Toulouse 2009: HIL Validation of embedded software - Application on hybrid industrial vehicle.

Sylvain RENARD[1], Eric BIDEAUX[2], Xavier BRUN[2], Daniel THOMASSET[2] & Patrick LAGER[1]
[1] *Volvo Powertrain France, 99 route de Lyon, 69808 Saint-Priest, France*
Email : sylvain.renard@volvo.com

[2] *Ampère Laboratory, INSA Lyon, 25 avenue Jean Capelle, 69621 Villeurbanne Cedex, France*

Abstract

Current requirements of ecology and mobility are at the basis of the development of less polluting vehicles. In most of the cases, these vehicles need a clever and optimized energy management, which can be reached thanks to a more complex electronic embedded system. That is why, a special care on electronics reliability is required to ensure a good product quality. In this paper, we will focus on a systematic reasoning to obtain a suited test bench for covering the verification of a complete set of embedded electronics functional specification of a hybrid powertrain. The HIL (Hardware In the Loop) test bench enables the control unit environment in real time to be simulated and their closed loop behavior in wished situations to be studied. Based on control unit and system specifications, the method describes how to define the HIL platform in terms of hardware interface and internal behavior (models).

Keys-words*: Validation, ECU, Electronic architecture, hybrid industrial vehicle, HIL specifications.*

Introduction

New functionalities are today controlled by electronics. Upscale vehicles can have up to 70 ECU (Electronic Control Units) with five communication networks. They are used to control engine injection, to apply the braking strategies, to correct vehicle course, to regulate the vehicle speed, to manage the central locking of the doors… The beneficial effects of electronics are then widely admitted. In fact, it has become essential in order to improve vehicle performances and to meet the more and more strict environmental regulations. This increasing use of electronics needs to be very focused by manufacturers because it also increases the risk of failure. Theoretically, a suited reaction should have been anticipated for each failure, but with the very high number of embedded functions today, this analysis leads to a combinatorial explosion of the number of possible failures. Formal verification is then not yet applicable and testing methods are needed to ensure a relevant verification.

Our study aims to propose an answer to this problem of the electronic system verification in the case of the development of a new complex electronic architecture. It has been raised in order to develop an efficient testing tool, a HIL simulation platform, for the verification of the electronic architecture of the hybrid powertrain developed by the Volvo Group for its buses, trucks and refuse vehicles. This new technology involves new hardware for the control of the complete driveline and more complex software strategies, which are implemented in several embedded ECU, whose behavior has to be first checked aside at ECU level and then jointly at system level.

The systematic reasoning that we present in this paper wants to give a rigorous approach to develop any HIL platform for verification purpose. It is based on the specifications of an electronic system to test.

In the context of hybrid software system, we will show with one example how the implementation of this method has lead to the development of well sized models. Then the result of a representative ECU validation will be presented. Based on this result, we will finally conclude with an evaluation of the efficiency of this method.

1 – Systematic reasoning

Among the verification tools, the HIL platform appears as the most powerful to cover the complete specification booklet verification in a safe way, with reproducible behavior, in a limited time and with the best financial efficiency **[Brunet 05]**, **[Morita 08]**. However, the challenge is to be able to identify the required phenomenon to simulate in order to design the platform in a sufficient way. Therefore, we propose to adopt an approach, based on the study of each requirement to verify, leading to the definition of the perimeter and the need in terms of model. First, we will give an overview of what is the HIL platform, and then we will present the proposed approach.

HIL platform presentation

The main function of the HIL platform is easy to understand: the ECU or the system to test is connected to the HIL platform in the same way as it is on its actual environment. Then it is stimulated with external actions similar to the ones usually done on the system. For example, if the system to test is an ECU used in a vehicle, the stimuli would be the actual actions done by a driver: an acceleration demand, a braking demand, a retarder demand, a gear shift demand, etc.

The **Figure 1** shows how the system to test is connected to the platform, in this example, it is an ECU. A specific harness needs to be built in order to connect each input/output affected by the tests. Then the HIL platform should be able to understand each connection and to close the loop between the outputs of the system to test and its inputs. It means that the actions requested by the system to test through its power modulators (outputs) should be visible on the sensors of this system (inputs). This can be realized thanks to behavior model developed into the HIL platform. Then the user can access to any variables or parameters of the model through HMI (Human Machine Interface) to stimulate a wished action.

The main challenge of this installation is to fulfil the real-time constraint and to get a sufficient level a modelling to cover the complete set of tests to perform. Indeed, a good equilibrium between accuracy and real-time operation of the model needs to be found. Some phenomenon might be neglected if not essential in order not to overload the HIL platform processor.

But first of all, it is necessary to define what should be able to do the platform. In this aim, we have used an approach based on the specifications of the system to verify in order to design the platform in a suited way.

Method presentation

The method is based on the study of each functional requirement to verify. Each one is then identified to a test to perform. Then the tests are used to specify the HIL platform from a hardware point of view, with the definition of the inputs and outputs interface, and from a software point of view, with the definition of the modelling environment. In order to help the designer of the platform, we have written a test typology; each type of test is defined by a similar model need. Doing the identification of each requirement to a test of the typology, the complete software and hardware needs of the HIL platform can be obtained more easily. That is why, we will first define the test typology and then the proposed approach.

Test typology

Two kinds of tests have been defined: the low level and the high-level tests. Low level tests include the direct verification of the inputs and outputs. The operation of the supply, the sensors, the power modulators and the communication networks are verified. High level tests are focused on functional verification. They are sorted according to their verification criteria, for example, are we interested only on the final state of the function or on the dynamic behavior to get this state? The main difference between high level and low-level tests is the scope of modeling. The low-level tests are mainly focused on interfaces in open loop and then to limited functions, for example is the received data from a sensor well interpreted by a control unit? The high-level tests usually require physical phenomenon modeling. They need to be preceded by low level tests on the affected inputs and outputs. The following table summarizes this topology (**Table 1**).

Approach description

The developed method shows with a systematic reasoning how to verify functionalities at a control unit level or system level. The purpose is to build a HIL platform well sized to answer in a sufficient way to the verification need. We have chosen to follow a V-cycle, which is a well-known structure in the software development (**Figure 2**).

1. First, all the requirements of the system to test are extracted from specifications. Then for each one, the high-level tests to perform are identified. From these tests, the physical phenomenon to take into

account can be got and, according to the test typology, the software need, scope and type of the models, can be deduced.

2. Then, from the scope of identified high level tests to perform, the low-level tests are now selected. The hardware need and interface with the control units can thus be specified. Some additional low-level models can be required at this step.

3. The above formulated hardware and software needs are sufficient to build a suited HIL platform. The third step is then the development of this platform.

4. As soon as the platform is operational, tests can begin. First, the low-level tests are done. If one of them fails, the requirements linked to this test should be identified and it will be useless to perform related high-level tests.

5. For the other high-level tests, they can be performed and finally it will be possible to end the verification and to analyze the result.

From this method, it is important to remember some key points:
- Each requirement should be linked to a test from the typology
- Each type of low-level test defines a hardware interface to develop between the HIL and the system to test.
- The developed HIL platform is able to cover all the specifications
- This method is generic and can be applied to any HIL development
- The V-cycle process is suitable to update a HIL platform at any time.

This method has been applied to develop the HIL platform for the verification of the electronic system related to the hybrid powertrain. Because the complete specifications booklet represents a large amount of requirements and in order to illustrate the method in a relevant way, we have selected an apposite example: the gearshift.

2 – Application on electronic system of hybrid powertrain

System presentation

Volvo Powertrain has developed a parallel hybrid powertrain for medium duty trucks and buses. An electric motor, supplied by a 600V battery pack, is added to the conventional driveline between the clutch and the robotized gearbox (**Figure 3**).

The associated electronic architecture is then more complex than in a conventional vehicle since it needs the control of new hardware parts. Previously, two control units were used to control the powertrain: the Engine ECU, EECU, was in charge of the engine control taking information from engine sensors (temperature, air flow ????, pressure) and acting on the engine power modulator (injectors, waste gate, variable geometry turbo, exhaust gas recirculation???, exhaust brake…) and the Transmission ECU, TECU, in charge of the robotized pneumatic control of the clutch and the gearbox.

For the hybrid powertrain, 8 control units have been added to the EECU and the TECU.
- 4 of them are dedicated to the cooling system. Indeed, two cooling circuits have been designed, one for the electrical machine and the power electronics and the second one for the battery. Each circuit consists of a Pump ECU, PUCU (Pump Unit Control Unit), and a Fan ECU, HFCU (Hybrid Fan Control Unit).
- Then three specific control units are linked to the new electric components in order to diagnose them and control them if necessary. The MCU (Machine Control Unit) controls the DC/AC converter with a closed loop on the electric machine. The DCU (DC/DC Control Unit) handle the DC/DC converter between the 600V and the 24V network. The BMU (Battery Management Unit) collects information from the battery and controls its main operations.
- Finally, the HPCU (Hybrid Powertrain Control Unit), a new hybrid specific control unit, acting as a master ECU, manages the complete cooling system and analyzes the state of the seven new control units aforesaid

This analysis shows that with the hybrid technology, ten control units are necessary to ensure the control of the powertrain instead of two in the conventional case.

In summary, we observe that the electronic architecture is more complex, that a large number of interactions exist between the control units and that several functionalities are realized with a combination of several control units. Therefore, a new level appears in the scope of the powertrain control: the system level. In terms of reliability, it requires then also a new level of verification and validation.

The method that we want to present in this paper has been developed in order to build a HIL platform to ensure the verification of the electronic system conformity, in relation to its specification, at a control unit level and a system level, namely the 10 control units connected together.

Application of the method

The gearshift is a system level functionality requiring the direct interaction between three control units: EECU, TECU and MCU. Moreover, the gearshift operation is not simple since the dynamic behavior is very important. Therefore, from three requirements on the gearshift, we will identify the high-level test to perform. Then we will define the scope of the environment for each requirement. Finally, we will be able to deduce the low-level tests and the hardware interface to build.

Requirements selected to illustrate the method:

- The gear shift can happen only when the input torque is null
- Synchronization of the input gear shaft with the output gear shaft should be performed by a TECU torque demand on the electrical machine in a limited time.
- The engine speed should be synchronized with the input gear shaft before the clutch closure except if the engine does not have any self-retardation system in the case of an up shift. Then, the clutch will be used.

First requirement - The gearshift can happen only when the input torque is null -

This requirement can be verified with a "II-A-3" test from the typology in order to check a state sequence. Indeed, the input torque should be first equal to zero before engaging a new gear.

The input torque is equal to the sum of the torques delivered by the engine and the electric motor. Each one is estimated by the related control unit, respectively, EECU and MCU, and then transmitted to the TECU via the CAN network. The TECU will then be able to begin the gearshift as soon as the torque data will be equal to zero.

From a model point of view, a representative dynamic behavior is not needed. However, many dependencies between inputs and outputs can interfere and should be also taken into account:

1. For example, the electric motor control and the choice of the fuel injection angle depend on the same rotational speed of the engine axle (clutch closed).

2. The electric motor estimates its torque according to the rotational speed and the phase currents. From that, it deduces the control sequence of each IGBT (Insulated Gate Bipolar Transistor) of the power electronic converter. This affects the period and the voltage of the output alternative power link of each branch of the converter. Then they are used to supply the stator of the electric machine whose electromagnetic flux makes rotating the rotor. The electric machine and the converter require then a quite good level of modeling.

3. The engine is equipped with a common rail injection system upon which the injectors of each cylinder are connected. The rail is then supplied with fuel at a wished pressure. The fuel is injected in the cylinder with high accuracy timing depending on pressure, temperature, air flow and rotational speed sensors information. To get better performances, the air flow can be regulated with a Waste Gate or a VGT. During braking phases, a brake on the exhaust gas can be applied to enhance the retardation effect of the engine. To be able to estimate a good torque value, the EECU will need most of the above data, which require a good level of modeling.

This first requirement needs a model able to react to the control of the IGBT, the control of the injection, the control of the main engine actuators and the control of the gearshift forks. In parallel, the model should simulate number of essential variables and ensure their consistency. The **Figure 4** illustrates the required modeling environment for testing the first requirement.

Second requirement - Synchronization of the input gear shaft with the output gear shaft should be performed by a TECU torque demand on the electrical machine in a limited time -

Two kinds of test of the typology can be used to verify this requirement: the type II-A-2, which aims to check a dynamic response, and the type II-B-1, which aims to check the boundaries behavior. Indeed, the input shaft speed is synchronized by the electric motor control. The MCU uses thus the CAN information sent by the TECU. It is a dynamic regulation since the input shaft speed is dependant from the vehicle speed. That is why, it will be required to check that the gearshift operation does not exceed the limited time whatever the environment configurations. In these tests, the clutch is open and the gearbox is at neutral.

In terms of modeling needs, the vehicle dynamics will have to be taken into account as a simple variable load for example. The input shaft load has to be representative too (Inertia of electric motor, input shaft itself,

a part of the clutch, the frictions…). As previously shown, the electric motor control depends on rotational speed and currents. However, as it is a dynamic test, a dynamic response will be more suited than a mapping. Consequently, the speed control should be time representative. The test environment of this requirement is shown on **Figure 5**.

Third requirement - The engine speed should be synchronized with the input gear shaft before the clutch closure except if the engine does not have any self-retardation system in the case of an up shift. Then, the clutch will be used -

For this requirement we have chosen a II-A-2 test, which aims to check a dynamic response. Indeed, the engine speed will have to be synchronized with the input shaft speed. This action is controlled by the TECU which sends a CAN message to the EECU. In case of a downshift, the engine will be accelerated and in case of an upshift, it will be retarded. If possible, this should occur before closing the clutch. Whatever the case, the gearbox is engaged.

- To accelerate the engine speed, a torque will be applied by fuel injection. Thus, the TECU sends a speed request to the EECU.
- To retard the engine speed, there are two cases: either the engine is equipped of a self retardation system, or the clutch is used. The EECU should then send information on its ability to retard the engine. According to that, the TECU will choose the retardation strategy to apply.

The modelling environment should be able to react to the control of the injection, the control of the main engine actuators, the control of the self retardation system and the control of the clutch displacement.

As we have explained previously, the engine modelling requires the air loop to be modelled and the energy transmission from the fuel to the crankshaft too. The retarding action should lead to an overpressure in the exhaust manifold and then decrease the "pumping effect" of the engine.

The engine shaft loads need to be taken into account too (inertia and frictions) in order to have a consistent dynamic behavior during open clutch operation.

A model of the clutch is also required. According to the clutch position and the speed difference between the engine shaft and the input gear shaft, a certain torque will be able to be transmitted. The load behind the clutch (gearbox, axles, wheels and vehicle) needs to be also modelled to ensure a correct dynamic behavior. Finally, the engine synchronization duration will be representative if the dynamic of the mechanical system (especially the clutch) and the air loop are close to the real system.

The test environment of this requirement is shown on **Figure 6**.

Complete required modelling

The analysis of the three identified environments for the verification of each requirement shows that theyare quite close. Very often a modeled phenomenon for a requirement is also used in another one. That is why, only one complete environment (cf. **Figure 7**), including the three previous ones, can be used to verify the gearshift functionality. It defines the software specification of the HIL platform. The modeling level is defined by the most demanding tests. In the studied case, it is needed to get a representative duration of the gearshift, the dynamic behavior needs then to be considered.

Low level tests identification

The low-level tests can be deduced from the **Figure 7**. Each input and output of the system to test can be identified to a test from the typology (cf. **Table 1**). This will define the hardware specification of the HIL platform.

In this environment, all the sensors are continuous: engine speed, air flow, pressures, temperatures, clutch position, forks position, input gear speed, output gear speed, machine speed, currents. Excitation tests (test I-5) can be used to verify them. The variables, read by sensors, need to be transformed in electric signals and then linked to the correct input of the system to test. The knowledge of each sensor characteristics is needed. According to the sensor, the electric signal will be different. The HIL platform should be able to generate the expected signals by the electronic device to test.

For the power modulators, we can identify four of them as on/off power modulator (clutch cylinder, fork cylinder, injectors, IGBT) and two of them as continuous power modulator (waste gate and self retardation system). Diagnostic test (type I-6 and type I-7) can be used to verify them. The platform will need to be able to analyze the electric signals delivered by the control unit and to link them to the software part of the platform: the modelling.

To verify the CAN communication between the control units a network diagnostic test can be used (test I-3). A CAN interface will be needed on the HIL platform to analyze these CAN exchanges.

Widespread use

Finally, with this example, the aim was to show how to define a suited HIL platform to verify the requirements of a quite complex functionality. In this paper, we have consciously reduced the number of requirements in order to be as clear as possible but the method would have been the same.

In our work, we have applied the two first steps of the method (**Figure 2**) on the complete specifications of the electronic system of the hybrid Powertrain. We have been able to get first a list of phenomena to take into account and the required degree of modelling, and then a list of the inputs/outputs to connect to the HIL platform. With this information, the platform realization could be planned.

Platform realization

This paper is focused on the HIL platform specifications. However, without dealing with all the challenges of the realization, we will give an overview of the reasoning we had and the platform we have developed. This step is the third one of the methods (cf.) **[Renard 07]**.

Although we had clear specifications of the HIL platform, it has not always been possible to build an exactly equivalent model. Several reasons can explain this; for example, the lack of input data or the real-time simulation constraints. In these situations, it was needed to define new targets in order to develop representative and feasible models. We have based our modelling work on the bond graph theory, very convenient for multi-domain modelling **[Karnopp 00]**. It is also a powerful tool in order to understand easily the phenomenon taken into account, to detect inconsistency or to analyse the causality.

After the modelling step, the model validation should occur based on their comparison with real data **[Brunel 06]**. We have defined a stationary and dynamic validation for each developed model. Then they can be integrated in the platform and validated in closed loop with the control unit. For the hybrid powertrain, ten control units were needed to be added to the platform. As a status of our work, we have developed the complete needed environment to support the verification of this electronic system; however, some of them are not yet available on the platform and will be integrated in the next months. This status shows that we are able to perform verification at control unit level of all the integrated control units and already at reduced system level between these control units.

3 – Result and discussion

Since more than one year, verification of the electronic architecture for a hybrid Powertrain has been performed complementing physical tests. Each set of verification consists of two parts: the control unit level and the system level verification. It takes place in a strong industrial context with a strict planning. That is why the HIL platform is very helpful to accelerate the verification process in order to meet the deadlines. We will take as example the EECU verification and then will discuss about the complete software enhancement that we observed by using this tool and this process.

EECU verification

Four verification sessions have been illustrated in the following bar graph (**Figure 8**).

They occur each 3 months, which matches with the development of a new set of functions and error corrections from the previous session. The test description is written according to the requirement to verify before performing the test. Then the test session lasts normally two weeks but depends on the number of new requirements to take into account. For each set, three data are illustrated: the number of covered requirements by the test description, the number of tests really done during the session and the number of successful tests.

During the first session, about eighty requirements were covered by the test description. But only 50% have been verified. This can be explained by the fact that the requirements were new and sometimes quite rough, moreover it was difficult to write test description on a completely new system, and finally, the first iteration were also used to correct the test description according to the system to test. During the following sessions, we have been able to increase this figure. The decrease of the succeeded tests that we observe for the fourth session can be totally explained by an important increase of the number of requirements from 95 to 165. In order to get back a correct quality of the software, some further steps of verification are expected.

Software quality enhancement

As we saw for the EECU, we have been able to evaluate the quality level of the software with the help of

the verification statistics. This is a very important point since it gives a way to qualify the quality. Nevertheless, it is important to remember that the tests cases are based on the requirements and if the requirements are not representative of the system, the statistics won't be relevant.

Thanks to the use of HIL platform and its strict realization, we are able to verify 100% of the requirements at control unit level. It would not have been possible to reach this figure with the traditional ways of verification (on vehicles or in test cells).

The test description can also be improved with a good robustness analysis of each requirement in order to add relevant tests. Then applying the introduced method, we can get a HIL platform able to simulate the tight environment to perform these tests.

It is interesting to know that for the four verification sessions, we found around hundred software problems at control unit level, which have then been corrected by the software development. These figures are very promising in order to continue the complete integration work to reach a full hybrid electronic architecture test support. It is also a good example for other Volvo projects in order to develop as soon as possible the HIL platform.

Conclusion

With the new hybrid technology, the electronic architecture of the powertrain has become more complex in order to control the new electric components, providing further features on the commercial vehicles. Ten control units are then involved in the propulsion function instead of two previously. This has led to a new level of verification: the system level in addition to the control unit level. The suggested method enables to get an adapted verification tool, a HIL platform, able to cover the complete specifications of a system at several levels. The method is based on the requirements analysis in order to write the specifications of the HIL platform. The results show that the developed tool is efficient and answers to its specifications.
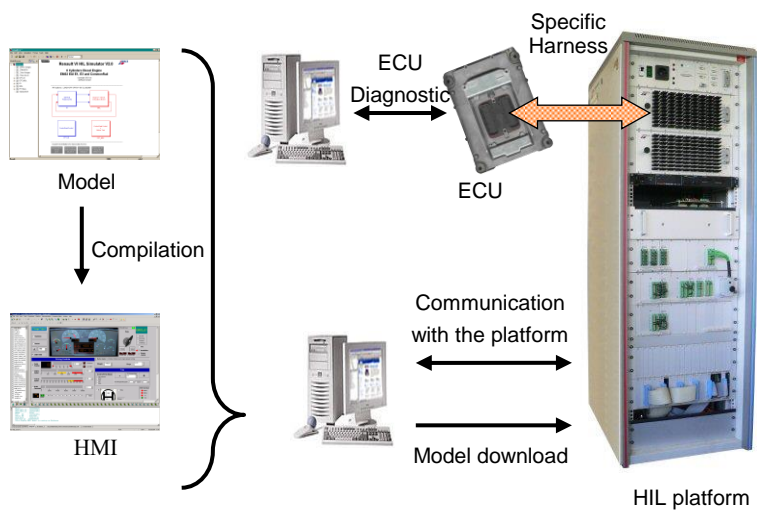
References

[Brunel 06]   D. Brunel, L. Flambard, C. Lompre, A. Sarrazin, *Hardware in the loop (HIL) tools development and use process for testing and validating car and trucks electronic control units (ECU)*, OTEKON'06, Bursa, Turkey, June 26-28, 2006.

[Brunet 05]   J. Brunet, L. Flambard, A. Yazman, *A hardware in the loop (HIL) model development and implementation methodology and support tools for testing and validating car engine electronic control unit (ECU)*, TCN CAE 2005, Lecce, Italy, October 5-8, 2005.

[Karnopp 00]   D. Karnopp, D.L. Margolis, R.C.Rosenberg, *System Dynamics: Modeling and Simulation of Mechatronic Systems*, 3rd edition, New York, USA, John Wiley & Sons, 507p, 2000, ISBN 0-4713-3301-8.

[Lumpkin 06]   E. R. Lumpkin, M. Gabrick, *Hardware/Software Design and Development Process*, 2006-01-0170, SAE World Congress, Detroit, USA, April 3-6, 2006, 2006-01-0170.

[Morita 08]   K. Morita, K. Shimamura, Y. Susuki, S. NishiKawa and al., *Development of a HILS Method for Testing Heavy-duty HEVs*, 22nd International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium, EVS-22, 11p, Yokohama, Japan, October 23-28, 2008.

[Renard 07]   S. Renard, E. Bideaux, X. Brun, D. Thomasset, W.Marquis-Favre, P.Lager, *Machine Drive System Modelling for HIL Simulation,* 2nd International Conference : Automotive Power Electroniocs, SIA, Paris, France, September 26-27, 2007.

[Renard 08]   S. Renard, « Hardware in the Loop » validation of electronic architecture of hybrid powertrain for industrial vehicles, phD study, INSA Lyon, 261p, ISAL-0131, 2008.

Table 1: Test typology

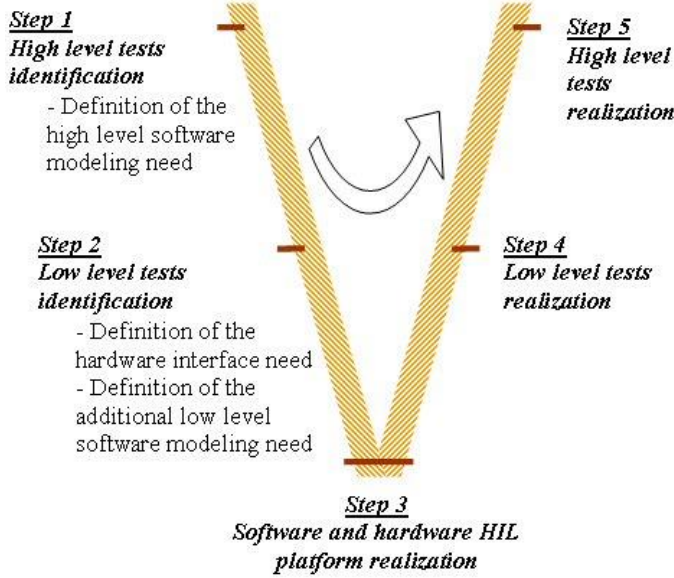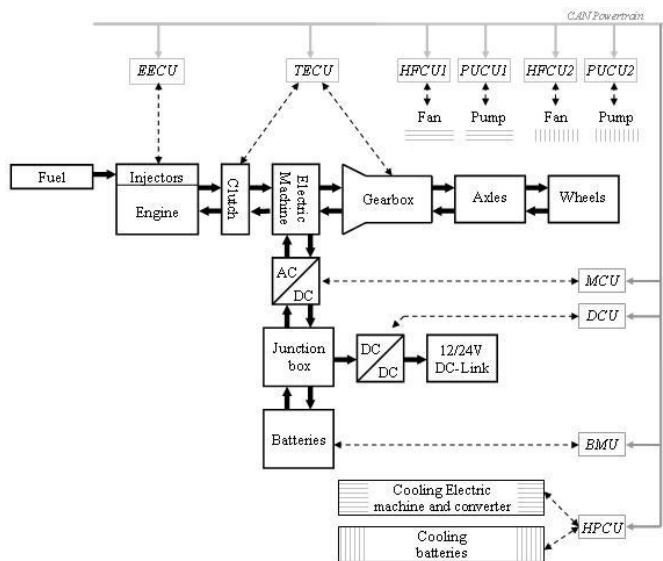| Type I – Low level tests |
| --- |
| 1) Supply test |
| 2) Network excitation test |
| 3) Network diagnostic test |
| 4) On/Off sensor excitation test |
| 5) Continuous sensor excitation tests |
| 6) On/Off power modulator diagnostic test |
| 7) Continuous power modulator diagnostic test |
| Type II – High level tests |
| A – Normal operation tests |
| 1) Test in order to check a target state |
| 2) Test in order to check a dynamic response |
| 3) Test in order to check a state sequence |
| B – dysfunctions relative tests |
| 1) Test in order to check the boundaries behavior |
| 2) Test in order to check a downgraded mode |

Figure 1: HIL platform connection
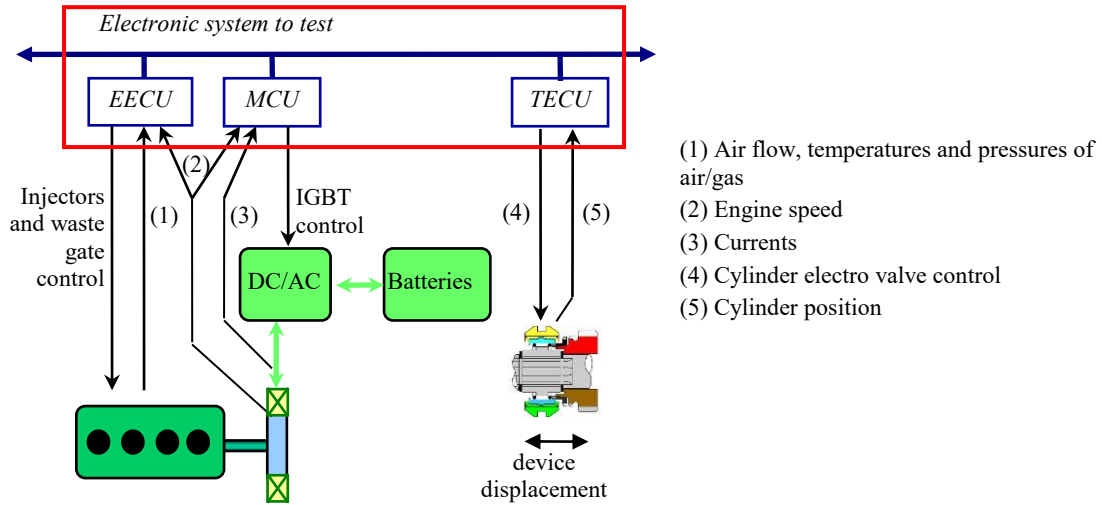
Figure 2: Method to specify a HIL platform

Input data: Requirements of the component or system to test

Step 1
High level tests
identification

- Definition of the high level software modeling need

Step 5
High level tests realization

Step 2
Low level tests identification

- Definition of the hardware interface need
- Definition of the additional low level software modeling need

Step 4
Low level tests realization

Step 3
Software and hardware HIL platform realization

Figure 3: Hardware and electronic architecture

Figure 4: Environment of the first requirement

(1) Air flow, temperatures and pressures of air/gas
(2) Engine speed
(3) Currents
(4) Cylinder electro valve control
(5) Cylinder position

Figure 5: Environment of the second requirement

(1) Currents
(2) Input shaft speed
(3) Output shaft speed
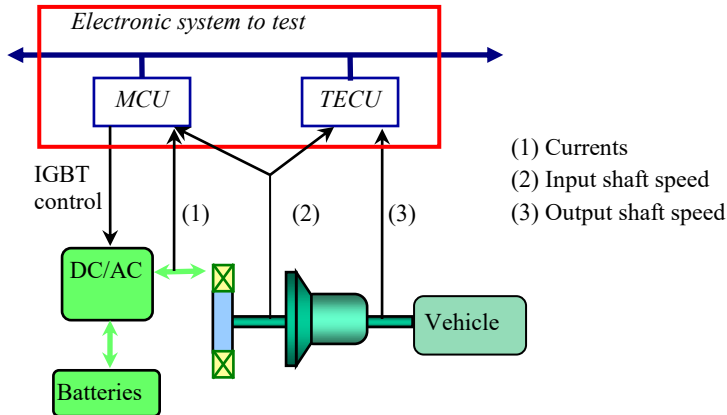
Figure 6: Environment of the third requirement

(1) Injectors, waste gate and self retardation system
(2) Air flow, temperatures and pressures of air/gas
(3) Engine speed
(4) Cylinder electro valve control
(5) Clutch position
(6) Input Shaft Speed
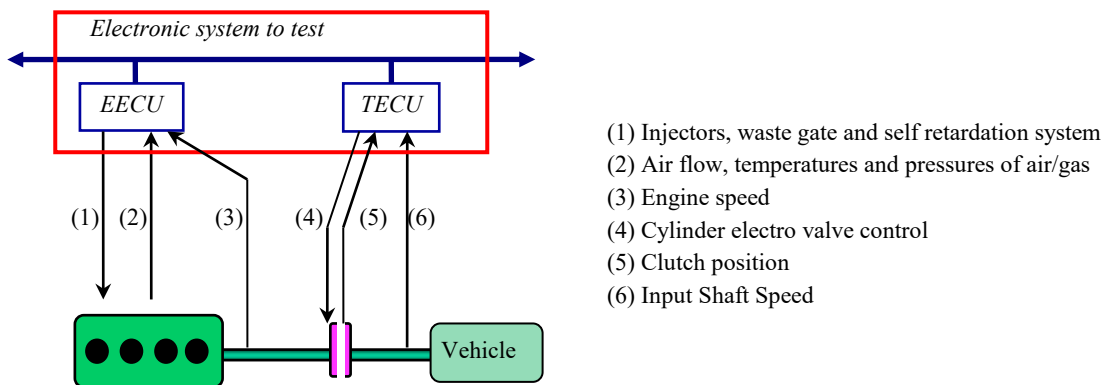
Figure 7: Complete needed environment

(1) IGBT Control
(2) Currents, electric machine speed
(3) Injectors, waste gate and self retardation system
(4) Air flow, temperatures and pressures of air/gas
(5) Engine Speed
(6) Clutch cylinder electro valve control
(7) Clutch position
(8) Gearshift cylinder electro valve control
(9) Forks position, input gear speed, output gear speed.
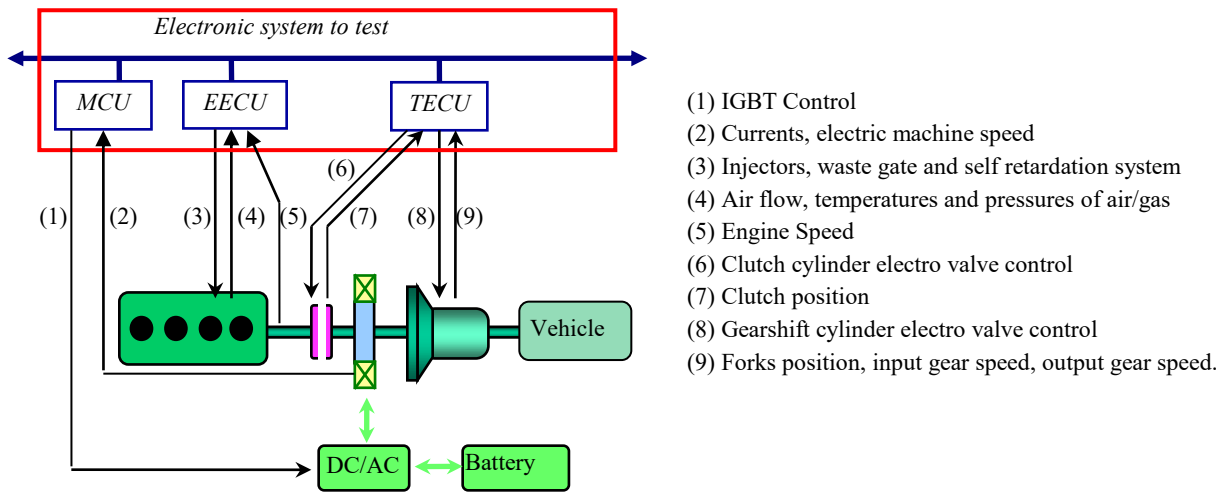
Figure 8: EECU test statistics