INTERNATIONAL DESIGN CONFERENCE - DESIGN 2016
Dubrovnik - Croatia, May 16 - 19, 2016.

KOPPELSYSTEMS: OBLIGATORY ELEMENTS WITHIN VALIDATION SETUPS

A. Albers, T. Pinner, S. Yan, R. Hettel and M. Behrendt

Keywords: validation, design evaluation, design methods, design models, modelling

1. Introduction

Validation is an essential part of product engineering which is driven by the system of objectives (e.g. by product requirements, application aims). Validation activities represent the backlink between the current product state and the underlying system of objectives: validation reveals differences between created objects (like the product) and their objectives and, hence, it creates new objectives or completes the existing ones and serves as a source for new design solutions [Albers 2010].

Within validation activities, several validation models represent specific properties or functions of the system in development (SiD, see 2.3) or its user or environment. These models must be designed or selected according to the specific validation purpose within the product engineering process. To support these tasks several methods and descriptive models are available that focus on the validation models. For the installation on the test bench certain technical systems must be selected and installed, e.g. to interconnect simulation models and physical hardware with each other. Some research has been carried out to identify requirements on the models used in validation setups. But these requirements focus predominantly on simulation models in terms of real-time execution or model maturity and the simulation interface between simulated environment and the product [Demers et al. 2007] cp. [Monti et al. 2005].

Learning from conducted validation activities, it turns out that not only the used validation models determine the system behaviour of a specific validation setup. In fact, virtual or physical connections between validation models appear to have a strong influence on validation results. The virtual and physical models of a validation setup define the desired model behaviour, but further technical connecting systems (like a clutch actuator or a shifting robot) are necessary to interconnect these models (e.g. virtual driver behaviour and physical gearbox). Section 2.3 (Top-Down Modelling of Validation Environments) describes one example for a validation activity that is influenced by such connecting systems presented above.

2. State of the art

2.1 Validation in product design

Validation activities replenish the product's system of objectives as stated above. Hence, a continuous validation approach is necessary starting within early phases of product engineering. Validation triggers the development of product models (and further models) for various phases like idea generation, conception, and construction. In this understanding, validation pulls further models as an outcome from the product engineering phases for following validation activities. Therefore, this approach is called "pull approach of validation" [Albers et al. 2016]. As a result, both product and validation related results are generated simultaneously during product engineering.

Validation is a necessary activity in vehicle development and, therefore, a supporting framework has to meet various requirements (cp. [Albers and Düser 2010]). Established validation approaches share the idea of testing a specific subsystem of the product "in-the-loop". The loop is regarded as a model-based representation of the system's surrounding, like in Model-in-the-Loop (MiL), Software-in-the-Loop (SiL) or Hardware-in-the-Loop (HiL) setups [Bringmann and Krämer 2008]. Based on this idea, the IPEK-X-in-the-Loop (IPEK-XiL)-Approach has been developed [Albers and Düser 2010] that extends the scopes of previous approaches regarding requirements from mechanics and mechatronics [Albers et al. 2016]. It regards the System in Development (SiD) as only one validation model amongst the further ones of the Connected Systems (CS). Validation models can be virtual, physical or mixed. Compared with the common understanding of HiL setups (cp. HiL powertrain setups in [Powell et al. 1998], [Raman et al. 1999]) XiL expands the understanding of "Hardware" and "Loop" as it is defined by the validation purpose.

2.2 Modelling of technical systems

Generally, a model is a representation of an existing original (e.g. taken from reality). This representation comprises only relevant properties of the original and is designed for a specific purpose. [Stachowiak 1973] For models that are used for validation purposes, this definition applies as well: models may represent parts of a system in development; they can also represent parts of the product's environment or the product user. Every one of the models is designed and selected for a specific purpose of validation and they comprise only relevant properties of their originals. These models may be virtual (like a virtual driver model or a virtual simulation model for a combustion engine) or physical (like a product prototype or a flywheel representing rotational masses). Several models interact with each other to provide a desired system behaviour that is used to conduct a validation activity. In order to conduct a validation activity a suitable set of models is selected. These models define the model-based behaviour of the validation setup

In general, a validation model and further test bench hardware are technical systems. To enable a formalized description of technical systems, various descriptive models are available. One of these models is the Contact and Channel Approach (C&C^2-A). Generally, this approach relates a system's embodiment to its technical functions [Albers and Wintergerst 2014]. It comprises three basis model elements to describe technical systems: Working Surface Pairs (WSP) are interfaces between elements, established with two Working Surfaces (WS). As one prerequisite to fulfil a technical function, at least two WSP are required as well as an interconnecting Channel and Support Structure (CSS) that connects these two WSP. Furthermore, system input and system output from outside the system border are represented using at least two Connectors (C). They comprise the system behaviour from further systems that is relevant for the systems under consideration [Albers and Wintergerst 2014].

2.3 Top-down modelling of validation environments

For planning and documentation of validation activities several modelling approaches are available. Stechert shows an approach within his methodology for complex requirements modelling that represents properties of the test criterion, the test case and the test environment [Stechert 2010]. It relates test criteria to product requirements and keeps focused on the product under development [Stechert and Franke 2009]. For the practical realization of a validation activity (like a test bench development), a top-down modelling approach is of advantage [Geier et al. 2012], [Albers et al. 2016]. This approach subdivides connected systems (like the high-level systems "Driver", Environment" and "Vehicle" within the context of drivetrain engineering) into the relevant sub-systems. These systems are categorized into one or more System(s) in Development (SiD) and further Connected Systems (CS). All of them can be represented as a virtual, physical or mixed model. Also, the single model fidelity can vary depending on the validation aim (cp. [Stachowiak 1973]). Figure 1 shows an exemplary representation of a powertrain validation setup. Every model is represented as a small text box with an icon. In addition, there is a depiction of sensor-actuator-systems that are meant to connect virtual and physical models.

Combé et al. present an approach to realize a drivetrain simulation using several models of different levels of detail [Combé et al. 2005]. They propose a layer approach to cope with different model fidelities. Depending on the application scope, the suitable model level is identified. Such a layer approach raises the models' flexibility and interoperability on the model level.

Example - Powertrain test bench: description and influencing factors

A Powertrain in the Loop test bench (PLP, see Figure 2 left) is used to provide a specific system behaviour to parts of a vehicle powertrain in order to make it feel and behave like being used within a real vehicle. The following use-case of the PLP represents a specific validation setup that is used to examine the course stability of a vehicle [Pinner et al. 2013, 2015]. This behaviour depends on the existing tolerances in shaft elasticity and tooth clearance. To identify the specific powertrain behaviour a gearbox with side shafts builds the System in Development (SiD). Further parts of the drivetrain (e.g. flywheel and clutch) are physically installed on the test bench. Other components of the vehicle - e.g. combustion engine, chassis, tyres or environment and driver - are provided with virtual models. The link between virtual models and physical components on the test bench is established by use of sensor-actuator systems, such as electric drives to transfer the combustion engine behaviour to the gearbox or as a shifting robot to change the selected gear according to the virtual driver demands.

The validation activity led to the knowledge about the theoretical displacement in driving direction of the car in comparison to the idealized straight on drive. The resulting vehicle-speed over time progress does not only depend on the selected validation models and their parameterization but also on the behaviour and the interaction of the sensor-actuator-systems.

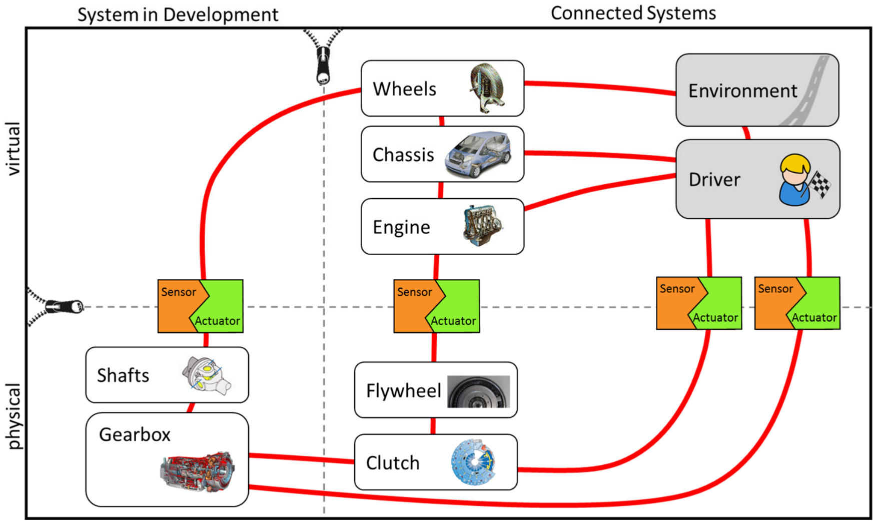

Figure 1. Representation of the Powertrain in the Loop validation setup (cp. [Albers et al. 2016], [Pinner et al. 2013])

These sensor-actuator-systems are shown in Figure 1 just on the border between the physical and virtual models of the validation setup. They are (from left to right):

- Load engines and torque sensors: According to properties of the virtual vehicle model and according to the manoeuver the load engines are provided with the calculated load on the powertrain shafts. The electric engines transform this torque demand into a physical torque at the shafts. Torque sensors capture the torque values and use them for monitoring and control.
- Drive engine and torque sensors: The drive engine provides torque to the flywheel according to the driver demand and the combustion engine's behaviour. Torque sensors transduce the physical torque values.
- Clutch actuator: The clutch actuator realizes a percentage clutch opening value according to the virtual value provided by the virtual driver model. Position sensors monitor the robot's position and the clutch state.

- Shifting robot: The shifting robot is connected to the virtual driver model. The virtual driver provides the desired gear and the robot uses internal shifting models to conduct a shifting process. It actuates the gearbox' shifting lever. Position and force sensors monitor the robot's gearshift state.

Furthermore, test bench properties influence the overall validation setup behaviour by interfacing with some of the validation models, both of virtual and of physical kind:

- Gearbox mounting plate: The way in which the gearbox is connected to the test bench has a strong influence towards the gearbox' system behaviour. The mounting plate's stiffness determines the frequency and vibration behaviour of the gearbox in comparison to its behaviour inside a car.
- Simulation data interfaces: Interfaces between simulation models are subject to errors caused by problems in time synchronization (e.g. concerning sampling rate, resolution).
- Physical entities between SiD and sensor: Physical parts (clutches, shafts) that are needed to connect to the SiD may add an influence to the SiD's system behaviour. This can be caused by rotational masses or damping behaviour (cp. [Albers et al. 2015a]).

3. Motivation and research aim

Validation setups are established during different phases of product engineering. For planning, realization and documentation the identification of relevant validation models has proven to be beneficial as well as their breakdown into virtual and physical validation models (cp. [Stechert 2010], [Albers et al. 2016]). Interoperability of models is ensured either on the model level itself or by using a mediator between models.

Several descriptive models are available for planning and documenting validation setups. Nevertheless, these descriptions focus predominantly on models and their properties and understand the need for model interoperability as a model requirement. Some interconnecting systems are represented as sensor-actuator-systems but there is no universal method for considering systems that interconnect models in virtual, physical or mixed validation setups.

The example within section 2.3 depicts several systems that influence the overall validation setup behaviour without being meant as a model. Besides sensor-actuator-systems, the gearbox mounting plate turned out to be of high relevance for the gearbox' behaviour on the powertrain test bench. It is a mechanical part between the physical validation model "Gearbox" and the test bed. Another influencing factor for the described validation setups is the connection of virtual validation models: it requires a data transmission system to connect virtual models and to realize the overall setup. This transmission system can apply an unwanted behaviour like time-delay or bandwidth limits. However, it is needed to cope with the validation setup's boundary conditions (e.g. by using real-time data interfaces).

Aim of this paper is the introduction and application of a new descriptive model to consider systems that interconnect models in virtual, physical or mixed validation setups. To allow further consideration of such systems within validation activities and to allow their selection and documentation, the authors suggest the introduction of a descriptive model for such "Koppelsystems" to describe them (from the German word "koppeln" which is "to connect" or "to link"). The objective of this paper is to discuss the role of Koppelsystems within validation activities. Therefore, several exemplary validation activities are presented and analysed. The distinction between models and Koppelsystems within validation is discussed as well.

4. A new element to describe model connections

The descriptive model of Koppelsystems is introduced. This is a generalized description of systems that interconnect two (sub)-systems of validation setups with each other.

4.1 Definition

Models and Koppelsystems are parts of validation environments. Models (virtual, physical or mixed) can represent the relevant system behaviour within validation activities. Koppelsystems can be *necessary to connect validation models. One model provides model output that is meant as an input for another model. Koppelsystems are every system that is between these models.*

The main function of Koppelsystems is to provide a connection between models in validation setups. In comparison to models, Koppelsystems are not meant to add a relevant system behaviour. However, they are necessary for the realization of the models' system behaviour. An unsuitable Koppelsystem biases or avoids the desired model behaviour. Koppelsystems are needed if a connection or linkage cannot directly be established between models in validation e.g. because of an incompatible model input and output. The Koppelsystems' task is to receive model output flows and to provide these flows to further models, without biasing their transmission. Generally, Koppelsystems have a specific (if possible well-known) transfer function for input and output flows.

A Koppelsystem comprises at least two Working Surfaces (WS, see 2.2) with one Channel and Support Structure (CSS, see 2.2). The WS connect to further WS of the connected models and build Working Surface Pairs (WSP, see 2.2). Koppelsystems can both connect to physical and to virtual flows respectively physical and virtual models. In general, the transfer path is considered as bi-directional.

For virtual and physical models three types of Koppelsystems can be identified:

- Physical/physical Koppelsystem KS_PP: A Koppelsystem between two physical models, e.g. the connection between the vehicle chassis and the test bench environment.
- Virtual/virtual Koppelsystem KS_VV: A Koppelsystem between two virtual models, e.g. the connection between distant virtual models via network.
- Virtual/physical Koppelsystem KS_VP: A Koppelsystem between one virtual and one physical model, e.g. the robot shifter that transforms a virtual gear change demand into physical action.

4.2 Koppelsystems: abstract view

A Koppelsystem can be regarded as a system with inputs and outputs of system flows, e.g. with two inputs and two outputs. Input 1 and Output 1 connect to a specific validation model and Input 2 and Output 2 to another validation model. The input- and output-ports can be regarded as Working Surfaces, their connection is regarded as a sub-system of Working Surface Pairs and Channel and Support Structures. Following the understanding of the C&C²-approach, the models provide working surfaces that connect to the Working Surfaces provided by Koppelsystems and build Working Surface Pairs (WSP). Hence, a Koppelsystem can only realize its intended function when it interacts with its intended validation models according to the basis definition of C&C²-A. The type of the connection (the WSP) between a Koppelsystem and its neighbouring validation models depends on the type of the Koppelsystem itself. In general, the Koppelsystem's WSP and the internal LSS properties define the transmission behaviour and, hence, define the system's application range.

Based on the C&C²-modelling approach Koppelsystems and their interaction with models can be represented on different levels of detail: a Koppelsystem connects directly to the models or a Connector can comprise a Koppelsystem and all following sub-systems to represent their relevant properties to the adjacent model. Even models and their input and output to the Koppelsystem can be represented by Connectors.

4.3 Application examples

This section analyses three exemplary validation activities using the abovementioned understanding of Koppelsystems. The three examples are Powertrain in the Loop test bench (PLP, see Figure 2), Vehicle in the Loop test bench (VeHiL, see Figure 2) and a Distributed Validation Environment.

4.3.1 Example focussing on KS_VP: Powertrain test bench

Figure 3 (left) shows the integration of Koppelsystems into the representation presented before (cp. Figure 1). Every one of the sensor-actuator-systems is considered as one Koppelsystem and is named to identify the respective technical systems behind. Additionally, a KS_VV is inserted to clarify the relevance of the simulation data interface between the virtual models "Driver" and "Engine"/"Chassis" for the behaviour of the virtual models within this specific validation activity.

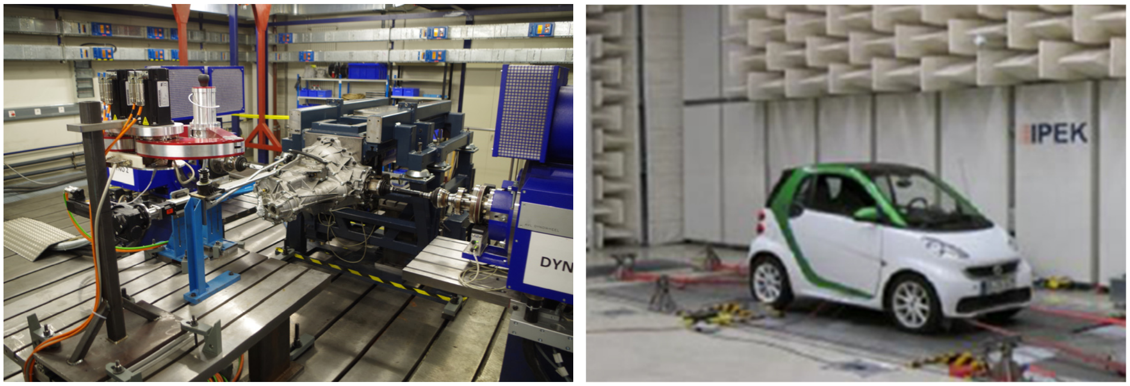

Figure 2. Exemplary validation activities (from left): gearbox test on the Powertrain in the Loop test bench (PLP), vehicle test on in the Loop test bench (VeHiL)

With this view on the setup of models and necessary Koppelsystems additional information is provided in comparison to the former representation in Figure 1. With the definition of Koppelsystems in mind, the model "Driver" defines a desired behaviour of both the clutch (concerning engagement and disengagement) and the gearbox (concerning the selected gear). This desired behaviour is transferred between the models by the Koppelsystems "Clutch Actuator" and "Shifting Robot". In this case, they are only a means to transfer predefined in- and output signals from one model to another.

In contrast to this setup, other validation activities could need a specific humanlike shifting behaviour or a physical model of the contact to the shifting lever. In these cases, additional models would be needed to define the desired behaviour (according to the definition).

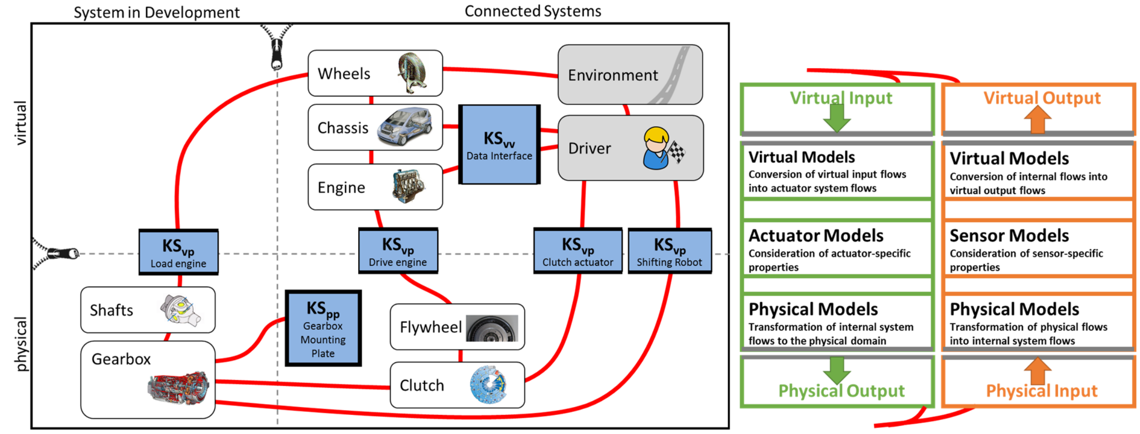

Figure 3. Representation of the Powertrain in the Loop validation setup with Koppelsystems (left) and descriptive model for KS_VP (cp. [Pinner et al. 2015]) (right)

In addition to the presented descriptive model of Koppelsystems, more detailed models can be derived to focus on specific properties. One example for such a model is given by Pinner at al. [2015] to describe virtual/physical Koppelsystems (see Figure 3, right). Therein, a Koppelsystem_VP is described as a combination of various models. Depending on the validation objectives, any of these models can be relevant and, hence, it would be represented as a specific model within the representation of the validation setup (like Figure 3, left).

4.3.2 Example focussing on KS_VV: distributed validation

A distributed test bench configuration was set up to demonstrate a location-independent validation environment. The motivation behind these efforts is the enhancement of combined physical and virtual validation across different locations. This is particularly required to support the fast and flexible cooperative validation either including different locally distributed company departments or within a supplier-OEM-relationship. The use-case of this validation setup was to investigate the longitudinal vehicle drivability for different driver types of different countries whereas the electric engine was the SiD. However, at first the objective was to focus on the connectivity of the two test bench systems. The overall system (vehicle, driver and environment) and its sub-systems were spread over different locations [Albers et al. 2014b]. The electric engine, the power electronics, the rest-vehicle system and the environment were represented at the Clean Energy Automotive Engineering Center, Tongji University, while the driver is located at the IPEK-Institute of Product Engineering, Karlsruhe Institute of Technology (KIT). Figure 4 shows the configuration including the Koppelsystems.

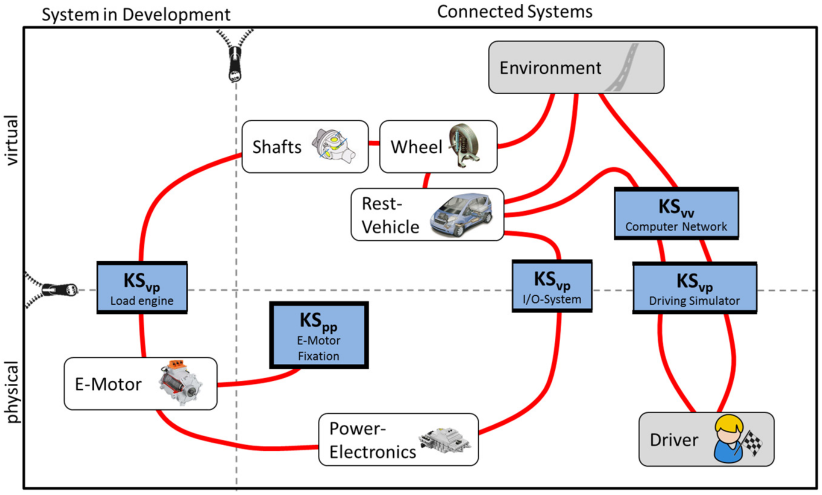

Figure 4. Representation of a distributed validation setup including Koppelsystems

The result depends not only on the used models itself. The Koppelsystems and its properties have an essential influence on the results. They are:

- Load engine: According to properties of the virtual vehicle model and to the manoeuver the electric load engine provides a calculated torque on the output shafts of the E-Motor.
- I/O-System: The I/O-System is able to provide the relevant CAN-Bus signals according to a real vehicle. It transfers the calculated torque demand into a signal for the power electronics.
- Driving Simulator: The driving simulator connects the human driver to the virtual rest-vehicle model and represents typical input devices for a driver (e.g. pedals or steering wheel). Additionally, the interface is able to generate visual and haptic feedback of the vehicles condition according to the calculated state (e.g. force feedback-steering wheel). The current vehicle condition in the virtual environment is displayed by a large monitor providing a similar forward view compared to a driver sitting in a physical vehicle.
- Computer Network: The network and its interfaces interconnect the models between the locations. They have a major impact on the overall system behaviour. Even when the single virtual models at the two places have a high sampling rate the interfaces in between can slow down the system behaviour significantly. Beside the interfaces there are many other influencing factors for the delay (e.g. distance between both locations or traffic load)
- E-Motor Fixation: The motor has to be connected to the test bed so that a generated torque on the output shaft of the motor can be supported. Due to the validation objective the testbed connection can be as simple as possible because it has little influence on the longitudinal vehicle dynamic. In case of a more detailed investigation on the vibrational behaviour of the motor it has to be designed as close as possible to the real connection in the vehicle. Here, the fixation is regarded as a physical model for the original motor connection in the vehicle.

The feasibility of a spatial separated validation environment is given under certain constrains. Main restrictions are the data transfer quality (frequency and delay) between the different models. Therefore, the admissibility for a distributed environment depends heavily on the validation objective and connectivity in between the validation environment.

4.3.3 Example focussing on KS_PP: Vehicle-in-the-Loop test bench

A Vehicle-in-the-Loop test bench can be used to provide the system behaviour of the environment and the driver to the vehicle. When performing driving manoeuvers the vehicle feels like being on a test track driven by a driver. A typical application is the evaluation of driving comfort of the overall vehicle. Hereby, a typical validation setup includes a driver, a test track and a vehicle. The driving comfort consists of a subjective perception of the lower and medium frequency chassis vibrations as well as the audible noise that is perceived by the driver. To save costs and avoid inconsistent environmental influences on the test track leading to non-reproducible evaluation results, the vehicle developers increasingly use the chassis dynamometer for complex tests. In addition, the evaluation by the perception of a human subjective can also be objectified by using an automated evaluation based on acceleration measurements.

A typical test setup on the Vehicle in the Loop test bench (VeHiL) is depicted in Figure 5 and consists of three main components [Albers et al. 2014a]:

- Chassis dyno: The longitudinal driving behaviour of the test track is reproduced by a roller and dynamometer. It transfers the calculated driving resistance of the virtual environment simulation to the wheels of the vehicle.
- I/O-System: The I/O-System synchronizes the state and configurations of the vehicle with its virtual model.
- Driving robot: The driving robot is fixed to the driver's seat and consists of pedal and gear actuators. It substitutes the human driver input to the vehicle driving on the test bench.

Additionally, the Koppelsystem "Vehicle Fixation" turned out to be of high relevance:

- Vehicle Fixation: The fixation holding the vehicle in position above the roller apex has great impact on its behaviour on the test bench. There are different kinds of vehicle fixations that differ in geometric, mechanical properties and installation.

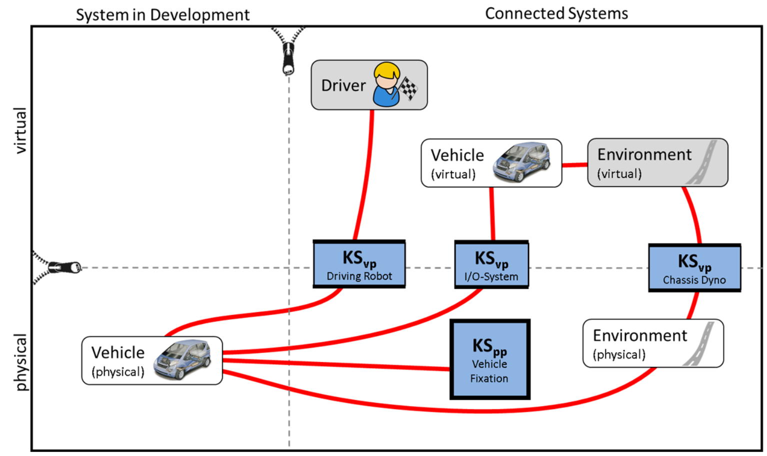

Figure 5. Representation of the Vehicle in the Loop validation setup

One application example is shown in [Albers et al. 2015b], where the main focus is on the comfort evaluation of combustion engine start/stop and vehicle acceleration in a common hybrid electrical vehicle. Hereby, the evaluation is not independent from the influencing Koppelsystems and the vehicle fixation holding the vehicle in the right position above the roller apex during the test had a great impact on the evaluation [Albers et al. 2015b]. Depending on the specific vehicle and the selected executed manoeuvre, the suitable fixation has to be selected to maintain the desired model behaviour.

5. Discussion of "Koppelsystems"

Adding Koppelsystems into the discussion about validation adds a specific view on this topic. Koppelsystems are regarded as parts of validation setups that are not meant as a model but as a connection between models. With this additional view on validation, further systems come into focus. These systems may be the mechanical connection between engine and test bench or the vehicle fixation on the roller dyno as well as data interfaces between virtual models or an electric drive that connects virtual and physical models. Especially when already existing models developed independently from the current validation objective are reused in one validation setup together, the model compatibility is not naturally given. One vivid example is a virtual/virtual Koppelsystem which connects different simulation models running on different modelling software: these are naturally not necessarily able to interact due to different features and properties (e.g. solver algorithms, step size or modelling fidelity). Hence, it is obligatory to use such a Koppelsystem within the validation setup.

Referring to the application example focusing on virtual/physical Koppelsystems the shifting robot connects the virtual driver model with the physical gearbox (see 4.3.1). Hereby, the model fidelity of the used driver model in terms of shifting behaviour is very low, by only giving discrete values for each shifting action (e.g. 1st gear to 2nd gear). However, for the used prototype of a gearbox this virtual model output does not match the required input to perform a shifting action as a force and direction-over-time-behaviour is needed to actuate the gear shift lever. If this force and direction behaviour was a relevant factor, it would be represented as a e.g. virtual shifting model.

A crucial question for the selection of validation models is whether a Koppelsystem can be understood as a transfer element between models or if it is needed as an explicit model providing a specified behaviour. With this understanding Koppelsystems (or their parts) change over to models and models (or their parts) may change over to Koppelsystems. The difference between a Koppelsystem and a model is their respective meaning within the specific validation activity: a model adds a desired behaviour; a Koppelsystem enables the models to interact after all and in the desired way. The presented test bench examples show a specific distinction into Koppelsystems and models that is individual for the respective validation purpose. In every case, Koppelsystems must suit to the selected models and fit to the validation objectives.

6. Summary and outlook

Validation as an essential part of product engineering [Albers 2010] gives feedback of a created system of objects and their underlying system of objectives. Within validation activities the created models represent specific properties of the system in development and connect systems. They are designed and selected according to the validation purpose. This contribution focuses on the systems interconnecting the different kinds of models having a strong influence on the result of a validation activity. The descriptive model of Koppelsystems is introduced. It describes a system that is necessary to connect in- and output flows of interacting models if they are not able to. They are obligatory to link models which are not able to interact or are of different level of fidelity so that their in- or output flows do not match their respective requirements. In general, a Koppelsystem can both connect to physical and to virtual flows respectively physical and virtual models. According to the model understanding of C&C^2-A, a Koppelsystem establishes two Working Surface Pairs with its connected models providing itself as a Channel and Support Structure (CSS) realizing the desired function. Different application examples are given to present relevant Koppelsystems and to discuss their relevance for the specific validation activities. The examples clarify that whether a system on the test bench is regarded as a Koppelsystem or as a validation model depends on the specific validation purpose.

Future research will be done to support the formalisation and selection of suitable Koppelsystems according to the specific validation objectives and the properties of the validation models. Such a method could integrate with existing methods to describe and distinguish validation activities. The formalized description of Koppelsystems can also help to build a collection of systems for validation activities to identify the suitable models for validation and to implement and connect them on the test bench. With this, the conduction of validation activities may be accelerated and improved. By the direct connection between creation and validation activities such an improvement will be propagated to the product design as well.

References

Albers, A., "Five Hypotheses about Engineering Processes and their Consequences", Proceedings of the Eighth International Symposium on Tools and Methods of Competitive Engineering - TMCE 2010, Horváth, I., Mandorli, F., Rusák, Z. (Eds.), Delft University of Technology, Delft, 2010, pp. 343–356.

Albers, A., Behrendt, M., Klingler, S., Matros, K., "Verifikation und Validierung im Produktentstehungsprozess", Handbuch Produktentwicklung, Lindemann, U. (Ed.), Carl Hanser Verlag, München, 2016.

Albers, A., Boog, S., Koch, C., Stoll, C., "Identifikation des Federschnittmoments in Dämpfersystemen während verbrennungsmotorischer Anregung am hochdynamischen Komponentenprüfstand", Schwingungen in Antrieben 2015, VDI-Berichte, VDI-Verlag, Düsseldorf, 2015.

Albers, A., Düser, T., "Implementation of a Vehicle-in-the-Loop Development and Validation Platform", FISITA World Automotive Congress 2010, 2010.

Albers, A., Hettel, R., Schille, F., Behrendt, M., "The complexity of road-to-rig approaches and diskussion of influencing parameters on the example chassis dyno", FISITA 2014 World Automotive Congress, Maastricht, 2014a, p. 8.

Albers, A., Schille, F., Hettel, R., Behrendt, M., "Auswirkung der Validierungsumgebung und Manöverumsetzung auf Komfortbewertungen hybridspezifischer Triebstrangphänomene", VPC – Simulation und Test 2015, 17. MTZ-Fachtagung, Methoden der Antriebsentwicklung im Dialog, GmbH, S.F.W. (Ed.), Springer Vieweg, Hanau, 2015b, p. 17.

Albers, A., Wintergerst, E., "The Contact and Channel Approach (C&C2-A): Relating a System's Physical Structure to Its Functionality", An Anthology of Theories and Models of Design, Chakrabarti, A., Blessing, L. T. M. (Eds.), Springer London, London, 2014, pp. 151–171.

Albers, A., You, Y., Klingler, S., Behrendt, M., Zhang, T., Song, K., "A new validation concept for future automotive development", Proceedings of Fisita 2014, 2014b.

Bringmann, E., Krämer, A., "Model-Based Testing of Automotive Systems", Proceedings of ICST 2008, Presented at the International Conference on Software Testing, Verification, and Validation, IEEE, Lillehammer, Norway, 2008, pp. 485–493.

Combé, T., Kollreider, A., Riel, A., Schyr, C., "Modellabbildung des Antriebsstrangs zur Echtzeitsimulation von Fahrzeuglängsdynamik", MTZ Motortechnische Zeitschrift, Vol.66, No.1, 2005.

Demers, S., Gopalakrishnan, P., Kant, L., "A Generic Solution to Software-in-the-Loop", Presented at the Military Communications Conference, IEEE, Orlando, FL, USA, 2007, pp. 1–6.

Geier, M., Jäger, S., Stier, C., Albers, A., "Combined real and virtual domain product validation using top-down strategies", ASME Verification and Validation Symposium, 2012.

Monti, A., Figueroa, H., Lentijo, S., Wu, X., Dougal, R., "Interface issues in hardware-in-the-loop simulation", Presented at the Electric Ship Technologies Symposium, IEEE, 2005, pp. 39–45.

Pinner, T., Berger, J., Stier, C., Geier, M., Albers, A., "Gesamtfahrzeugvalidierung mit Schaltroboter", AutomobilKONSTRUKTION, 2013, pp. 40-41.

Pinner, T., Jost, F., Schmid, D., Albers, A., "Considering User's Impact in Validation Activities – an Approach for the Determination of Requirements", Proceedings of the 20th International Conference on Engineering Design (ICED15), Vol. 5: Design Methods and Tools – Part 1, The Design Society, 2015.

Powell, B. K., Sureshbabu, N., Bailey, K. E., Dunn, M. T., "Hardware-in-the-loop vehicle and powertrain analysis and control design issues", Proceedings of the ACC 1998, Presented at the American Control Conference, Vol.1, IEEE, Philadelphia, PA, USA, 1998, pp. 483–492.

Raman, S., Sivashankar, N., Milam, W., Stuart, W., Nabi, S., "Design and implementation of HIL simulators for powertrain control system software development", Proceedings of the ACC 1999, Presented at the American Control Conference, IEEE, San Diego, CA, USA, 1999, pp. 709–713.

Stachowiak, H., "Allgemeine Modelltheorie", Springer Wien New York, 1973.

Stechert, C., Franke, H.-J., "Managing requirements as the core of multi-disciplinary product development", CIRP J. Manuf. Sci. Technol. Vol.1, 2009, pp. 153–158.

Stechert, C., "Modellierung komplexer Anforderungen", 2010.

Tobias Pinner, Scientific Assistant (Dipl.-Ing.)
Karlsruhe Institute of Technology (KIT), Germany, IPEK Institute of Product Engineering
Kaiserstr. 10, 76131 Karlsruhe, Germany
Email: tobias.pinner@kit.edu