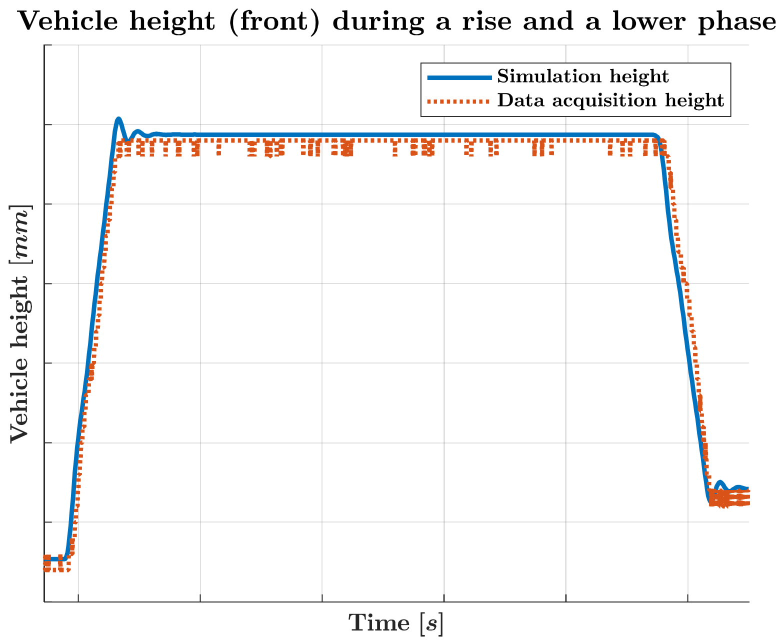


Figure 2.18 Vehicle height (front) during a rising and a lowering phase. The height is measured as the difference between the current height and the height in "normal" mode

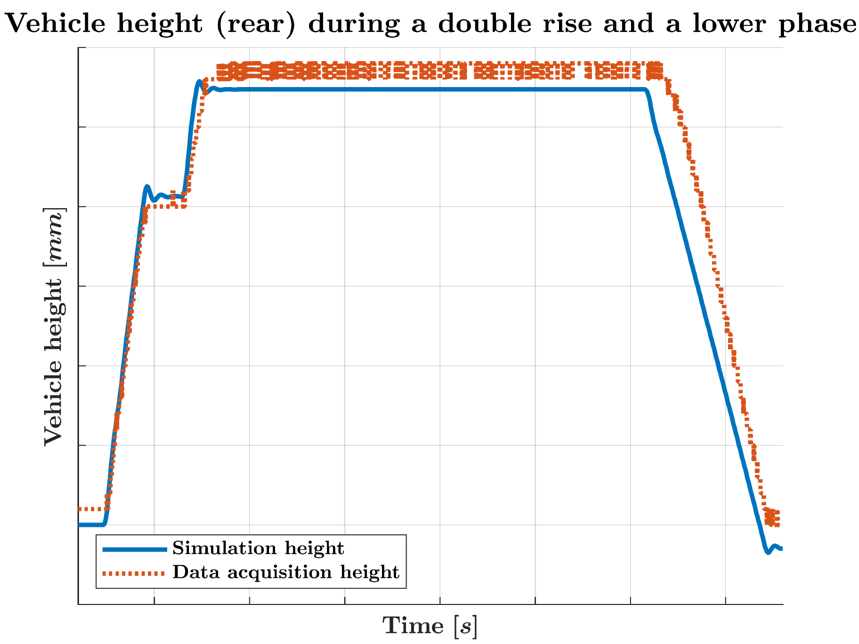


Figure 2.19 Vehicle height (rear) during a rising and a lowering phase. The height is measured as the difference between the current height and the height in "normal" mode

2.3.2 **Boost levelling**

Figure 2.20 shows a comparison between the simulated pressure inside the valve block and the actual measured pressure, during a boost mode lifting. The plot regards a lift of the rear and front axle.

The situation described by the plot is the following:

- at t_0 the valve block is at the residual pressure
- at t_1 the boost valve, the compressor and the rear spring valves are open, so the pressure rises and the real axes starts lifting
- at t_2 the rear spring valves are opened together with the front spring valves.
- at t_3 the front spring valves are open so the front axle starts lifting
- at t_4 only the exhaust valve is open so the block reaches the residual pressure
- at t_5 only the reservoir valve is open, so that the valve block reaches the reservoir pressure, and the ECU can acknowledge it.
- at t_6 only the exhaust valve is open so the block reaches the residual pressure

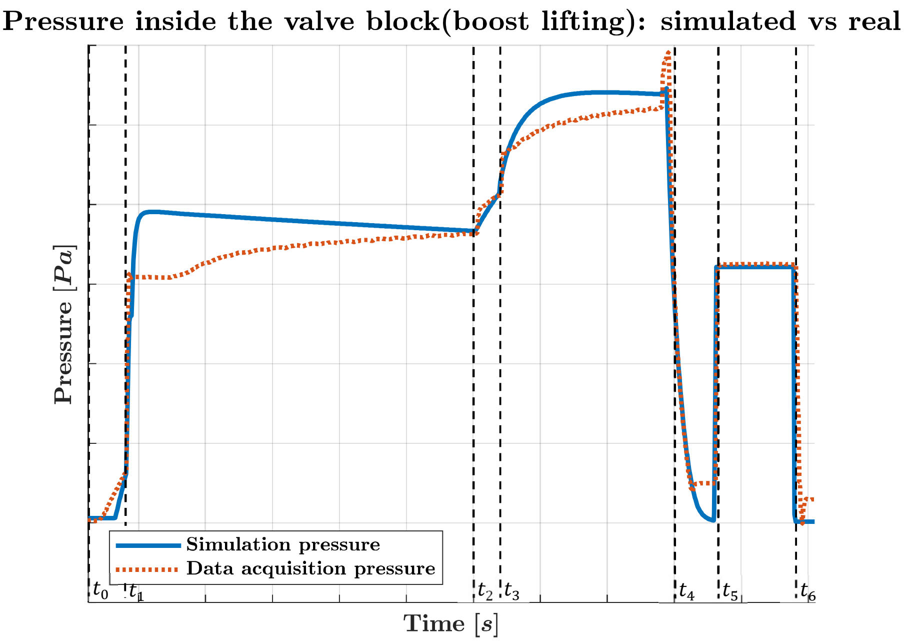


Figure 2.20 Pressure inside the valve block during a boost lifting phase. Simulated vs real.

In this plot the transient behaviour of the pressure shows some differences among the simulated and the measured signals.

During a test, the ECU reads this signal and checks that it does not rise dangerously fast or does not exceed the safe limits. The ECU also checks the value of this signal to decide the most appropriate levelling mode. Because of these reasons, even if the transient dynamics of the simulation outputs do not exactly correspond to the actual dynamics, this simulation output is acceptable for this work's goals.

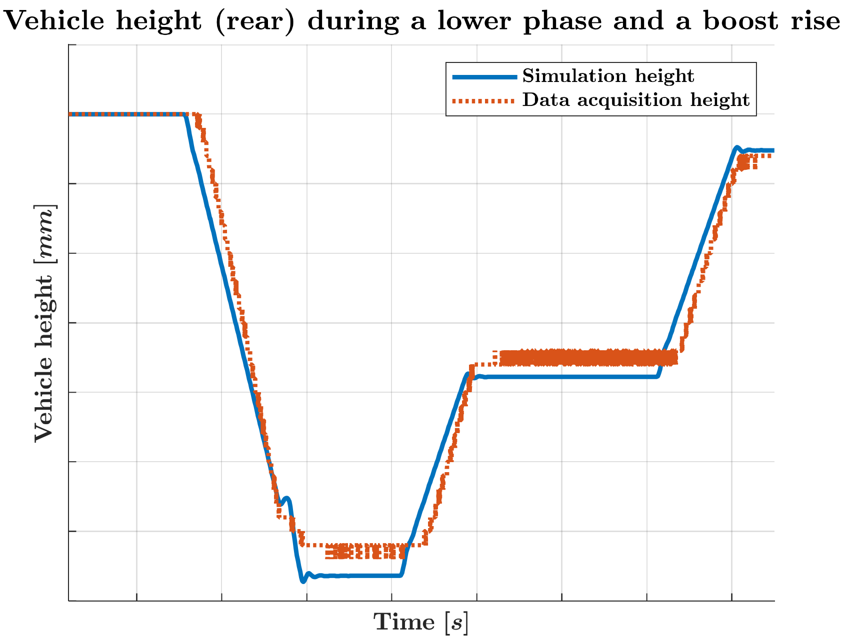


Figure 2.21 Rear axle vehicle height, during a lowering phase and two subsequent boost rising phases. Simulation vs Real

In Figure 2.21 and Figure 2.22 the plots of the vehicle height comparisons are shown. Also in this case the differences are acceptable, since they are lower than the ECU sensibility.

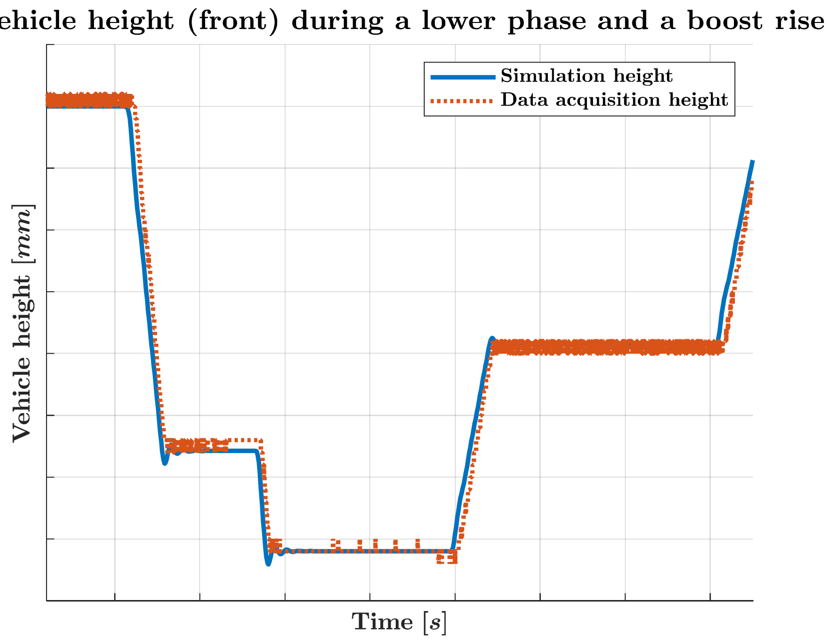


Figure 2.22 Front axle vehicle height, during two lowering phase and two subsequent boost rising phases. Simulation vs Real

2.3.3 **Compressor levelling**

In Figure 2.23a comparison between the simulated pressure inside the valve block and the actual measured pressure is shown. The plot regards a lift of the rear and front axle.

The situation described by the plot is the following:

- at t_0 the valve block is at the residual pressure
- at t_1 the compressor and the rear spring valves are open, so the pressure rises and the real axes starts lifting
- at t_2 the rear spring valves are opened together with the front spring valves.
- at t_3 the front spring valves are open so the front axle starts lifting
- at t_4 only the exhaust valve is open so the block reaches the residual pressure

The simulation pressure dynamics are close enough to the measured data to be used in this work.

In Figure 2.24 and Figure 2.25 the vehicle height signals comparisons are plotted. This result is acceptable since the differences fall inside the tolerance interval, set to the ECU sensibility.

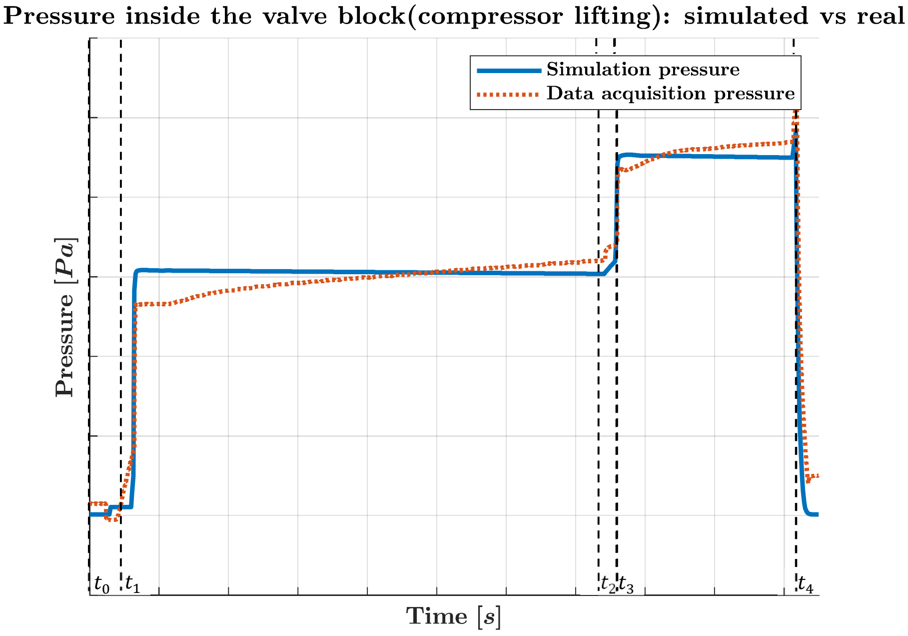


Figure 2.23 Pressure inside the valve block during a compressor lifting phase. Simulated vs real.

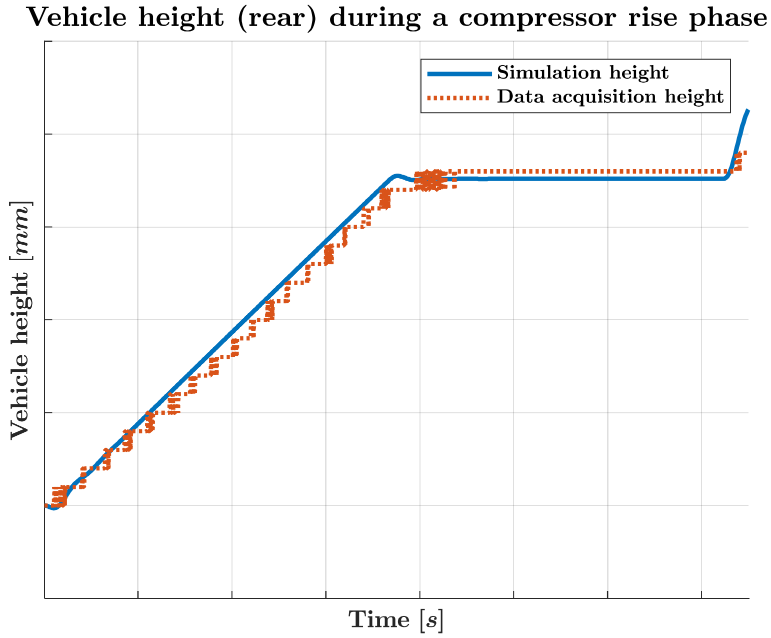


Figure 2.24 Rear axle vehicle height, during compressor rising phase. Simulation vs Real

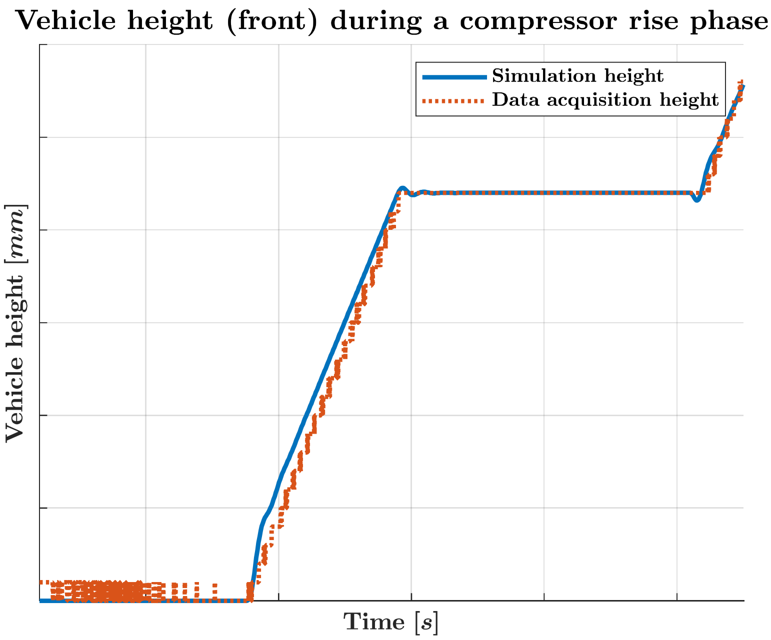


Figure 2.25 Front axle vehicle height, compressor rising phase. Simulation vs Measurement

3. HIL tests

In this chapter the setup, the design and the results of the Hardware-in-the-loop tests are going to be discussed.

The dynamical model ran on a real-time machine connected to the target ECU.

To handle the communication with the ECU a low-level input-output interface was given. The hardware level of the interface consists in expansion slots for the SCALEXIO machine, one per communication standard. The software level consists in a Simulink program that is responsible for:

- Ensuring the communication between the human interface and the ECU. Moreover, it sends signals to monitor the correct behaviour of each ECU's functionality.
- Sending the safety messages expected by the ECU. In absence of these messages, the ECU would rise faults that prevent its functionalities the system to work.

Figure 3.1 HIL bench integration. The dynamical model and the SW of the IO interface are merged into one Simulink model, that is loaded into the SCALEXIO machine to run the tests.

3.1 HIL integration

The first step to run the tests was the integration of the dynamical model with the abovementioned IO handler.

The integration is depicted in the block diagram of Figure 3.2. The block in the left represents the dynamical model of the air spring system, as stated before its inputs are the valve switch signals and its outputs are the pressure inside the valve block and the height of each vehicle corner.

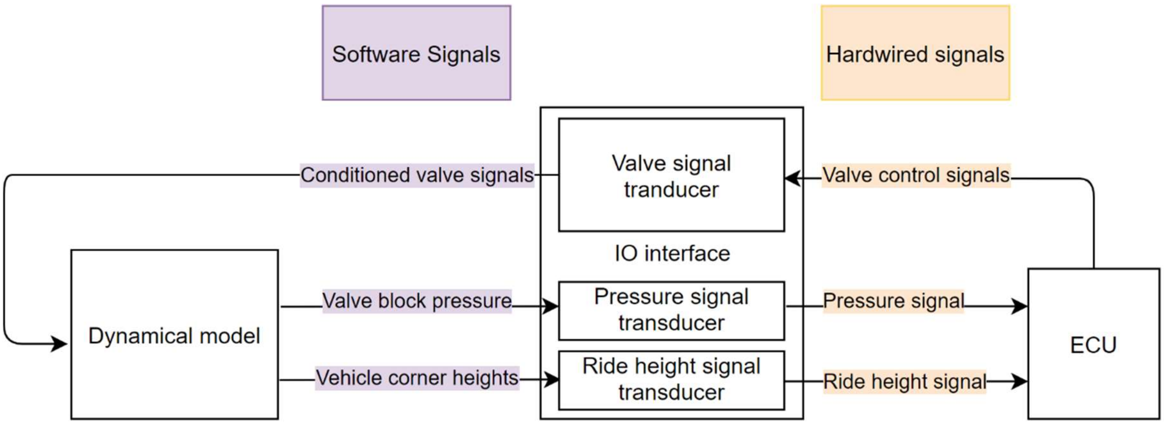


Figure 3.2 Block diagram of the integrated hardware software test bench

The outputs of the dynamical model are inputs for the “Ride height signal transducer” and the “Pressure signal transducer” blocks. The purpose of this blocks is to take a numerical input and translate it as an electric signal, the ECU perceives these signals as sent from an actual ride height sensor and an actual pressure sensor.

The ECU takes these signals as input and thanks to the implemented control law outputs a vector indicating the required status for each valve. These electrical signals are read and conditioned by the “valve signal transducer” block, so that they can be read by the dynamical model block.

The mentioned block diagram highlights the used Hardware-in-the-loop structure.

3.2 Test design and implementation

In this subsection the used test cases are going to be discussed.

The test cases regard the functions of the ECU, which are:

- Level the vehicle after a driver demand
- Level the vehicle automatically, if a load is detected or the vehicle’s speed falls into specified intervals
- Use the compressor to fill-up the reservoir in some specific situations

The tests have been implemented using the AutomationDesk software, from dSpace. The software provides an interface to write the test logic, to run, and to monitor the running tests. Additionally, an automatic test report can be generated.

All the tests are based on the basic structure depicted in Figure 3.3. The process starts setting the proper boundary conditions, this means exploiting the IO handler to simulate a specific external environment for the ECU. When the boundary conditions are set, a signal is sent, and the ECU response is checked. The report will be filled with the results of the comparisons between the expected and the actual responses.

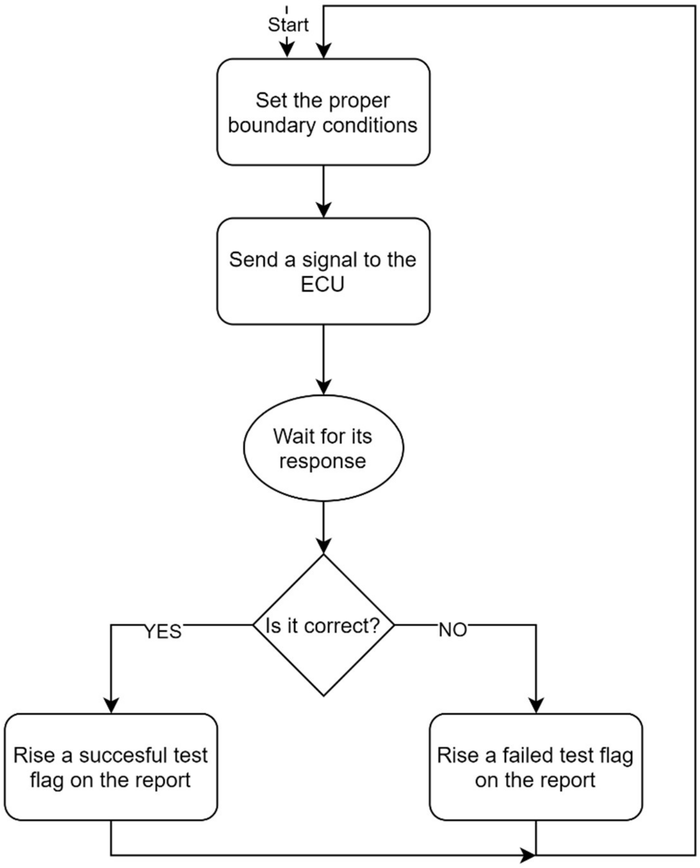


Figure 3.3 Block diagram of the basic structure the tests rely on

This structure will be adapted to each performed test, to describe their implementation. In each of the following tests, the “wait for response” phase will end after a time threshold. This threshold corresponds to the maximum allowable time needed for a levelling phase or, in the case of the reservoir filling test, in the maximum allowable time needed to fill the reservoir. The signals defining the boundary conditions are:

- Vehicle longitudinal speed
- Extra force on the vehicle

- RHS signal
- Initial reservoir pressure

3.2.1 **The driver selects a specific ride height value.**

In the Maserati cars equipped with the air spring optional, the driver can manually select a specific ride height value, that the car is required to reach. The signal used for this purpose can be sent using the IO handler, from now this signal will be called RHS, as Ride Height Selector.

To test this functionality the following test case has been set:

- Vehicle longitudinal speed: null.
- Extra force on the vehicle: null
- RHS signal: varying to achieve all possible ride height values.
- Initial reservoir pressure: three different levels to try different levelling modes.

This test has also been used to understand the ECU capability to switch among levelling modes so it ran three times, one per levelling mode.

This test goals are:

- Check that the ECU is able to receive the RHS signal and acknowledge it
- Check that the ECU addresses the correct level height.
- Check that during the whole levelling phase the ECU correctly sets its output signals
- Check that the control law implemented on the ECU allows it to reach the requested level height.
- Check that the ECU senses correctly the reservoir pressure and selects the most appropriate levelling mode.

Additionally, this test was the first being launched so its purpose was also to understand if the ECU, before allowing a levelling procedure, performed the required checks on the on board sensors.

The available ride heights levels are AERO_2, AERO_1, NORMAL, OFFROAD_1, OFFROAD_2, LIFTER. Each one of them is associated to an offset, in mm, with respect to a calibrated height. The calibration takes place at the end of the assembly line. The default ride heigh, corresponding to 0 mm offset, is the NORMAL level.

In Figure 3.4 the detailed block scheme of the test is depicted. First, the boundary conditions are set; then the chosen ride height value is requested to the ECU. The monitored signals are: [CurrentStatus DesiredLevel XXCornerLvl]. These signals are sent by the ECU; their values must be checked during and after a levelling phase.

The CurrentStatus signal contains information about the status of the springs: during a levelling phase will indicate if the car is rising or lowering, while after the levelling phase it will indicate the current ride height value. If any failure is detected, this signal assumes a value to indicate a generic failure. To determine the ride height, each corner height is measured. If the four corners are not at the same height (with some tolerance), the status of the system is *system fail*.

The DesiredLevel signal, contains information about the level that the car must reach after a levelling phase. This signal is sent by the ECU, the target level is evaluated by the ECU when an automatic levelling occurs. On the other hand, when the driver selects a target ride height, this signal is expected to assume that value.

The XXCornerLvl signals (XX stands for the generic corner) contain information about the current height level of each corner, expressed in millimetres.

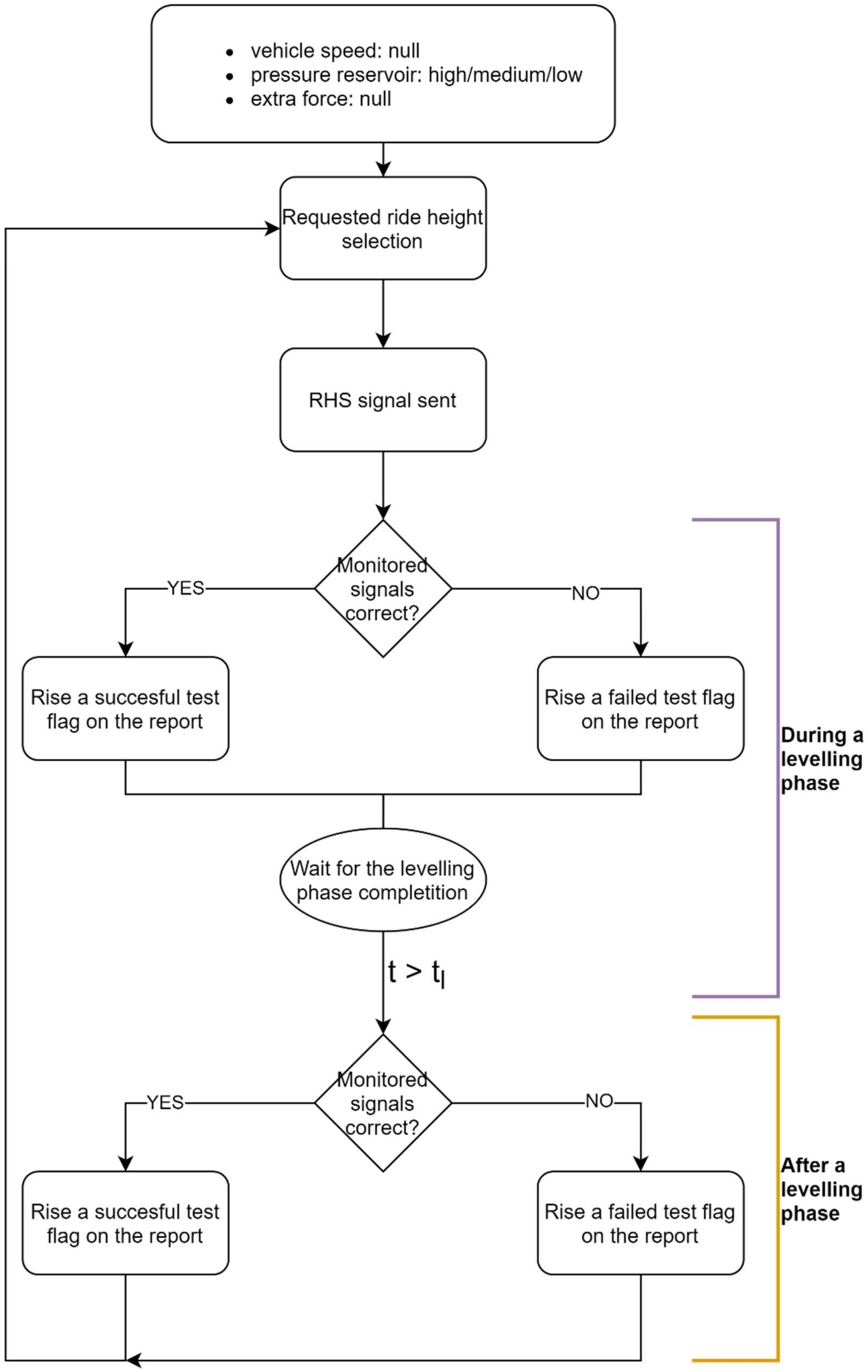


Figure 3.4 Detailed block scheme of the test used to check the behaviour of the ECU when the target ride height is selected by the driver

Figure 3.5(a) shows the evolution in time of the monitored ECU outputs DesiredLevel and CurrentStatus, together with the signal RHS, sent from the SCALEXIO machine.

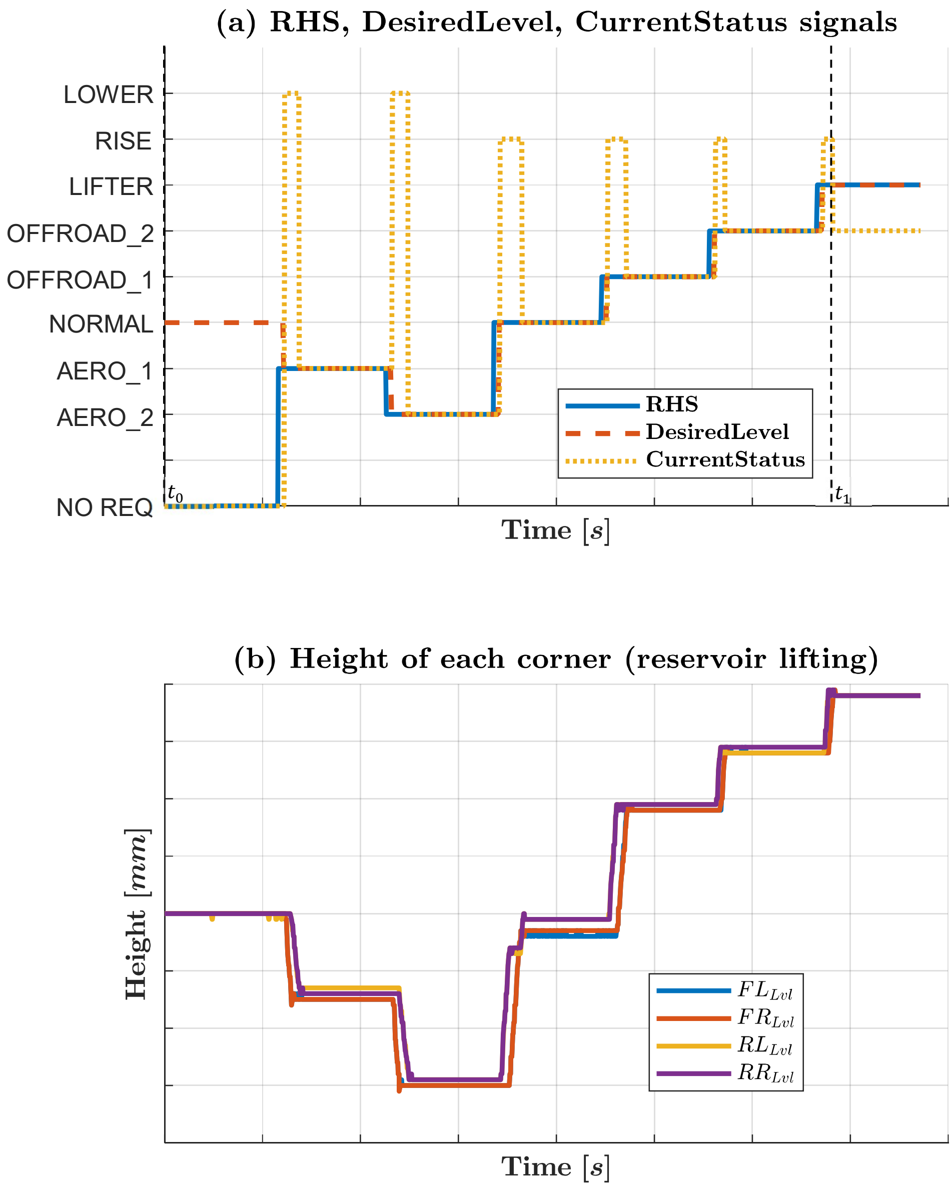


Figure 3.5 Manual ride heigh selection, test logs. (a) RHS signal, input for the ECU; DesiredLevel and CurrentStatus output of the ECU. (b) Height of each corner of the vehicle, expressed in mm as difference to the NORMAL level (@ time t_0), which corresponds to 0mm.

The plot shows how when the RHS signal is sent, after a short delay, the DesiredLevel signal is set to the same value as RHS and the CurrentStatus signal is set to RISE or LOWER, indicating a levelling phase. In this plot and in the analogous following ones, the signal CurrentStatus has entirely been plotted together with RHS and DesiredLevel, even if, unlike the others, it carries information about vertical motion of the vehicle (lowering and rising). This choice has been made to clearly visualize the delays between driver's requests and the ECU response.

At instant t_0 RHS and DesiredLevel do not concide, this is an expected behaviour since when the RHS is set to NO_REQUEST the desired level should be normal.

At instant t_1 the value of the CurrentStatus signal is set to OFFROAD_2 when it is expected to be LIFTER. In Figure 3.5(b) the height of each corner is shown, the signals plotted are the abovementioned XXCornerHeight. This sublot shows that at t_1 the vehicle corners are correctly brought to the LIFTER height. The considerations above led to report a bug in the ECU software: when in LIFTER only the CurrentStatus signal contains wrong information.

Figure 3.6 shows how the bug emerged in the test report. The checks performed in the levelling phase (653 and 654) are successful. Among the tests performed after the levelling phase, the only one that failed is the comparison between the RHS value and the CurrentStatus value. This test, highlighting a bug, proved the HIL bench to be useful and the test case design to be effective. Since in this bench the ECU controls a validated plant model, it is safe to say that the same bug would occur with the ECU mounted on board of the vehicle. Conversely, if a Not-Dynamic-HIL bench was used, the tester would not know anything about the ride height signal during the occurrence of this bug, so further tests would be needed. Tests performed onto the assembled vehicle could have spotted this bug, anyway that kind of tests is a lot more expensive than HIL testing, so it is preferrable to exploit in-vehicle tests to validate the user experience of the system. Moreover, HIL tests can be automated, as in this case, to furtherly decrease the test costs.

As mentioned before this test has been performed three times, one per lifting mode. Figure 3.7 and Figure 3.8 show the same data as Figure 3.5, but in boost and compressor only levelling modes.

The currently reached level is: OR_2
The currently chosen level is: Lifter while the current level is OR_2

653 Check

Passed *Check*
Result of comparison between the Desired Level and the Selected Level

654 Check

Passed *Check*
Result of comparison between current Level and RISING

663 Check

Passed *Check*
Result of comparison bwtween the Desired Level and the Selected Level

The currently reached level is: OR_2

664 Check

Failed *Check*
Result of comparison bwtween the Final and the Selected Level

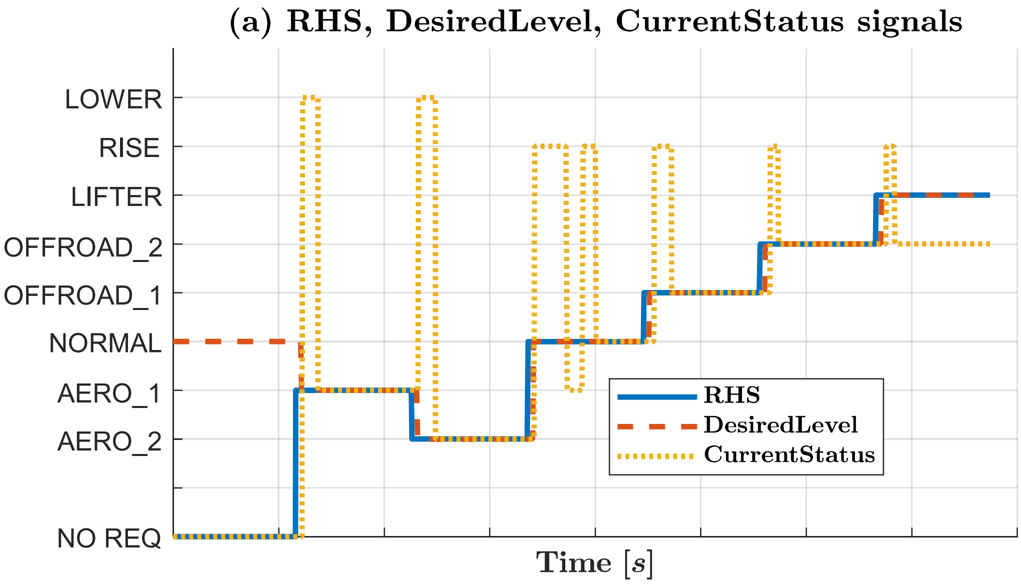
The currently reached level is: OR_2

665 Check

Passed *Check*
Checking correct height

The currently reached level is: Lifter

Figure 3.6 Exctract from the test report. During levelling phase checks: Check 653: comparison between DesiredLevel and RHS. Check 654: comparison between CurrentStatus value and RISING. After levelling phase check: Check 663: comparison between DesiredLevel and RHS. Check 664: comparison between CurrentStatus value and RHS value. Check 665: compare the corners height in mm with the height in mm associated to Lifter.



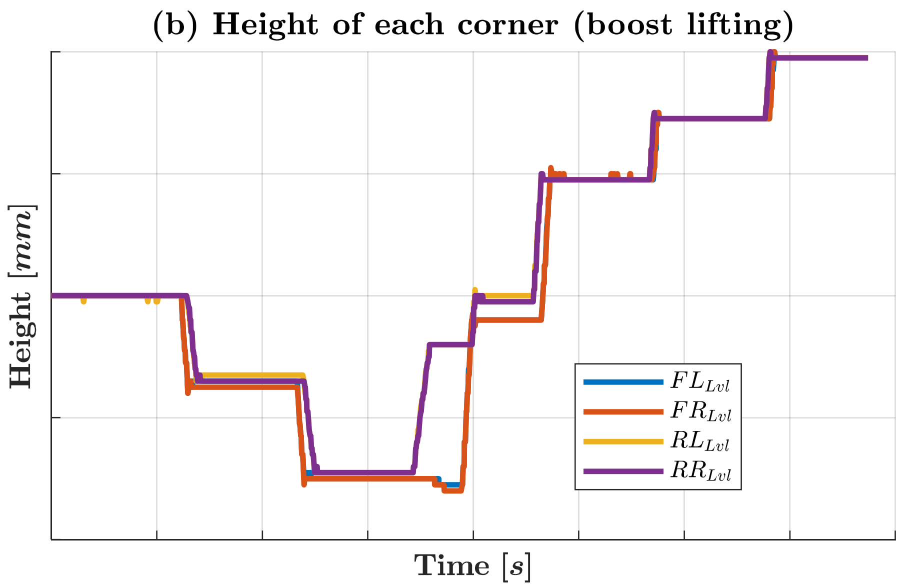


Figure 3.7 Boost levelling. (a) RHS signal, input to the ECU; DesiredLevel and CurrentStatus output of the ECU. (b) Height of each corner of the vehicle, expressed in mm as difference to the NORMAL level (@ time t_0), which corresponds to 0mm.

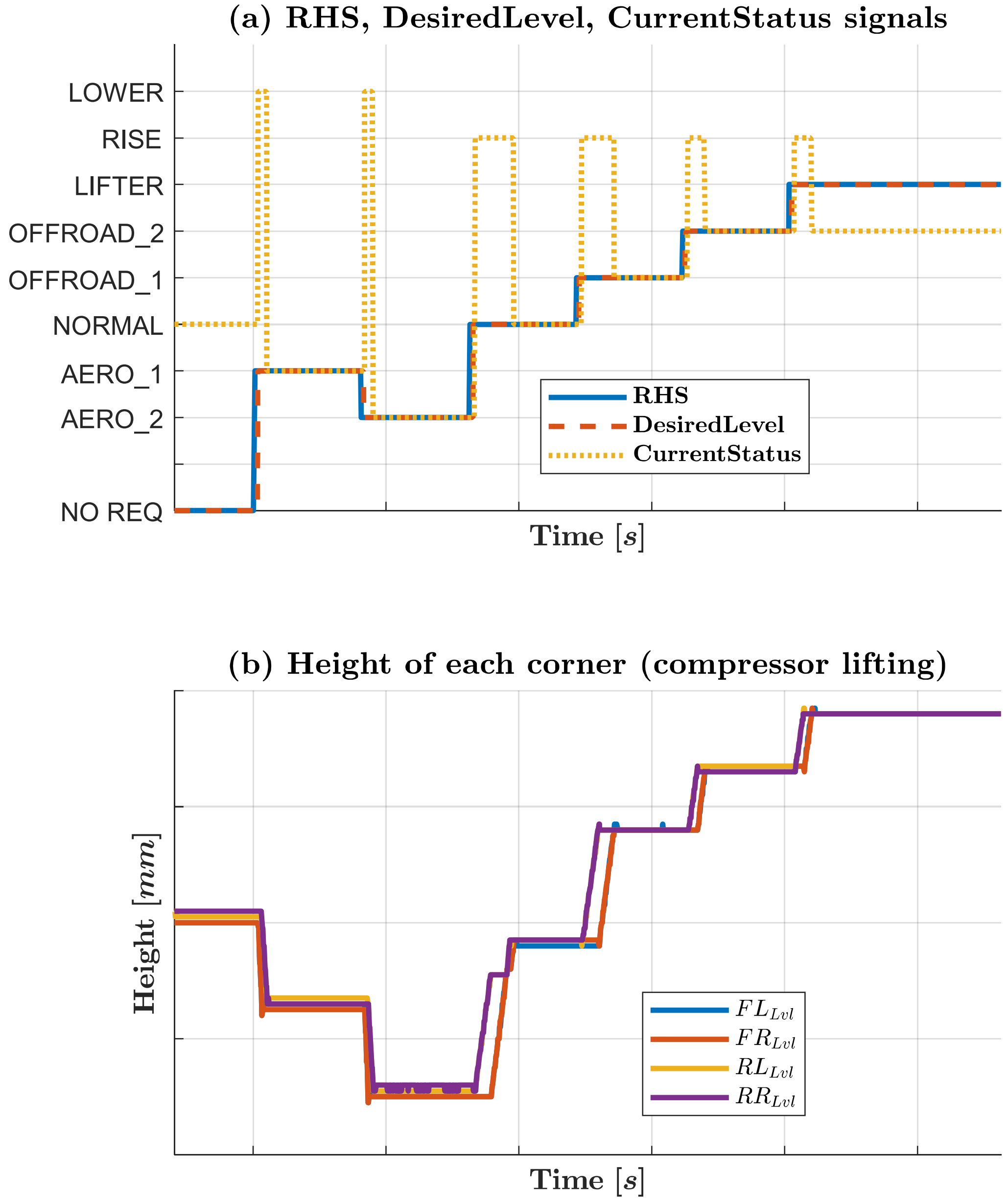


Figure 3.8 Compressor leveling. (a) RHS signal, input to the ECU; DesiredLevel and CurrentStatus output of the ECU. (b) Height of each corner of the vehicle, expressed in mm as difference to the NORMAL level (@ time t_0), which corresponds to 0mm.

In all the reported test logs the “LIFTER bug” is shown. Figure 3.7 and Figure 3.8 show how the compressor and boost modes are slower in the lifting phase than the reservoir only mode. The CurrentStatus signal, during reservoir only liftings, assumes the value RISING for a shorter amount of time.

3.2.2 **The driver selects a drive mode**

In the Maserati cars the driver can select among five drive modes: Sport_1, Sport_2, Comfort_1, Comfort_2, Offroad. When a drive mode is selected some parameters regarding the setup of the car are changed. In the cars equipped with the air spring optional, one of these parameters is the ride height. Each drive mode is associated to a specific ride height. The signal used to select the drive mode can be sent using the IO handler, from now on this signal will be called DMS, as Drive Mode Selector. This signal and RHS are independent; it can happen that the two signals target different ride heights. The ECU picks as desired level the one from the most recently received signal.

To test this functionality the following test case has been set:

- Vehicle longitudinal speed: null.
- Extra force on the vehicle: null
- RHS: NO_REQUEST.
- DMS signal: varying to achieve all possible drive modes
- Initial reservoir pressure: reservoir at nominal pressure for reservoir only levelling

This test goals are:

- Check that the ECU is able to receive the DMS signal and acknowledge it
- Check that the ECU addresses the correct level height for the correct drive mode.
- Check that during the whole levelling phase the ECU correctly sets its output signals

The capability of the ECU to reach the proper height level regardless the reservoir pressure has already been tested, so it is not among the goals of this test.

During this test the same signals as the previous test are monitored.

The target ride height values for each drive mode are:

- For drive modes Sport_1 or Sport_2: AERO_1
- For drive modes Comfort_1 or Comfort_2: NORMAL
- For drive mode Offroad: OFFROAD_1

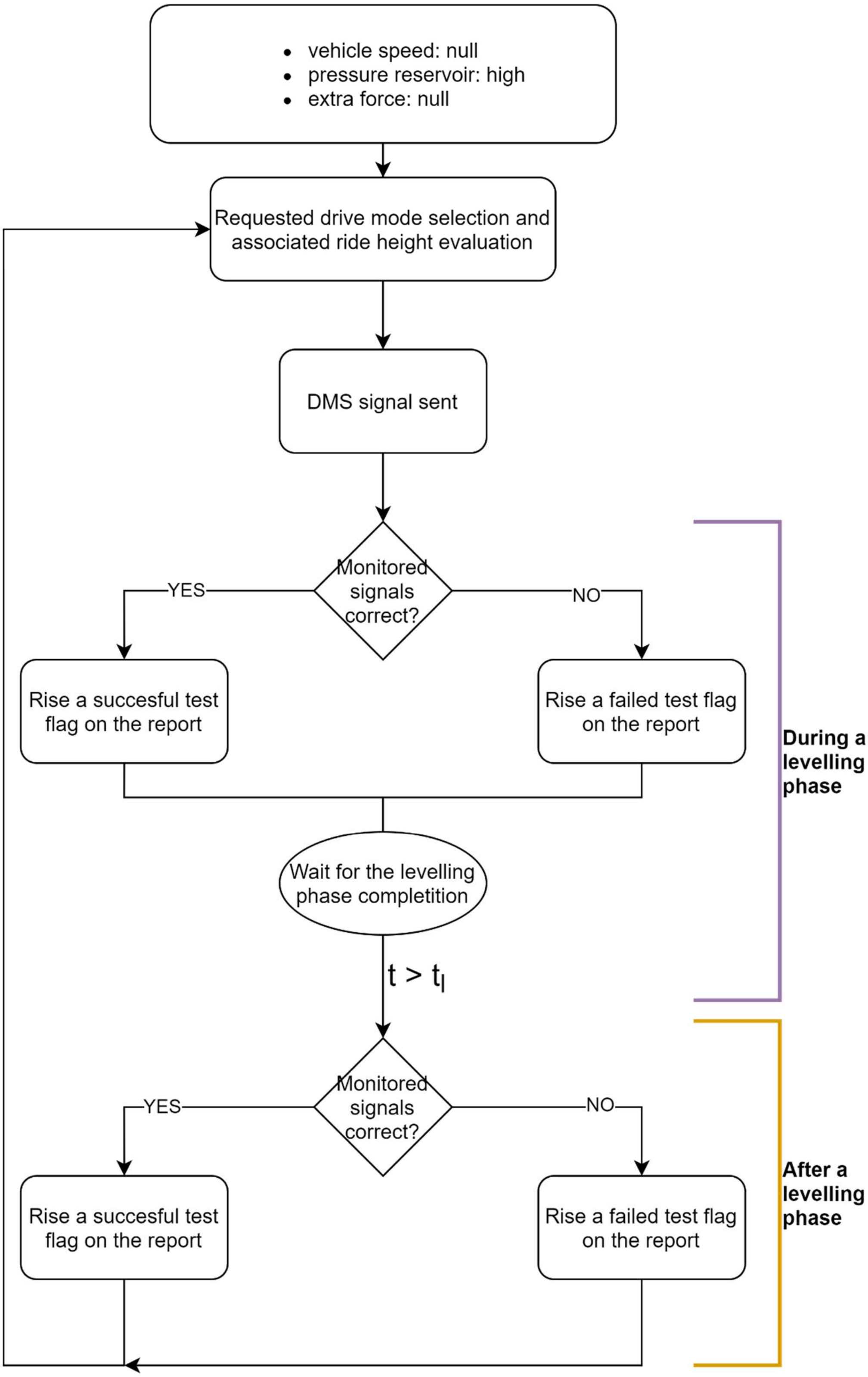


Figure 3.9 Detailed block scheme of the test used to check the behaviour of the ECU when the driver selects a drive mode

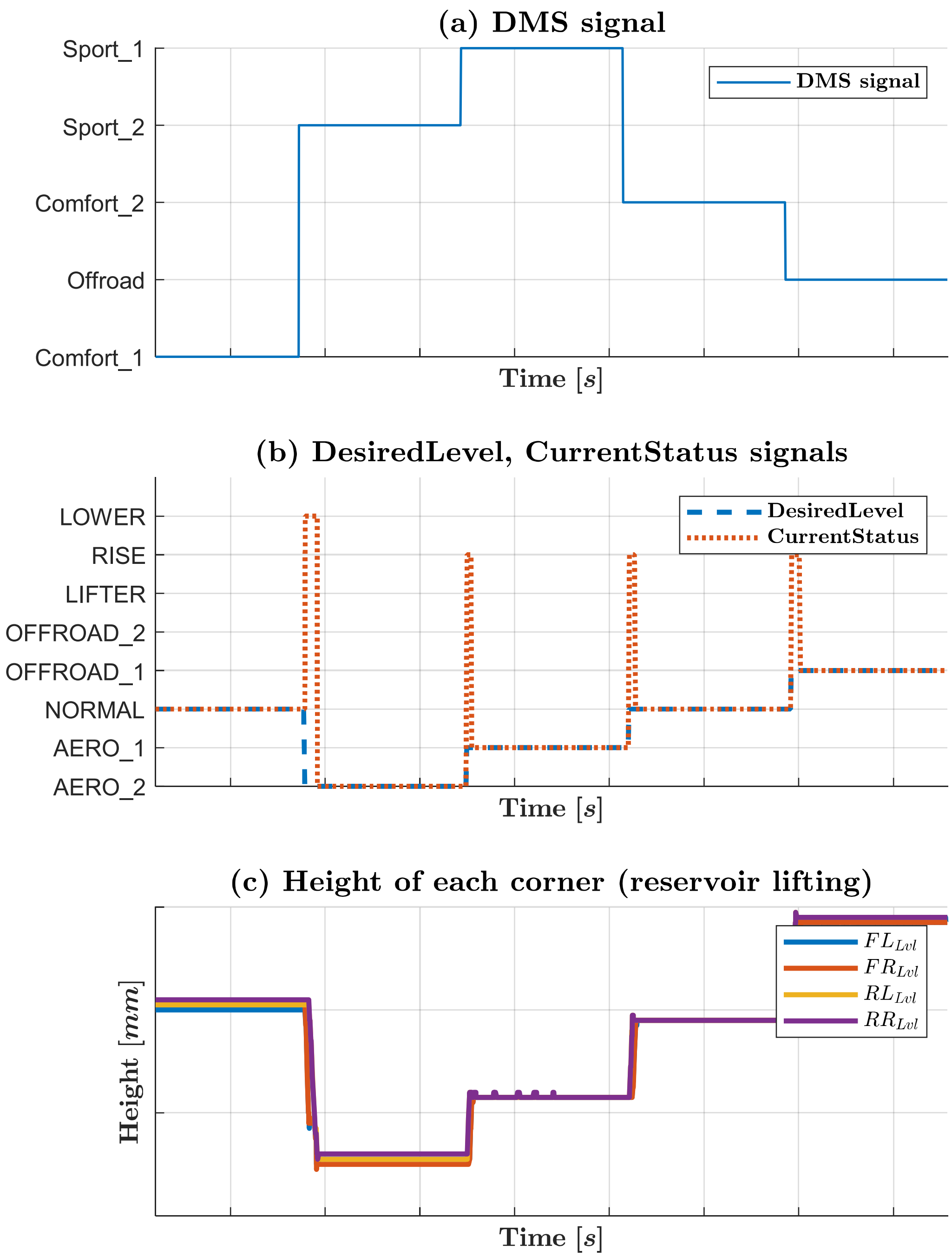


Figure 3.10 Boost levelling. (a) DMS signal, input to the ECU; (b) DesiredLevel and CurrentStatus output of the ECU. (c) Height of each corner of the vehicle, expressed in mm as difference to the NORMAL level (@ time t_0), which corresponds to 0mm.

Figure 3.10 shows the logged signals during the test. The ECU behaviour reflects the expected one, except one drive mode: Corsa. In this drive mode the ride height should be AERO_1 but the ECU targets AERO_2. This issue has been reported as another bug.

The currently reached level is: NORMAL iteration number: 0
The currently chosen level is: AERO_1 while the current level is NORMAL
The current drive mode is: Sport_2

4695 Check

Failed *Check*
Result of comparison between the Desired Level and the Selected Level
ASCM_DesiredLevel=AERO_2

4705 Check

Failed *Check*
Result of comparison between the Desired Level and the Selected Level
ASCM_DesiredLevel=AERO_2

The currently reached level is: NORMAL iteration number: 0

4706 Check

Failed *Check*
Result of comparison between the Final and the Selected Level
ASCM_Stat=AERO_2

The currently reached level is: AERO_2 iteration number: 1

4707 Check

Failed *Check*
Checking correct heightFL_Lvl=-29

Figure 3.11 Test report extract showing the Sport_1 bug. During Levelling Phase: Check 4695: during the levelling the desired level is not AERO_2 as expected. After Levelling phase: Check 4705: after the levelling the desired level is not AERO_2 as expected. Check 4705: monitoring the CurrentStatus signal emerges that the actual ride height value is AERO_2 instead of AERO_1. Check 4707: The ride height value expressed in mm is different than AERO_1.

Figure 3.11 shows an extract of the test report that highlights the “Sport_1 bug”. In top of the figure the currently chosen level refers to the expected target level when the Corsa drive mode is selected. From the performed checks emerges that the ECU is programmed to target the AERO_2 ride height value when Sport_2 drive mode is selected, since the DesiredLevel signal addresses the AERO_2 level. The levelling itself is successful, since the ECU reaches the targeted ride height, even if it is wrong.

3.2.3 **Automatic extra weight levelling**

The ECU is required to keep each corner's height in a specified interval around the required level. Thanks to specifically implemented algorithms, the ECU can detect an extra load weighting on the corners. To counteract the car being lower than the chosen ride height value, or one corner being lower than the others, the ECU lets more air flow inside the springs to reach again the nominal configuration.

To simulate an extra weight the input F_{extra} of the dynamical model will be used. The test will be launched for the car being in each ride level height.

To test this functionality the following test case has been set:

- Vehicle longitudinal speed: null.
- Extra force on the vehicle: fixed value, applied separately on each corner and evenly distributed
- RHS: varying to achieve all possible ride height values
- DMS signal: null
- Initial reservoir pressure: reservoir at nominal pressure for reservoir only levelling

This test goals are:

- Check that the ECU is able to detect an extra load on the car
- Check that the ECU corrects the ride height affected by the extra load
- Check that during the whole levelling phase the ECU correctly sets its output signals

During this test the same signals as the previous tests are monitored.

Figure 3.12 shows the detailed block scheme of the test case implementation. It is based on two nested loops: one for the weight application on each corner and the outer one to test the ECU behaviour in each ride height value.

Figure 3.13 shows the logged signals evolution during the whole test. The nested loop structure emerges confronting (a) and (b): in each ride height value the weight is applied on each corner and evenly distributed on all of them.

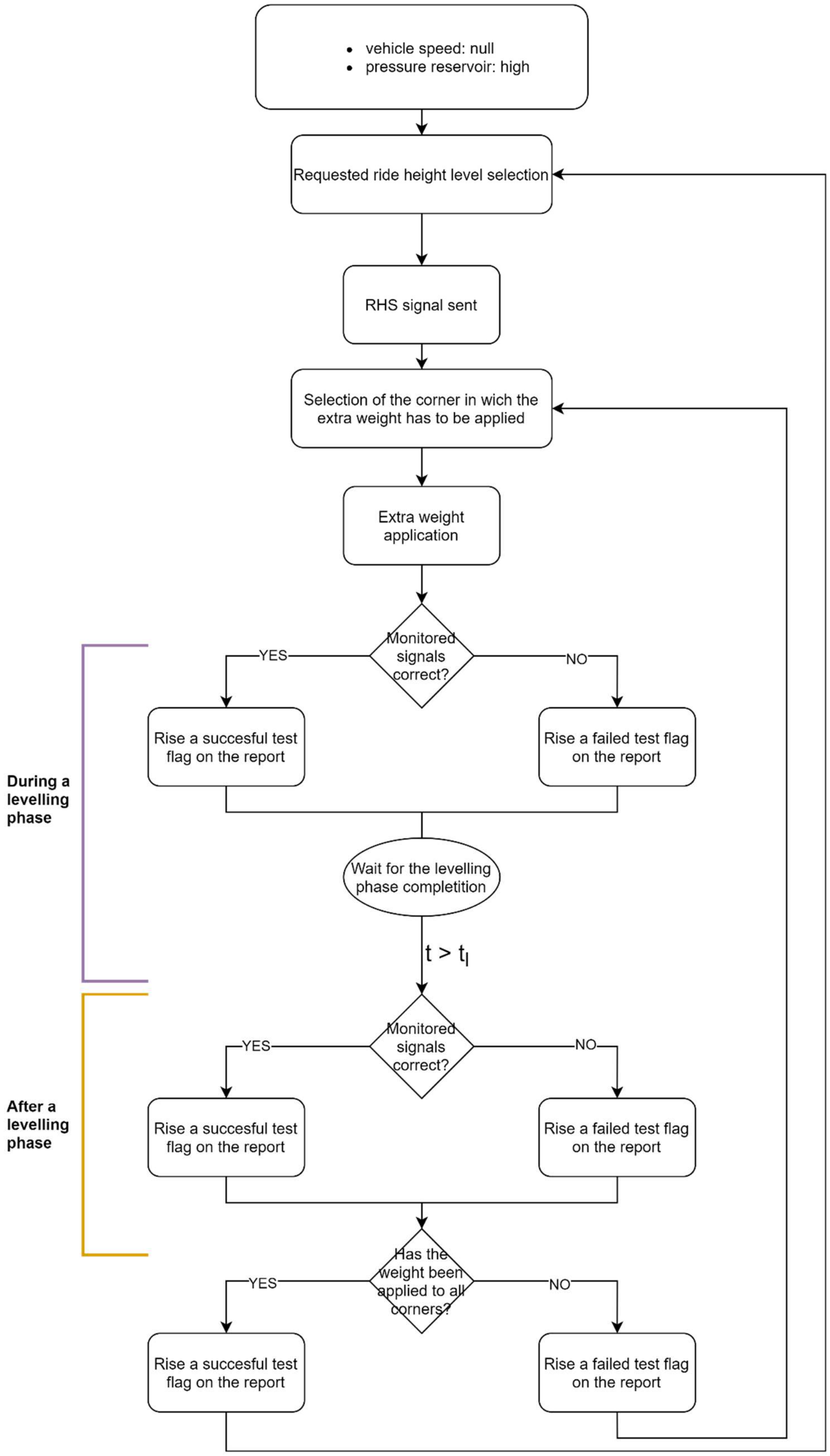


Figure 3.12 Detailed block scheme of the test used to check the behaviour of the ECU when an extra load is applied.

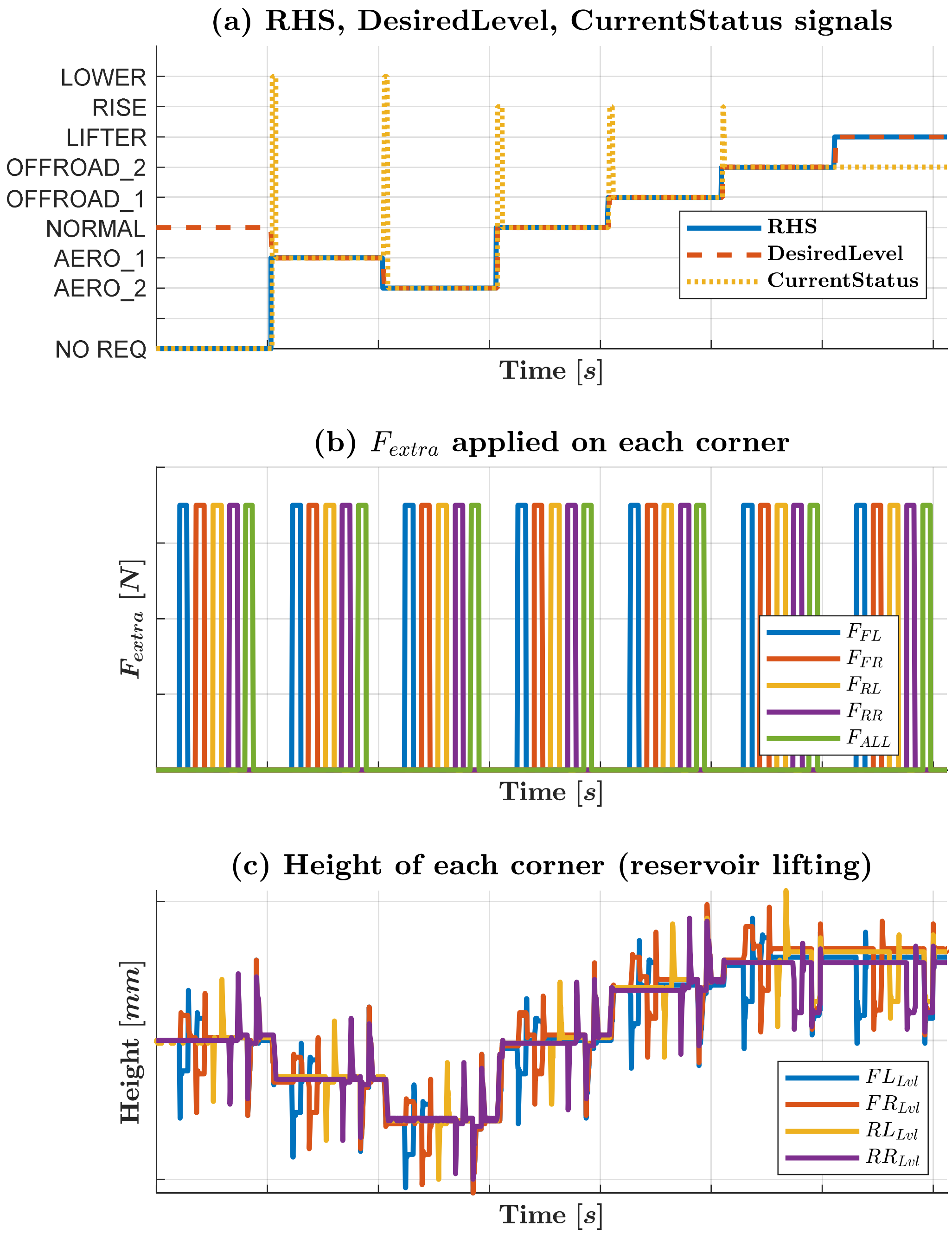


Figure 3.13 Log of the whole test. (a) RHS input to the ECU, DesiredLevel and CurrentStatus output from the ECU. (b) Extra weight applied to each corner. (c) Height on each corner, expressed in mm

To better understand the ECU behaviour, Figure 3.14 shows the weight application loop in only one ride height value.

Figure 3.14 (c) shows the evolution of the ride height in mm of each corner:

- At t_0 the desired height level is reached (OFFROAD_2).

- At t_1 the extra load is applied to the Front Left corner; the ECU immediately starts to level the corner to counteract the lowering due to the applied load. The two front wheels are levelled together, the goal is to make the average of the two heights equal to the desired height level.
- At t_2 the levelling is accomplished.
- At t_3 the load is deactivated. At this point the pressure inside the corners is higher than the nominal pressure, remembering $F_s = P_s * A_{eff}(s)$ this is necessary to counteract the extra load effect and balance the forces applied at the top of the spring. This higher force brings the car to rise when the extra load is deactivated, the ECU immediately counteracts this effect removing air from the spring chamber to lower the pressure inside. Note that this happens only for the Front Left corner, the one affected by the extra load
- At t_4 the load is applied to the Front Right corner and the same strategy is applied.
- At t_5 the weight is applied to the Rear Left corner, immediately the ECU starts levelling to counteract the load effect. The affected corner alone is levelled, to reach the desired height level.
- At t_6 the levelling is accomplished
- At t_7 the weight is applied to the Rear Right corner and the same strategy is applied.
- At t_8 the weight is applied to all the corners in the same amount. The ECU immediately starts to level the corners to make them reach the desired height level.
- At t_9 the levelling is accomplished.

Note that, in case of a load affecting one corner, two different strategies are applied to the two axes. When the weight is evenly distributed on all the wheels the levelling phase is divided in two phases: first the rear axle, then the front axle is levelled.

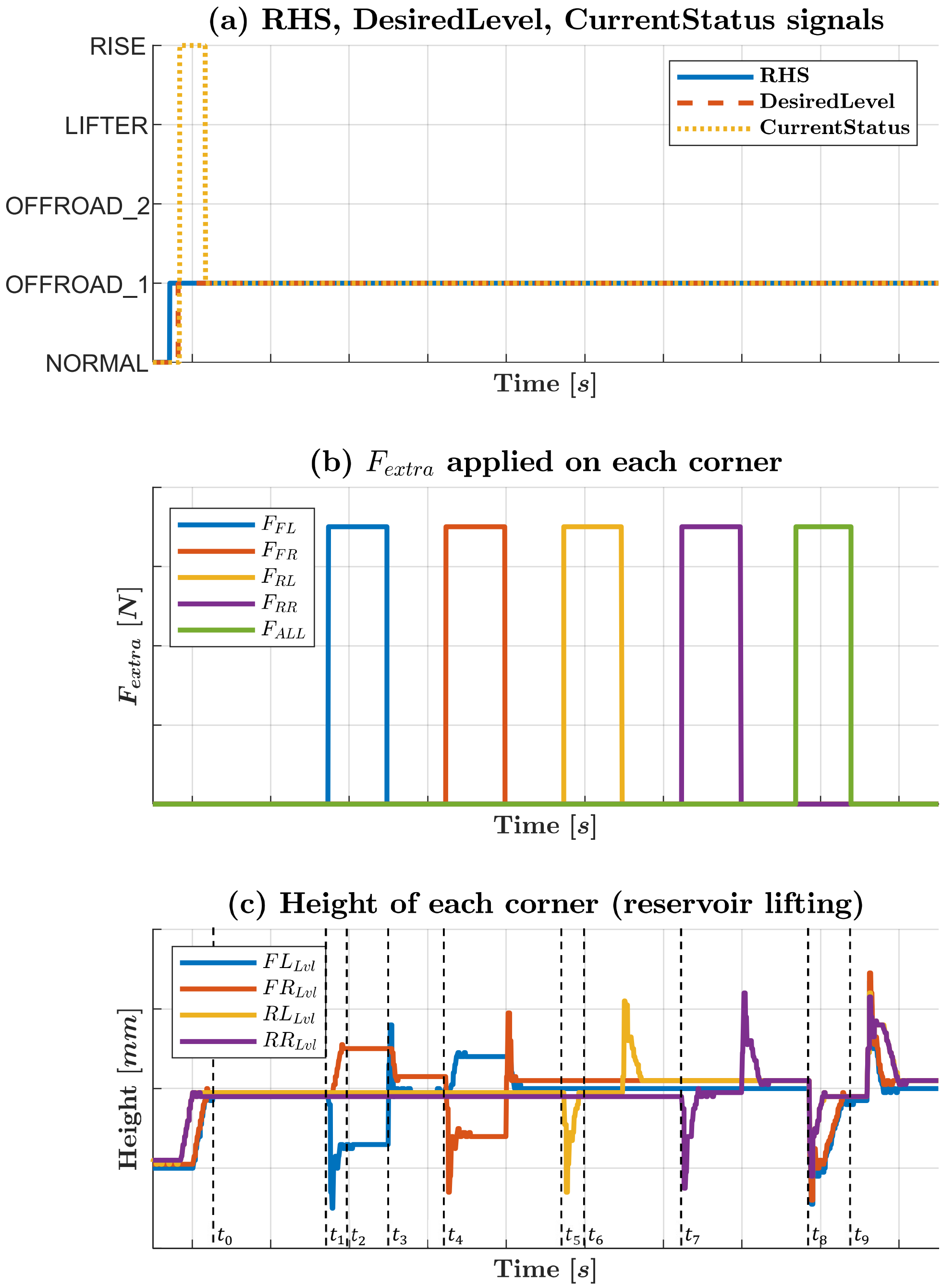


Figure 3.14 Log of one phase of the test. (a) RHS input to the ECU, DesiredLevel and CurrentStatus output from the ECU. (b) Extra weight applied to each corner. (c) Height on each corner, expressed in mm

3.2.4 **Automatic levelling according to longitudinal speed**

In the Maserati cars equipped with the air spring optional, the driver can manually select a specific ride height value, that the car is required to reach. The manual selection of the ride height value can be overrun by the ECU. In fact, when the vehicle speed crosses some thresholds, for aerodynamic efficiency purposes, the vehicle is lowered. Moreover, having a lower centre of gravity makes the car more responsive to high-speed manoeuvres.

To test this functionality the following test case has been set:

- Vehicle longitudinal speed: varying to cross all the pre-set thresholds.
- Extra force on the vehicle: null
- RHS signal: set to OFFROAD_2 at the beginning.
- Initial reservoir pressure: nominal pressure for reservoir levelling.

This test goals are:

- Check that the ECU is able to receive the vehicle speed signal and acknowledge it
- Check that the ECU, once the current vehicle speed is acknowledged, targets the correct ride height value.
- Check that during the whole levelling phase the ECU correctly sets its output signals

During this test the same signals as the previous tests are monitored.

Figure 3.15 shows the detailed block scheme of this test. Starting from OFFROAD_2 the speed has been increased in steps, at every step a threshold is crossed so the ECU is required to level the car.

In Figure 3.16 the test logs are plotted:

- At t_0 the vehicle is not moving longitudinally, and the selected ride height is OFFROAD_2
- At t_1 the longitudinal speed is brought above the V_HOR2 threshold, so the car is considered moving too fast for this ride height value and the ECU is required to lower the car a step down. The CurrentStatus signal is set to OR_FAIL to communicate that this height level cannot be kept.

- At t_2 the speed crosses the $V_H OR1$ so the ECU is required to level the car from OFFROAD_1 to NORMAL. The CurrentStatus signal is set to OR_FAIL to communicate that this height level cannot be kept.
- At t_3 the $V_H N$ threshold is crossed, the ECU is required to level the car from NORMAL to AERO_1
- At t_4 the $V_H A1$ thtreshold is crossed, the ECU is required to level the car from AERO_1 to AERO_2
- At t_5 the car moves slower than the $V_L A2$ threshold, the ECU is required to rise the car from AERO_2 to AERO_1
- At t_6 the speed decreases
- At t_7 the car is fully stopped, and the ECU rises it back to NORMAL level.

The ECU is not required to level the car back to the OFFROAD_2 height level.

Note that the ride height signal is noisy when the longitudinal speed is above zero km per hour. A white noise signal was added to the ride height signal because if the ECU, when the speed is not null, senses a too steady signal, assumes that the sensor is broken. This causes an error flag to be risen and the whole levelling function to be inhibited.

Note that in this case, since there is automatic levelling, the ECU sets the DesiredLevel signal ignoring the RHS signal.

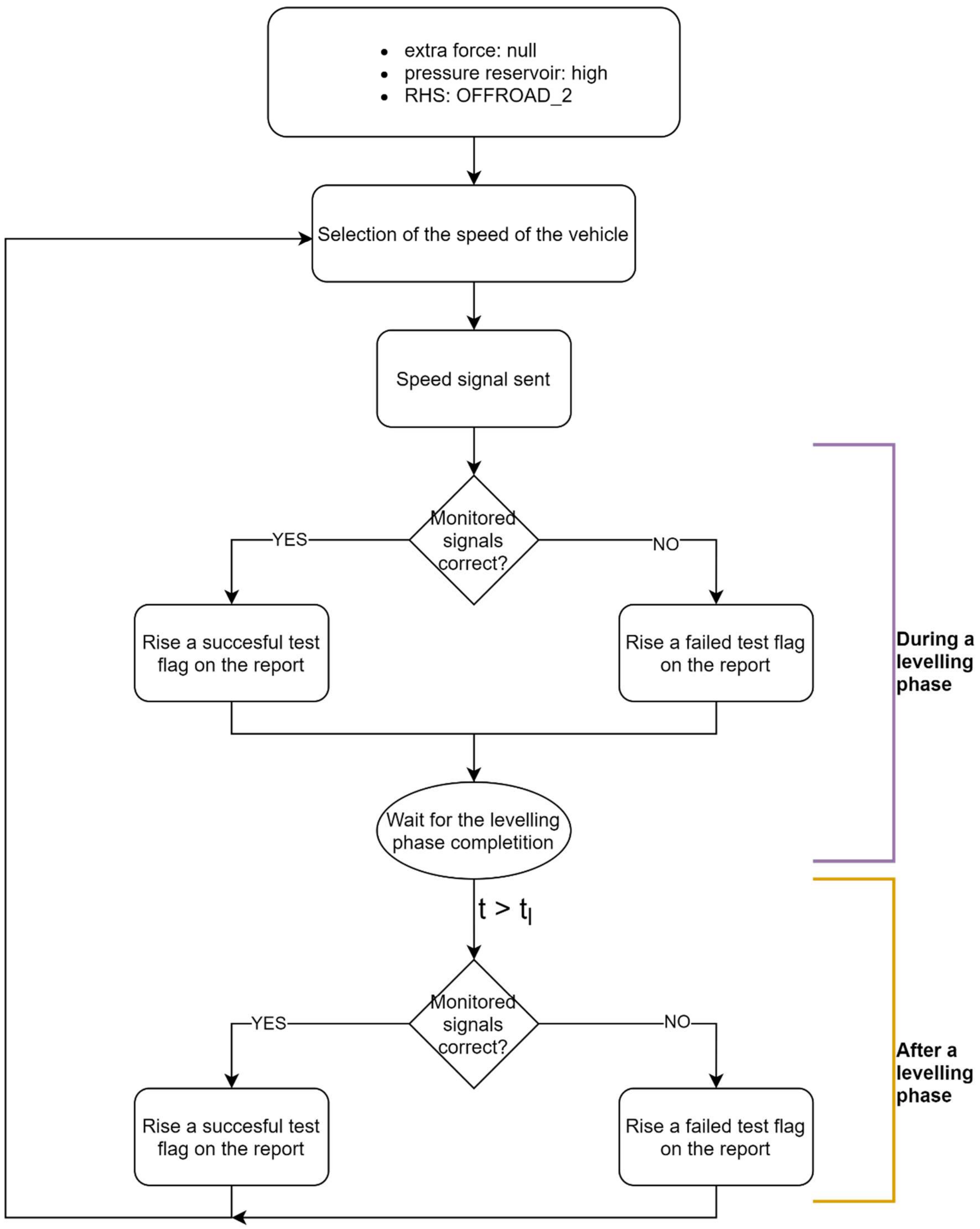


Figure 3.15 Detailed block scheme of the test used to check the behaviour of the ECU when the vehicle speed varies.

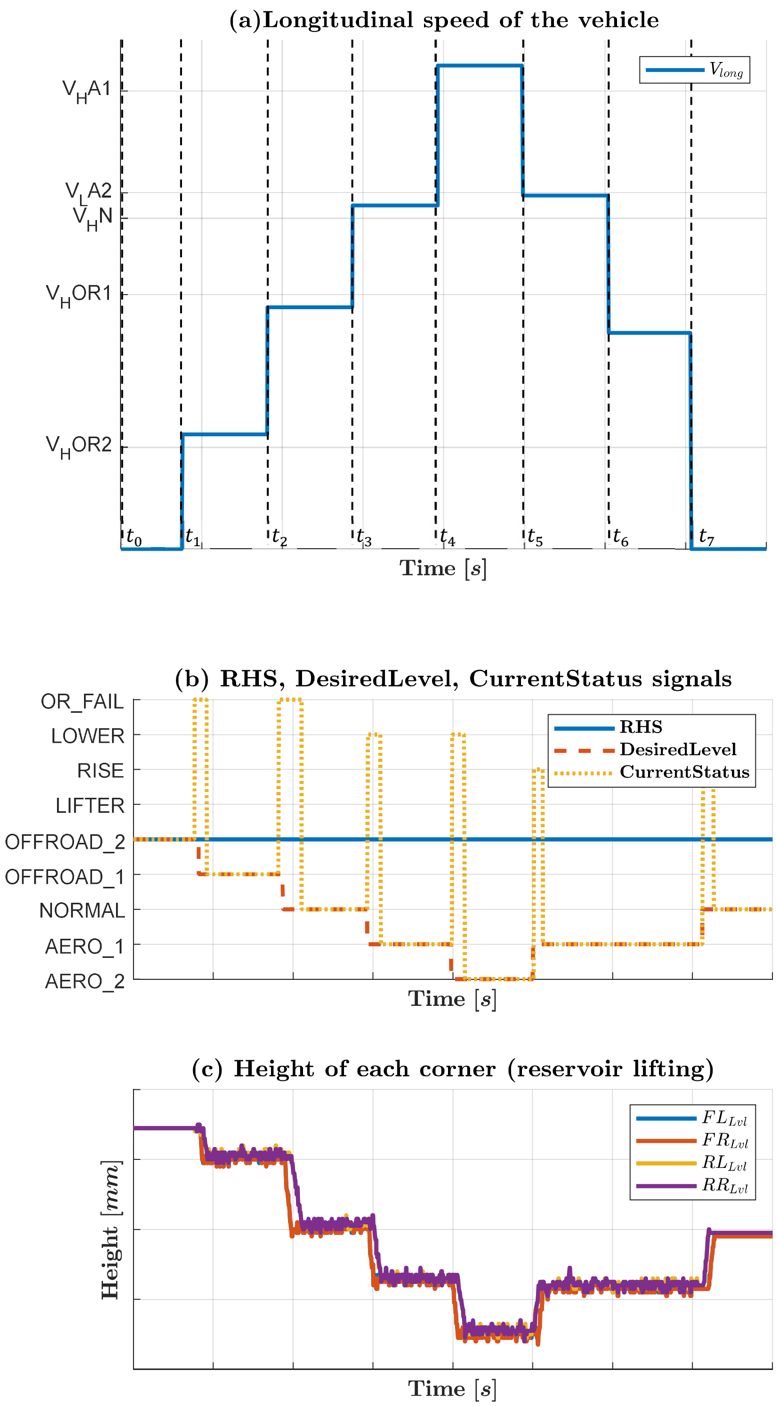


Figure 3.16 Log of the automatic levelling according to vehicle's speed. (a) Input pattern of vehicle speed. The speed crosses all the thresholds of interest. (b) Input: RHS constant to normal. Output: DesiredLevel computed by the ECU to accomplish the height requirements due to the speed. Output: CurrentStatus. (c) Output: Vehicle height in mm.

3.2.5 **Reservoir filling**

Among the available lifting modes, the fastest and most comfortable one is the reservoir only mode. This because the pressure drop among the reservoir and the spring leads to a higher airflow than in the other modes, moreover with the compressor turned off there is no noise due to the levelling.

When the reservoir pressure is below a threshold this mode cannot be used anymore, so the compressor must be turned on to fill the reservoir again. The ECU is required to handle this situation. The reservoir filling will start when the pressure inside the reservoir is low enough and the car is moving faster than V_{FILL}. This threshold is designed to be high enough so that the noise level inside the car would be not affected by the noise produced by the compressor turned on.

To test this functionality the following test case has been set:

- Vehicle longitudinal speed: varying to cross the pre-set threshold V_{FILL}.
- Extra force on the vehicle: null
- RHS signal: set to OFFROAD_2 at the beginning.
- Initial reservoir pressure: low.

This test goals are:

- Check that the ECU performs the reservoir filling procedure as expected.
- Check that during the whole levelling phase the ECU correctly sets its output signals

During this test, the monitored signals are the reservoir pressure and the compressor command.

Figure 3.17 shows the block scheme used to implement the test. The vehicle speed initially is set to zero, then it is risen to cross the V_{FILL} threshold. The signals are monitored during and after the filling phase

Figure 3.18 shows the test logs:

- At t_0 the car is not moving and the reservoir pressure is low.
- At t_1 the vehicle speed crosses the V_{FILL} threshold and the compressor is turned on

- At t_2 the filling procedure is complete

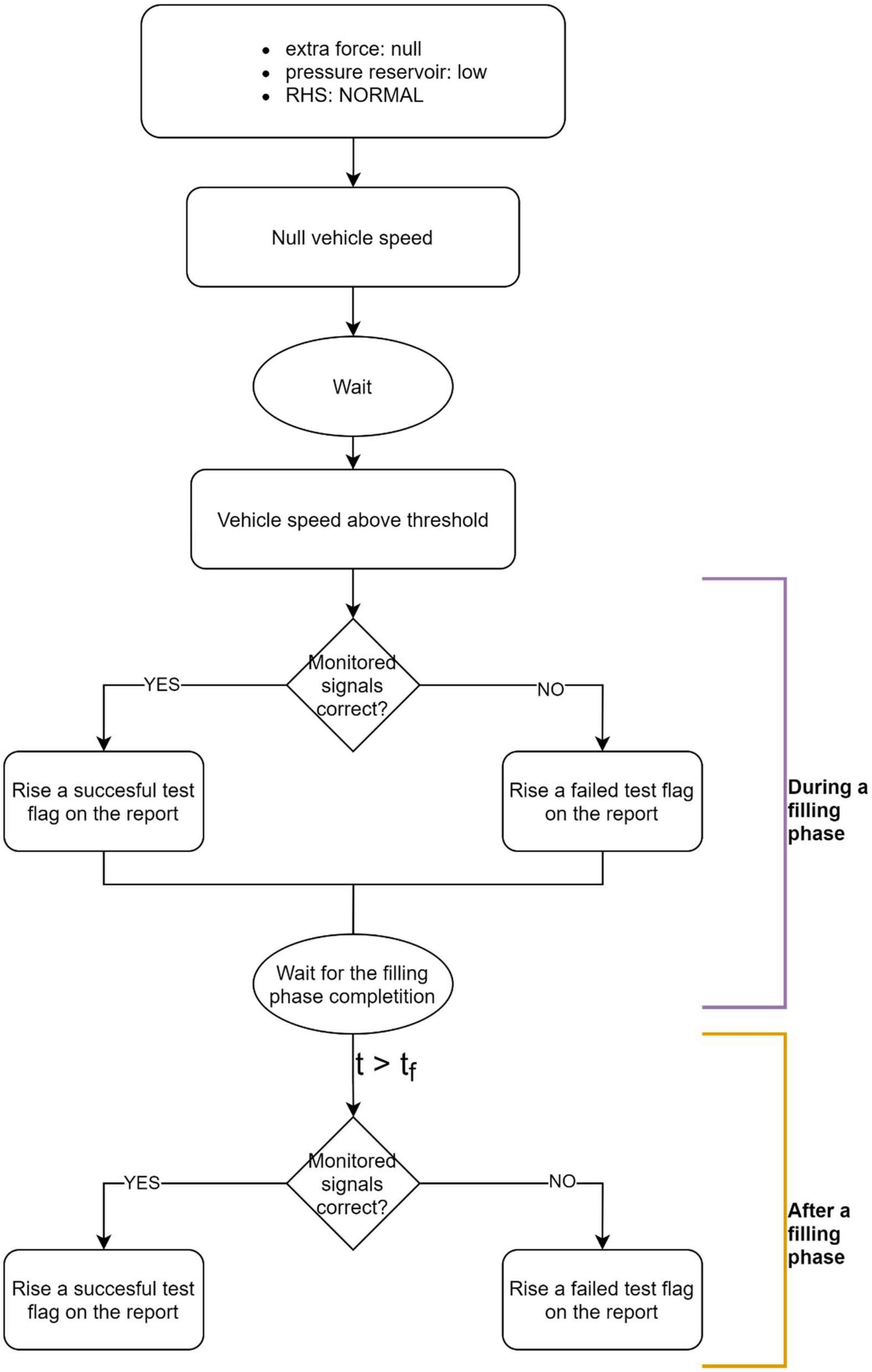


Figure 3.17 Detailed block scheme of the reservoir fill test case

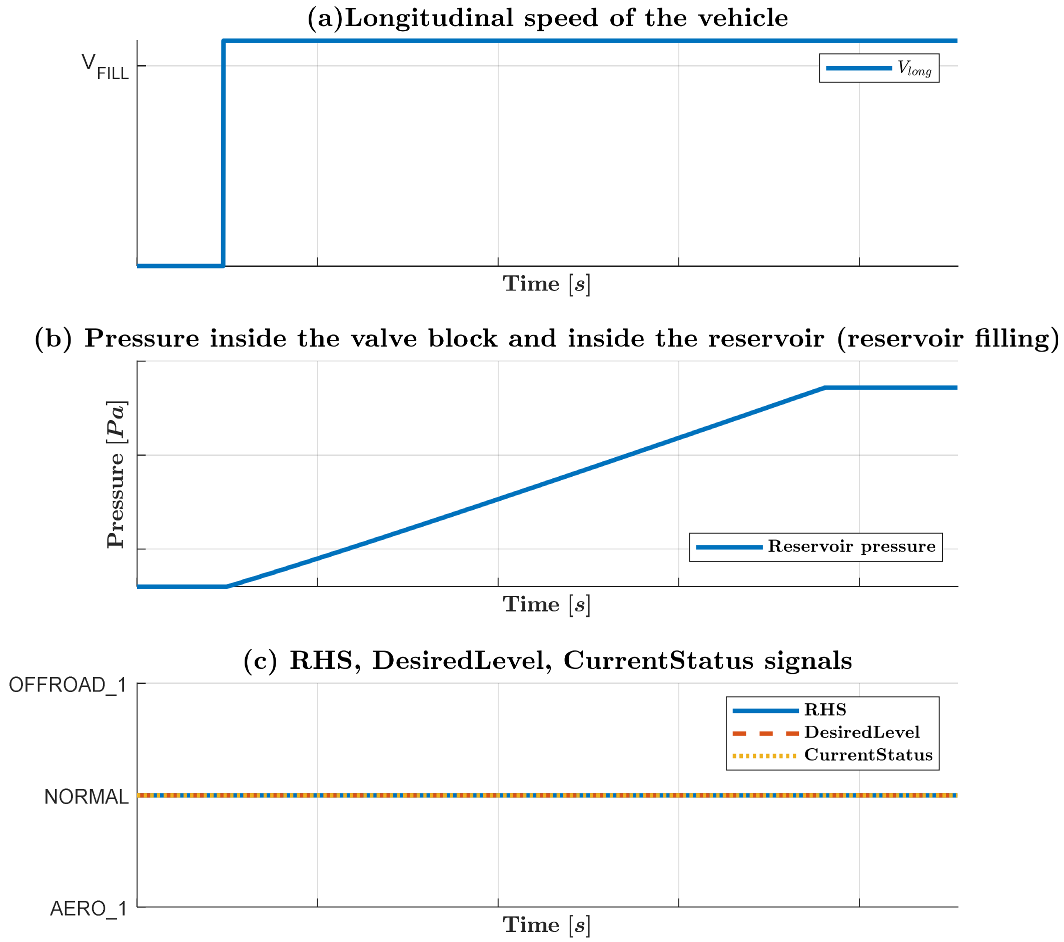


Figure 3.18 Logs of the compressor filling test case. (a) The longitudinal vehicle speed is brought from zero to higher than V_{FILL} (b)The reservoir and the valve block pressures, during the reservoir filling procedure. (c) the RHS, DesiredLevel and CurrentStatus signals, all set to normal.

4. Conclusions

Air springs are a complex system, they require a complex pneumatic circuit and a complex electronic support to be implemented. The design phase of this type of systems is longer if compared to traditional suspension systems. As described in this dissertation, also the testing and validation phases are longer.

Increasing the amount of time needed to develop and produce a component means increasing the production costs of the whole vehicle. The HIL testing technique is a very powerful tool to reduce these costs.

The HIL testing technique is very common nowadays and it is used by every car manufacturer [4]. It is very efficient, in some cases it is necessary, since it is not possible to test some systems in labs, due to safety reasons.

The HIL bench was successfully used to highlight some bugs of the system and, if needed, can be updated to be a useful tool to also test the future air spring systems in Maserati. The designed dynamical model was found useful for the target application, nonetheless can be furtherly improved starting from ad hoc data acquisitions thanks to which the internal parameters of the model can be finely tuned to obtain plots as close as possible to the real measured data.

A lot of manufacturers, as Maserati, are bringing the design and test phases as close as possible, also thanks to driving simulators. Future steps of the presented work are the integration of the mathematical model of the springs with the tool used in Maserati to model the whole vehicle dynamics. In this way the air springs can be simulated in the dynamic and static simulator, so that the test drivers can evaluate the springs performances at the earliest stages of the suspension design.

The abovementioned integration can be useful not only for human-driven simulators. The air spring and the dynamical vehicle models can run on a SCALEXIO machine to automatically simulate complex driving scenarios, so that very specific test cases can be implemented.

5. Bibliography

[1] Z. Sihong, W. Jiasheng and Z. Ying, “Research on Theoretical Calculation Model for Dynamic Stiffness of Air Spring with Auxiliary Chamber,” *IEEE Vehicle Power and Propulsion Conference (VPPC),* 2008.

[2] W. L. Wen, Y. Chen, S. Zhang, E. Mao, Y. Du and H. Wen, “Damping Characteristic Analysis and Experiment of Air Suspension with Auxiliary Chamber,” in *IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd.*, 2018.

[3] G. M. Coletta, “Sospensioni Pneumatiche per Autoveicoli Sviluppo di un Tool per il Dimensionamento di Molle ad Aria,” Università degli Studi di Napoli Federico II, 2019.

[4] dSpace, “Partial list of costumers,” dSpace, July 2021. [Online]. Available: https://www.dspace.com/en/inc/home/company/customers.

[5] H. Zhu, J. Yang and Y. Zhang, “Dual-chamber pneumatically interconnected suspension: Modeling and theoretical analysis,” *Mechanical Systems and Signal Processing,* no. 147, pp. 125-145, 2021.

[6] S. J. Lee, “DEVELOPMENT AND ANALYSIS OF AN AIR SPRING MODEL,” *International Journal of Automotive Technology,* vol. 11, no. 4, pp. 471-479, 2010.

[7] T. D. Gillespie, Fundamentals of vehicle dynamics, Warrendale, PA: Society of automotive engineers, 1992.

[8] W. F. Milliken and D. L. Milliken, Race car vehicle dynamics, Warrendale: Society of automotive engineers, 1995.

[9] C. Washington and S. Delgado, “Improve design efficiency and test capabilities with HIL Simulation,” in *2008 IEEE AUTOTESTCON*, 2008, pp. 593-594.