A Mechatronic CAN-Based Functional Design and Verification Unified Approach

Mohamed Shedeed*, M. Galal Elshafey*, Mohamed Sobh, Sherif Hammad***

**Mentor Graphics Egypt, 78 El-Nozha St., Heliopolis, Cairo 11361, Egypt.*
mohamed_shedeed@mentor.com, mohamed_galal@mentor.com, sherif_hammad@mentor.com
***Ain Shams University, 1 Elsarayat St, Abbaseya, Cairo, Egypt.*
Mohamed.sobh@eng.asu.edu.eg,

Abstract: A mechatronic system needs an integrated design, implementation and verification unified approach due to multi-disciplinary interactive sub-system components. This paper presents a systematic methodology for a detailed migration from "model in the loop" (MIL), and "software in the loop" (SIL) to "hardware in the loop" (HIL) in order to full fill complicated mechatronic automotive system requirements. Break by wire anti-blocking car system model is implemented and simulated on Matlab™ in real time. For MIL, and SIL a virtual CAN bus channel is designed in order to communicate sensors and actuators signals to/from several electronic control units (ECUs). HIL is implemented to experiment the actual embedded controller performance over a real CAN bus with the real time simulated car model. Both virtual and real experimental results show the efficiency of the proposed approach.

Keywords: MIL, SIL, HIL, ECU, BBW, ABS, CNS.

1. INTRODUCTION

Mechatronic automotive system design, implementation, integration and verification represent an industrial research challenge [1, 2]. Integrating heterogeneous subsystems requires diversified tools and knowledge domains. Meanwhile, vehicle mechatronic systems are distributed on sensors, actuators, ECUs, and controller area network (CAN) bus communication. They become mandatory instrumentation to achieve more comfort and safety. This paper presents a unified approach to verify functionality of a mechatronic vehicle subsystems interaction.

Real time integration of testing, debugging and process simulation tools is necessary for software verification against mechatronic vehicle models [3, 4]. In [3], script–based test tool is integrated with Matlab™ [5] to perform real-time powertrain simulation. It made a good use of code generation feature in order to run tests on real targets. System Vision™ can be used to simulate car engine using VHDL/AMS language [6, 7]. They are fundamental tools that can be used in design and verification of developed software models against hardware and other components. Main challenge is applying early design tests and verification that could be reused in late implementation phases. Verification of requirements, system and software design is part of an iterative V-Model [8] development process shown in Figure 1.

An architectural description of automotive virtual verification is given in [9, 10]. Using virtual prototyping concept, bus communication protocol can be verified to address safety, cost and real time issues [11, 12]. Real time mechatronic car simulation is essential to emphasize on practical implementation of control algorithms [13]. It is a step towards virtual prototyping [14].

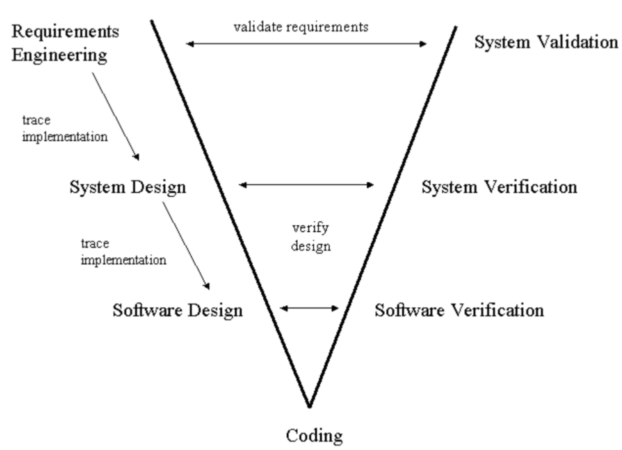

Figure 1. Formal V-Model

This paper proposes an experimented functional verification unified approach that moves seamlessly, along the V-Model, from a step to another.

The paper is organized as follows. Section 2 defines the problem statement listing the needed features of verification environment. Sections 3, 4 and 5 describe simulation levels. Section 6 proposes a new functional verification environment. In Section 7, Brake-By-Wire use case is described while real time experimental results are also presented. Finally, Section 9 concludes the paper.

2. PROBLEM STATEMENT

A generic environment is needed to support simulation in multiple levels, and in different stages in the development process. This environment shall support CAN based networks for interactions in all simulation levels whether by virtual simulation of the CAN protocol or actual communication with other CAN-based nodes. To cover this requirement, it is needed to support PC to target communication through CAN bus. Environment shall also provide an easy to use GUI to visualize the simulation results so that offers better analysis for simulation outputs. Different simulation levels, of the verification environment should support these requirements.

3. MODEL IN THE LOOP SIMULATION (MIL)

Physical or behavioural models of the car, mechatronic sub-systems and environment, are the starting point for all verification steps. Software control algorithm, which is subject to verification, is also modelled as a black-box. Matlab™ [5] is an appropriate tool for the required modelling and simulation at this level. Bus communication, messages/frames coding and decoding are behind the scope of this simulation level. It is only required to verify control design approach and performance. Real time simulation is important just for the coming verification stages. **Figure 2** summarizes MIL based simulation.

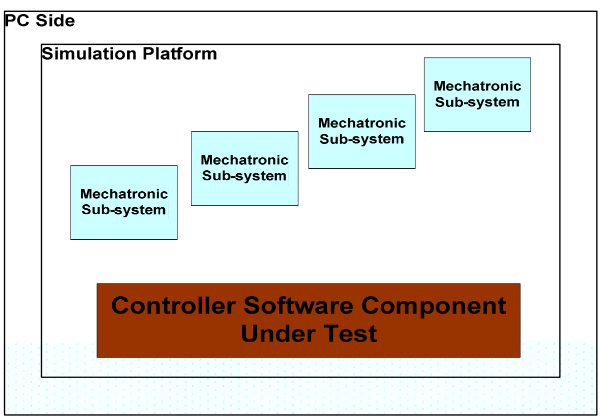

Figure 2. MIL Simulation

4. SOFTWARE IN THE LOOP SIMULATION (SIL)

After verifying the correctness of the model and also proving correct integration with other sub-systems, the coding process starts. It is now important to verify that the generated/written code against its high-level abstracted model. SIL environment provides a verification approach for coding correctness by replacing the module simulated in MIL with the implemented code. Virtual CAN bus communication and coding/decoding can be verified at this stage. Figure 3 shows SIL based verification.

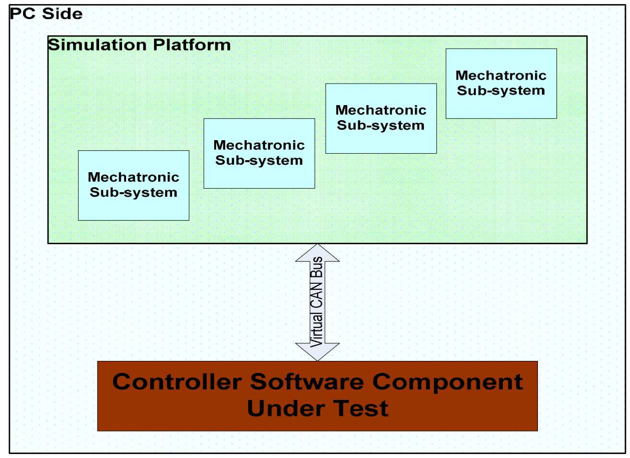

Figure 3. SIL Simulation

5. HARDWARE IN THE LOOP SIMULATION (HIL)

On-target implementation should consider communication software stack along with real time CNS (Control Networking System) requirements. HIL verification implies integrating ECUs with the process and environment models. CNS includes multiple ECUs that handle communication between controllers, sensors and actuators. This phase can be done as two sequential activities. Typically, CAN acquisition box is introduced to enable the communication between the vehicle model and different ECUs. By the end of this verification phase, the control algorithm is verified and ready to be applied on the real parts of the targeted vehicle. Bus loading, scheduling, and real time aspects are emphasized at this stage. Figure 4 highlights verification on a real CAN bus.

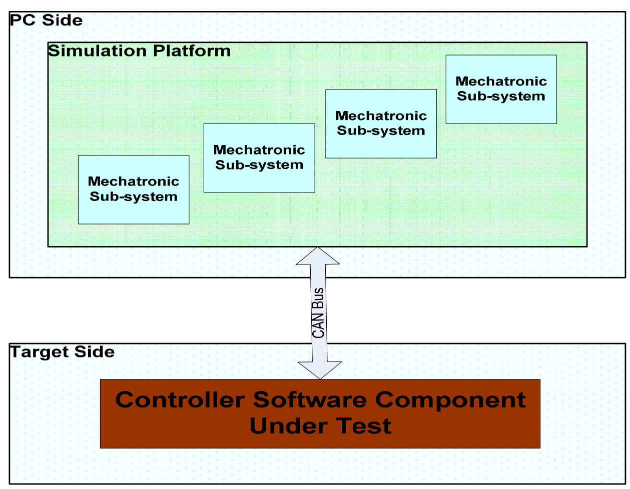

Figure 4. HIL Simulation

6. PROPOSED VERIFICATION ENVIRONMENT

The proposed environment consists of two categories; PC-side and target side modules. Figure 5 details the different components of the environment.