VII International Conference on Computational Methods for Coupled Problems in Science and Engineering
COUPLED PROBLEMS 2017
M. Papadrakakis, E. Oñate and B. Schrefler (Eds)

FORMAL VERIFICATION OF COUPLING PROPERTIES FOR AN AUTOMOTIVE SOFTWARE INTEGRATION ACROSS XIL

NATARAJAN NAGARAJAN*, EVREN ERMIS*, ANDREAS THUY* AND BERND-HOLGER SCHLINGLOFF†

*ETAS GmbH, Borsigstraße 14, 70469 Stuttgart, Germany
e-mail: firstname.secondname@etas.com, Web page: http://www.etas.com/

†Humboldt Universität zu Berlin
Rudower Chaussee 25, Johann von Neumann Haus, 12489 Berlin, Germany
e-mail: hs@informatik.hu-berlin.de, Web page: https://www.informatik.hu-berlin.de

Key words: Coupling Properties, Coupled Problems, Formal Verification, Applications, XiL

Abstract. Virtualization and desktop testing of an integrated system without inclusion of a physical hardware is a well-established concept due to today's abundant computing power availability. However, only few aspects of reality are introduced in steps into these virtual environments. The aspects of reality like hard-real time deadlines, timing events, coupling frequency and data synchronization between two subsystems in a system offer complexity without fair estimation of its consequence on the system behavior. In this paper, we describe the abovementioned complexity as the coupling properties detailed for a combustion engine example along with its controller. We formally verify the timing, safety, liveness and deadlock properties of the coupling by modeling them as timed transition systems. The example is verified for the idle speed control, smooth mode switching and for injection cutoff control where the interaction between the subsystems is very critical. The paper highlights a very important perspective of strong and weak subsystem coupling while transiting from Model-in-the-loop (MiL) to Software-in-the-Loop (SiL) and finally to Hardware-in-the-Loop (HiL). In conclusion, the input-output behavior of the coupled subsystems is also presented for a realistic observation of the control loop.

1 INTRODUCTION

Virtualization of hardware refers to the process of creating a virtual replica of its physically existing components. In the automotive embedded software environment, virtualization of Electronic Control Unit (ECU) hardware is an established approach for early

software validation [1], [2]. The ECUs in a modern car contain several hundreds of control function modules. Hence, early validation requirements of ECUs drive the activities on virtualization of ECUs.

In the V-Cycle for model-based automotive software development [3], the process steps such as Model-in-the-Loop (MiL), Software-in-the-Loop (SiL) and Hardware-in-the-Loop (HiL) emphasize the early validation of embedded control software. Figure 1 describes the system composition of an engine plant model and its corresponding controller variants in MiL, SiL and HiL process steps. As shown in figure 1, in MiL , SiL and HiL, the control function, the embedded control software and the ECU respectively, are validated by simulating them with the plant model. While validating the controller variants, model coupling plays an important role in the overall system behavior. A controller model/ an embedded control software/ an ECU is said to be coupled with a plant model when there exists an exchange of control signals and data between them. During the validation of embedded control software, the coupling between the controller and plant models varies significantly. While progressing from MiL to HiL, virtual artifacts like virtual buses are replaced by real hardware artifacts such as CAN or analog/ digital hardware. This inclusion of hardware and software artifacts introduce constraints in ensuring a correct coupling between the controller variants and the plant model.

In this paper, we address the constraints introduced by hardware and software artifacts; we categorize the nature of coupling at each of the abovementioned process steps; we derive formal specifications from closed loop engine controller requirements for a correct coupling between controller variants and the plant model. We denote these formal specifications as the coupling properties that must be satisfied at each of the abovementioned process steps.

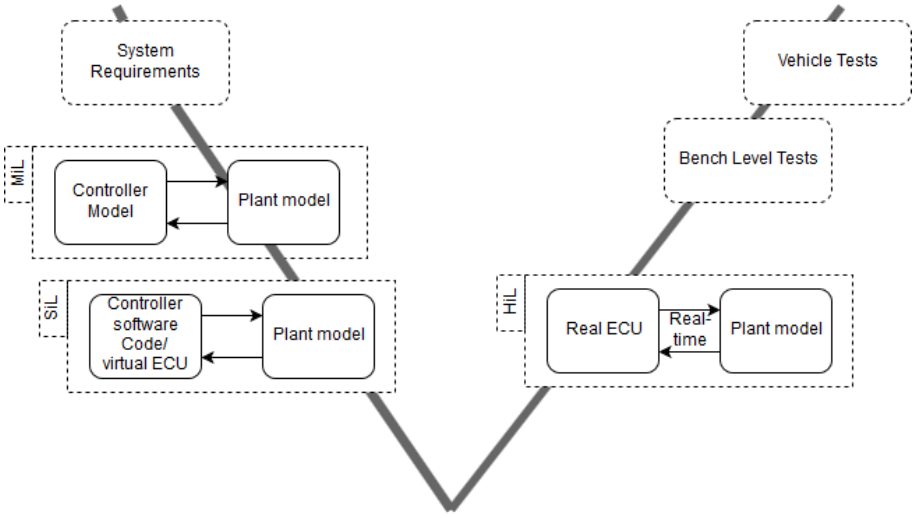

Figure 1: Automotive software development V-Cycle depicting an example system composition of an engine controller and an engine plant model in MiL, SiL and HiL process steps.

2 FORMAL VERIFICATION METHODOLOGY

In the field of engineering, the simulation of physical systems is important in validating the system behavior. However, unexpected system behavior may occur in reality due to incomplete test coverage of the system. Innumerous examples [4] exist in history where safety-critical systems encountered permanent failure after rigorous testing and validation. Formal verification [5] provides a solution by identifying such failures in advance and correcting the system behavior. In order to perform a formal verification, a formal description of the specification is a prerequisite.

In this paper, we describe some of the highly sensitive signals in the engine control loop such as fuel injection quantity, ignition angle, torque generation and throttle actuation. These signals influence the coupling between the controller variants and the plant model. The coupling properties are formally specified using Signal Temporal Logic [6] (STL). In STL, we specify real-valued signals in dense time; we test them on simulation traces generated out of the simulation runs. We prefer STL as it suits our application requirements and visualization of the results is straightforward.

2.1 Signal Temporal Logic (STL)

We provide an introduction to STL and its semantics [6], [7] before illustrating its application in the validation of control problems. Let $x_i[t]$ be a set of signals such that $i \in \mathbb{N}$ and t be a time instant, φ and ψ be STL Formulas, μ an atomic predicate given by $\mu = f(x_1[t], x_2[t], ..., x_n[t]) > 0$ where f is a real-valued function.

An STL formula is recursively defined as follows :

$$\varphi := \mu \mid \neg\varphi \mid \varphi \vee \psi \mid \varphi \, \mathcal{U}_{[\mathrm{a,b}]} \, \psi \tag{1}$$

The expression $(x_i, t) \models \varphi$ denotes that the STL formula φ satisfies the model of signals x_i at time t. The semantics of STL are given by the following clauses:

$$\begin{aligned}
(x_i, t) \models \mu &\iff f(x_1[t], x_2[t], ..., x_n[t]) > 0 \\
(x_i, t) \models \neg\varphi &\iff \neg((x_i, t) \models \varphi) \\
(x_i, t) \models \varphi \vee \psi &\iff (x_i, t) \models \varphi \text{ or } (x_i, t) \models \psi \\
(x_i, t) \models \varphi \, \mathcal{U}_{[a,b]} \, \psi &\iff \exists t' \in [t+a, t+b] \text{ such that } (x_i, t') \models \psi \\
&\qquad \text{and } \forall t'' \in [t, t'] \text{ holds } (x_i, t'') \models \varphi
\end{aligned} \tag{2}$$

In STL, the time references a and b ($a, b \in \mathbb{R}_{\geq 0}$) are added to temporal operators. We define two important temporal operators *eventually* and *always* as follows:

$$\begin{aligned}
&\textit{Eventually} : F_{[a,b]} \, \varphi = \top \, \mathcal{U}_{[a,b]} \, \varphi \\
&(x_i, t) \models F_{[a,b]} \, \varphi \iff \exists t' \in [t+a, t+b] \text{ such that } (x_i, t') \models \varphi \\
&\textit{Always} : G_{[a,b]} \, \varphi = \neg(F_{[a,b]} \, \neg\varphi) \\
&(x_i, t) \models G_{[a,b]} \, \varphi \iff \forall t' \in [t+a, t+b] \text{ holds } (x_i, t') \models \varphi
\end{aligned} \tag{3}$$

Informally, $\varphi \, \mathcal{U}_{[a,b]} \, \psi$ implies that for some time-step in simulation between the time references a and b the STL formula ψ holds true and for every time-step before ψ, the STL formula φ holds true. The temporal operators presented in this section are used to describe the properties of coupling.

3 X-in-the-Loop (XiL) and Coupling

In a V-Cycle for automotive software development, validating a system from MiL to HiL [8] and even further is strongly adhered to. The controller model development ends after its translation into an ECU in the HiL. Beyond the HiL, only the controller variables inside the ECU are adapted for the ECU network and vehicle validation. X-in-the-Loop (XiL) in the controller development process steps abbreviates MiL, SiL and HiL process steps. The X in the XiL represents a controller model in the MiL process step, embedded controller software in the SiL process step and an ECU in the HiL process step.

3.1 Coupling nature across XiL

Figure 2 shows a pictorial representation of our system composition in detail, explaining the coupling nature in the MiL, SiL and the HiL process steps. In the MiL process step, the simulation of an abstract physical plant model with corresponding control function is carried out. Here, we observe the physics involved and gain confidence on the developed control function. In this step, we validate the functional properties of the system. The simulation in MiL process step has a single numeric solver and therefore we denote the interaction between the participating models as *strongly-coupled* [9].

In the SiL process step, the controller model translates into a controller software with virtual software drivers and an operating system. The operating system is responsible for monitoring and triggering the internal events in the controller software. Therefore, the controller software can be concurrently simulated along with the plant model. The SiL process step can be executed both in virtual time and in real-time. In our case study, the SiL process step is performed in real-time. The interaction between the controller software and the plant model is handled by a single global clock. We categorize our SiL process step under *weakly coupled* [9] since the execution time of the control software and plant model can be chosen independently.

In the HiL process step, an ECU is in closed loop with the plant model. The ECU is a separate hardware entity with its own local clock and interacts with the plant model executed on another hardware platform. In the HiL process step, the simulation is real-time. We categorize the interaction of the ECU and the plant model in the HiL process step as *weakly coupled* as the ECU and the plant model have their own local time-scales.

In this paper, we derive closed loop engine control requirements and formalize them. These requirements characterize the closed loop system behavior. We therefore, refer to these requirements as coupling properties and further categorize them as timing, safety, deadlock and liveness properties.

We state the following: A model of a system (controller variants and the plant model) satisfies coupling properties in a XiL when every interpretation of the system model in MiL, SiL and HiL satisfy the respective coupling properties.

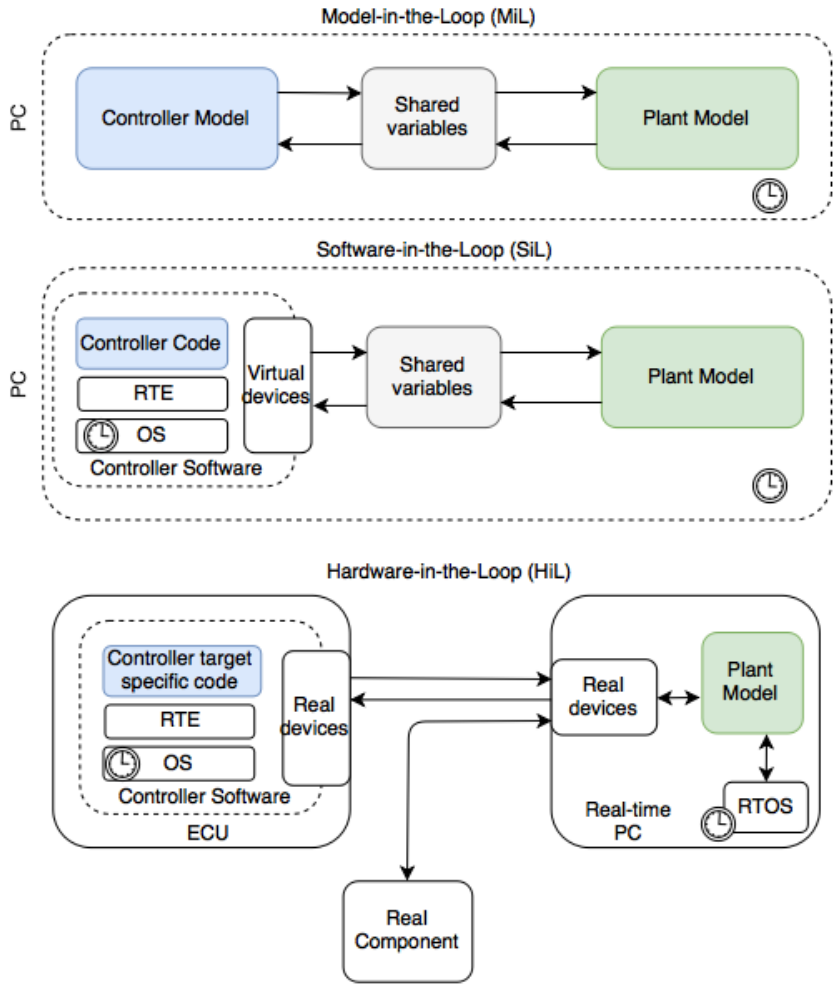

Figure 2: The picture showing the controller development and its coupling with the plant model in the MiL, SiL and HiL process steps.

4 CONTROL PROBLEMS AND COUPLING PROPERTIES

4.1 Control Problems

We have selected three highly critical engine control problems [10] to formally verify our coupling properties. The coupling properties have been formulated from the closed loop engine controller requirements. In this section, we describe each of the control problems and their significance.

Idle speed control [11] is an elementary example for validating a combustion engine closed loop control. During the idling of an engine, the engine speed must be maintained at an optimal desired value. The computations of engine speed in the ECU and in the engine model must be synchronous; any delay in fuel injection quantity estimation and ignition angle estimation will lead to abrupt variation in torque, causing the engine to stall. This demands strict timing and safety requirements in the coupling of the plant and the controller variants.

Smooth switching control is developed in order to avoid erroneous injection, ignition and throttle actuation while transiting to and from idle speed mode. During sudden acceleration demand of the driver, the ECU must linearly actuate the throttle to avoid a bad driving experience. In the above case, the ECU should eventually actuate the throttle by detecting the driver's intention to accelerate. Additionally, the ECU must not actuate the throttle for a duration longer than desired. It is important to witness how the deadlock and safety requirements of the coupling are being satisfied.

In injection cut-off control [12], the objective is to ensure that the passenger has minimal discomfort during release of the accelerator pedal. Upon quick release of the pedal, the torque demand is minimized. This is characterized by undesirable oscillations. In order to minimize the resulting oscillations, the engine controller must have a correct event detection mechanism. The injection cut-off control demands the injection and ignition timing events to be very precise. We translate them into timing requirements of the coupling.

4.2 Coupling Properties

The coupling properties are a set of formal specification of closed loop engine control requirements. These properties must be satisfied in order to ensure a correct coupling between the engine controller variants and the engine model. We consider that the plant and controller models are developed as per required specification and focus on coupling the two participating subsystems. An initial set of 12 properties have been formally verified on the case study. In this paper, we present the STL formalism of three timing properties namely φ_1, φ_2 and φ_3 and one safety property φ_4.

4.2.1 Timing Properties

We formally describe the critical event detection and timing interactions of the coupling between engine controller and engine plant subsystems. Some of the formalism have been generalized for verifying similar constructs. Table 1 describes the list of variables used in the formal description of the timing properties.

Table 1: List of variables used in the formalization of coupling properties.

Variable	Description	Range	Unit
D_{des}	Desired driver actuation	[0,1]	factor of 100 (%)
ω_{Lref}	Lower engine speed threshold	[0,6000]	RPM
ω_{Uref}	Upper engine speed threshold	[0,6000]	RPM
ω_{eng}	Current engine speed	[0,6000]	RPM
ω_{neng}	Calculated engine speed	[0,6000]	RPM
tol_v	Tolerance value	[0,6000]	RPM
t_i	Simulation start time	[0,∞]	seconds (s)
t_f	Simulation end time	[0,∞]	seconds (s)
KW	Current crank angle	[0,720]	degrees (^{0}CA)
CA_{Lx}	Lower crank angle threshold	[-720,720]	degrees (^{0}CA)
CA_{Ux}	Upper crank angle threshold	[-720,720]	degrees (^{0}CA)
E_{fqty}	Fuel injection request event	[0/1]	-
E_{ign}	Ignition angle request event	[0/1]	-
S_{fqty}	Fuel Injection duration	[0,5000]	milli seconds (ms)
S_{ign}	Ignition end angle	[-720,720]	degrees (^{0}CA)
$D_{throttle}$	Desired throttle angle	[0,90]	degrees

1. *In the idle speed control mode, no driver actuation is observed and the engine speed must be maintained within a specified upper and lower engine speed thresholds.*
We translate this specification in STL as follows:

$$\varphi_1 := G_{[t_i,t_f]} \left((\mid \omega_{eng}[t] \mid \leq \omega_{Uref}) \wedge (\mid \omega_{eng}[t] \mid > \omega_{Lref}) \wedge (D_{des} = 0) \right) \tag{4}$$

Explanation of the formulation:
The condition $D_{des} = 0$ implies that there is no driver actuation of accelerator pedal. The idle speed control should be robust enough to detect disturbance and counter-balance the effects. Therefore, it is important to check the engine speed for the complete time duration the controller is in idle control state i.e. in our case, ω_{Uref} is set to 800 RPM. To ensure that the engine does not stall while idling i.e. engine speed is zero, we must also check for a lower bound value ω_{Lref} . In precise formalism given by φ_1 , we use temporal operator *always* to describe a stronger notion on engine speed stability during idle control mode.

2. *A fuel injection request must be triggered within a specified upper threshold value in the crank angle scale.*
   We now translate this specification in STL as follows:

$$\varphi_2 := F_{[t_i,t_f]} \left((\mathrm{KW[t]} > \mathrm{CA_{Lx}}) \wedge \ (\mathrm{KW[t]} \leq \mathrm{CA_{Ux}}) \ \wedge (\mathrm{E_{fqty}} > 0.5) \right) \tag{5}$$

   Explanation of the formulation:
   The engine control functions are normally modeled in crank-angle scale. Therefore, each fuel injection request is a discrete event that is triggered to perform fuel injection estimation. One must ensure that right fuel quantity is updated per combustion cycle to enable desired engine operation.
   We define a measurement window having upper and lower threshold values in crank-angle scale such that we identify the request associated with particular cylinder in this region. i.e. we define the upper and lower threshold values for every cylinder of the engine.

3. *An ignition angle update request must be triggered within a specified upper threshold value in the crank angle scale.*
   We now translate this specification in STL as follows:

$$\varphi_3 := F_{[t_i,t_f]} \left((\mathrm{KW[t]} > \mathrm{CA_{Lx}}) \wedge \ (\mathrm{KW[t]} \leq \mathrm{CA_{Ux}}) \ \wedge (\mathrm{E_{ign}} > 0.5) \right) \tag{6}$$

4.2.2 Safety Property

We formally describe the safety requirements of the coupling between engine controller and engine plant subsystems. The STL formula φ_4 has been provided as an example.

1. *Always the difference between current engine speed in the engine model and computed engine speed in the ECU must be within tolerance limits.*
   We now translate this specification in STL as follows:

$$\varphi_4 := \mathrm{G} \left(|\omega_{\mathrm{eng}}[\mathrm{t}] - \omega_{\mathrm{neng}}[\mathrm{t}]| < \mathrm{tol_v} \right) \tag{7}$$

5 CASE STUDY

5.1 System Description

A three cylinder combustion engine model is coupled to its engine controller as shown in figure 3. An accelerator pedal model is provided to stimulate the driver's input for testing. The engine model [11] consists of analytical parts that involve differential equations and experimental parts that involve data from real measurements. The engine and controller models have processes which are time-dependent and crank-angle dependent. The engine processes related to combustion (the air system, throttle control, fuel injection and ignition) are computed with respect to crank-angle scale. The air flow, manifold

pressure, torque, crankshaft model and ECU event detection are computed with respect to the global simulation time.

Figure 3 shows the system composition with signal flow information between the subsystems. The engine speed is a very critical variable for effective synchronization between engine and controller models.

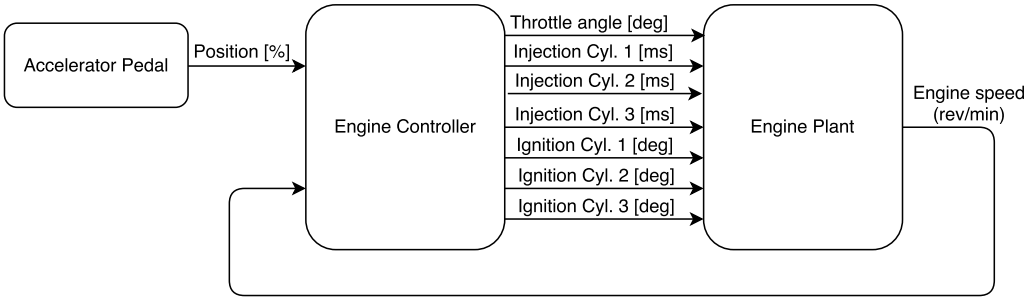

Figure 3: Model of engine controller and engine plant in closed loop

5.2 Results

We used the Breach [6] toolbox to analyze the STL formulations over the combustion engine example.

Figure 4 shows the simulation of engine speed and the satisfaction of STL formula φ_1 under idle speed control mode. The Breach toolbox categorizes the STL satisfaction problem into boolean and quantitative satisfaction. The boolean satisfaction indicates whether the property φ_1 has been satisfied within the specified simulation start and end time. The quantitative satisfaction provides a value for the variables describing the deviation occurring in the engine speeds from the thresholds. Figure 4 shows the boolean satisfaction (the red line) of each sub-formula and quantitative satisfaction (blue line) indicates the deviation. In formula φ_1, we set x as 6 seconds and run the simulation for 30 seconds. We witness that the idle speed is oscillating between 800 RPM and 850 RPM in the interval between 5 and 5.5 seconds. Hence, the boolean satisfaction is pulsating during this time period. We observe that the property is satisfied after 6 seconds.

Figure 5 shows the response curves of sensitive signals (engine speed, air-fuel ratio factor and desired throttle demands) for a step input with varying amplitudes of driver actuation. We performed a partial coverage test on the example by varying the driver actuation signal from (0-100)%. A random set of 15 driver actuation signals were simulated and satisfaction of the coupling properties were verified. At lower engine speeds, the driver demand causes the engine speed to go as low as 500 RPM. On higher engine speeds, the engine compensates the driver demand and achieves a higher engine speed. From this behavior, we inferred that the engine could be susceptible to stalling with lower sudden driver actuation. To confirm our inference, we analyzed the example by randomly

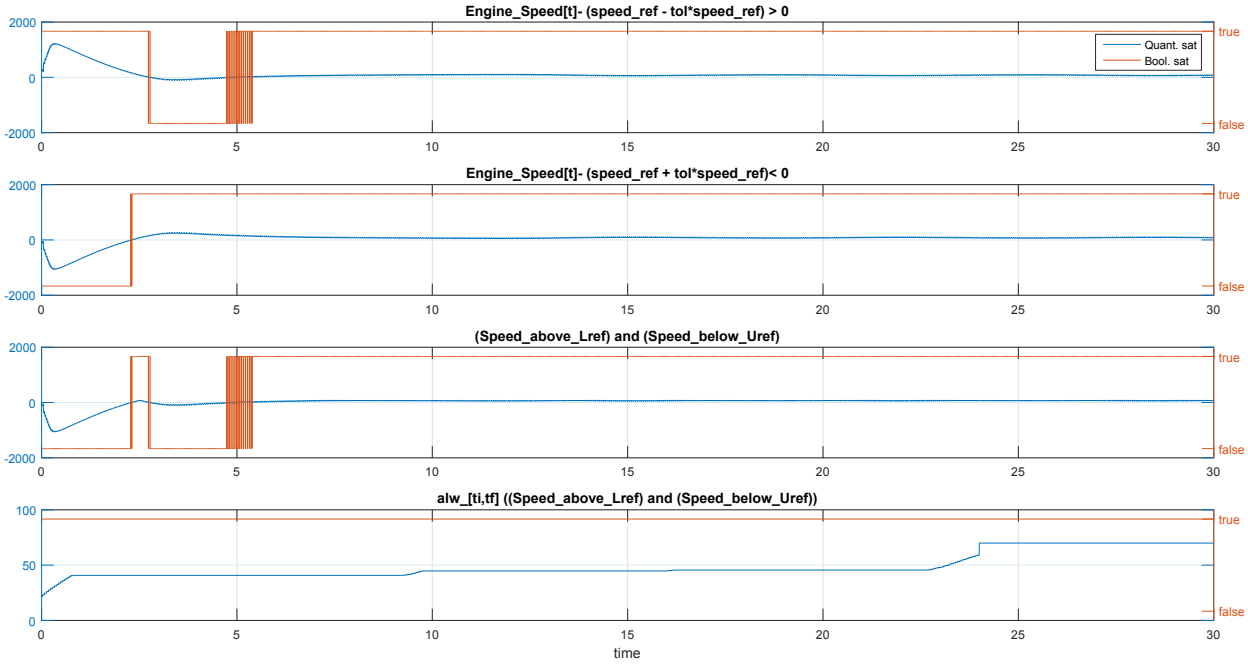

Figure 4: Formal verification of the idle speed control mode. The simulation results show that the engine is stable and the STL formula φ_1 has been satisfied.

generating 50 lower amplitude driver actuation inputs. The STL formulas φ_1, φ_2 and other timing properties could not be guaranteed anymore. These experiments were performed on MiL and SiL process steps by deriving requirements from the HiL process step since the HiL involves the most coupling constraint among the three process steps.

6 CONCLUSIONS & FUTURE WORKS

In this paper, we addressed the coupling constraints introduced in the HiL process steps and translated real-time requirements onto the MiL and the SiL process steps using our combustion engine example; we categorized the nature of coupling in each of the process steps in XiL; we formalized the coupling properties as STL formulas and illustrated its formal verification using the Breach toolbox. We illustrated how XiL can address coupled problems and explained the need to validate MiL and SiL process steps including crucial aspects of reality. We identified the scope of improvements in our modeling and simulation through formal verification of developed coupling properties.

In this paper, we presented coupling properties which address critical aspects of real-time deadline fulfillment, event detection and synchronization. As our next task, we wish to extend our set of coupling properties and address complex timing properties on cylinder pressure and co-simulation of the engine controller variants and the plant model.

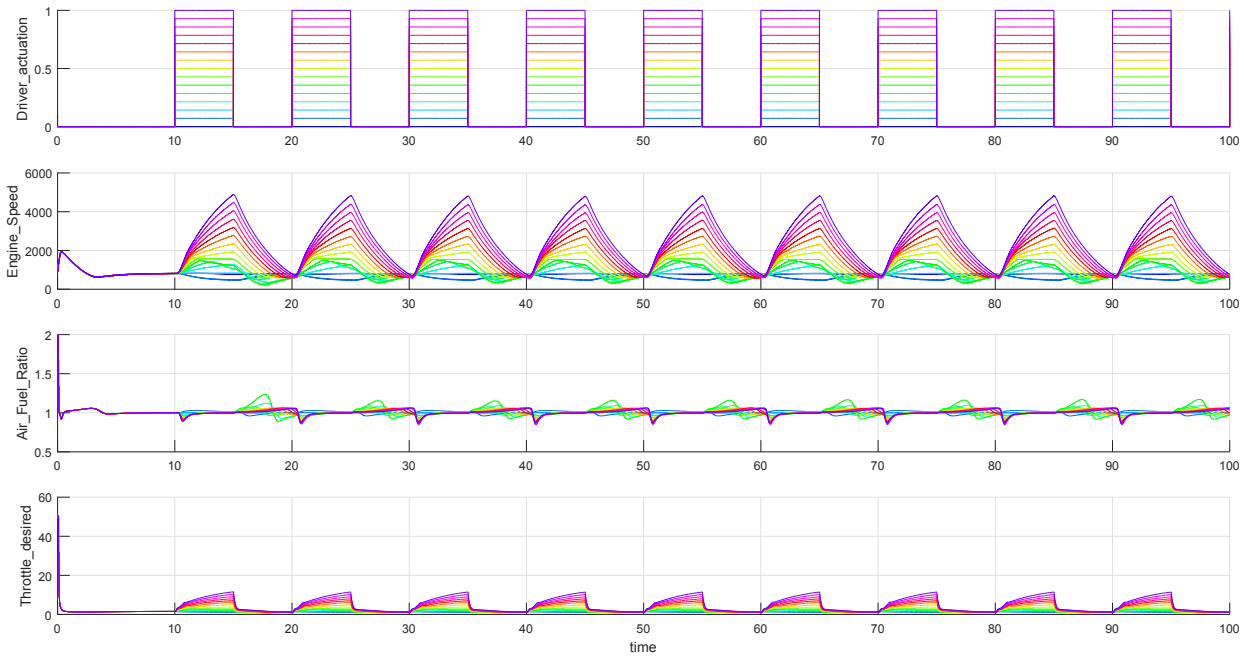

Figure 5: Simulation of random driver actuation inputs between 0% and 100% using Breach toolbox. The simulation results show that the engine is stable under these random input conditions.

REFERENCES

[1] Katyal, R. and S, S., *Virtualization for ECU Platform Software Testing in Automotive Embedded*, SAE Technical Paper 2011-01-1265, 2011, doi:10.4271/2011-01-1265

[2] ETAS GmbH, *Virtualization Is the Key to Greater Efficiency*, ETAS RealTimes Magazine, 2014, `https://www.etas.com/data/RealTimes_2014/rt_2014_1_52_en.pdf`

[3] Schäuffele, Jörg and Zurawka, Thomas, *Automotive Software Engineering - Grundlagen, Prozesse, Methoden und Werkzeuge effizient einsetzen*. ISBN 978-3-8348-03 64-1, Wiesbaden 2010.

[4] Tan, Gang, *A Collection of Well-Known Software Failures*. `http://www.cse.psu.edu/~gxt29/bug/softwarebug.html`, Penn State University, 2016.

[5] Alur, Rajeev, *Formal Verification of Hybrid Systems*. 978-1-4503-0714-7/11/10, EMSOFT 2011, Taipei, Taiwan, October 9-14, 2011.

[6] Donzé, Alexandre, *On Signal Temporal Logic*. EECS294-98, University of California, Berkeley, Spring 2014.

[7] Raman, Vasumathi et. al., *Model Predictive Control with Signal Temporal Logic Specifications*. 53rd IEEE Conference on Decision and Control, December 15-17, 2014, Los Angeles, California, USA

[8] Störmer, Christoph et. al., *ETAS LABCAR-XiL: Bridging the gap between development phases by harmonizing concepts and tools*. 15. Internationales Stuttgarter Symposium, Proceedings, DOI 10.1007/978-3-658-08844-6_30, Springer Fachmedien Wiesbaden 2015.

[9] Valasek, Michael, *Modeling, simulation and control of mechatronical systems*. Simulation techniques for Applied Dynamics, CISM Courses and Lectures, Vol. 507, ISBN 978-3-211-89547-4, Springer Wien, NewYork 2008.

[10] Ras, Jim and Cheng, Albert M.K., *On Formal Verification of Toyota's Electronic Throttle Controller*. 978-1-4244-9493-4/11, IEEE International Systems Conference 2011.

[11] Guzzella, Lino and Onder, C. H., *Introduction to Modeling and Control of Internal Combustion Engine Systems*. ISBN 978-3-642-10774-0, DOI 10.1007/978-3-642-10775-7, Springer-Verlag Berlin Heidelberg 2010.

[12] Villa, T. et. al., *Formal verification of an Automotive Engine Controller in Cutoff Mode*. 0-7803-4394-8/98, Proceedings of the 37th IEEE Conference on Decision & Control, Tampa, Florida, USA, December 1998.