Real-Time Simulation of Hybrid Electric Vehicle Efficiency using Functional Reactive Programming

Volker Schwarzer
Department of Computer Science
University of Applied Science, Worms
Worms, Germany
email: schwarzer@hs-worms.de

***Abstract*—An idea is proposed to compute the overall powertrain efficiency of a Hybrid Electric Vehicle (HEV) in real-time using the reactive programming paradigm. Multi-domain vehicle models typically depend on the availability of a multitude of real-time sensor data within a short timeframe (usually one discrete timestep). While using a fully imperative simulation approach is fast and efficient, it is not suitable when required data is not provided within fixed time intervals. Reactive programming would solve this problem by asynchronously responding to real-time data streams instead of actively being triggered by function calls. The advantage of the proposed idea would be a fast and stable simulation process and the ability to partially compute system states as soon as the required corresponding data is available.**

Keywords - Real-Time Vehicle Simulation; HEV Efficiency; Connected Powertrain; Reactive Programming.

I. INTRODUCTION

The efficient hybridization of powertrains presents a complex challenge to car manufacturers. The challenge ranges from early design decisions, such as determining powertrain component sizing, to late-stage decisions, such as the optimization of the energy management. Overall powertrain efficiency of Hybrid Electric Vehicles (HEVs) frequently does not meet expectations due to time constraints in the automotive development cycle and dependencies on shared modular design requirements within the fleet and third-party components.

While suboptimal powertrain efficiency can easily be identified by overall fuel consumption, a precise quantification and determination of drivetrain losses is not straightforward. Multi-domain and multi-physics modeling of electric and hybrid vehicles is a simulation technique to compare powertrain efficiencies for standard drive cycles to test bench results [1][2]. Efficiencies can then be computed by aligning and evaluating the obtained data offline. However, the multi-domain complexity of the underlying stochastic distribution significantly impedes simple correlations and the identification of potential improvements. Furthermore, standard drive cycles do not depict real-life driving conditions with all possible efficiency scenarios.The core idea of the proposed research is that human pattern-recognition could overcome the challenges described above. Instead of comparing significant amounts of offline test bench data with simulation data, it is intended to utilize a real-time, multi-domain HEV model based on live On-Board Diagnostics (OBD) and additional sensor data. Overall powertrain efficiency would be computed and displayed in near real-time. An HEV operator, when noticing below-average powertrain efficiency during certain driving maneuvers and/or vehicle states, could arbitrarily recreate similar states to generate more relevant data and isolate the decisive events.

A well-performing efficiency simulation could certainly be achieved using imperative programming techniques. However, the reactive programming paradigm is a better choice when considering the limited availabilty due to the synchronous nature of most OBD-communication protocols. A deterministic, discrete approach requires all available sensor data to be provided within fixed time intervals. Reactive programming solves this problem by asynchronously responding to real-time data streams instead of actively being triggered by function calls. Fast and stable simulation processes, as well as the ability to partially compute system states can be achieved as soon as the required corresponding data is available.

The rest of this paper is organized as follows. Section II introduces functional reactive programming. Section III describes the proposed reactive simulation methodology. The conclusion closes the article.

II. FUNCTIONAL REACTIVE PROGRAMMING

Functional reactive programming is a declarative paradigm that allows users the benefits of functional programming languages and an easy interface to a reactive environment [3]. It is based on data streams, so called observables, and their propagation of change, which is achieved by observers using various functional operators. While purely functional programming languages are available (e.g., Haskell), the reactive programming paradigm needs further software tools to be fully integrable in a simulation code base. ReactiveX (also known as Reactive Extensions) is a software tool kit for asynchronous programming with observable streams. It is based on the ideas of the observer pattern and the functional programming paradigm. It can be utilized to program various imperative languages (e.g., Java, Scale, C#, C++, Clojure, JavaScript and Python) reactively and is therefore integrable into various simulation frameworks. Observables are rendered as time-series of events. The observation of a stream is called subscribing. Functions are defined as

 ISBN: 978-1-68558-001-8

observers and the observables are the subjects being observed.

III. Reactive HEV Efficiency Simulation

Figure 1 visualizes the proposed real-time, reactive simulation methodology. Powertrain sensor data is collected and published to a central Message Queuing Telemetry Transport (MQTT) broker. A client subscribes to all relevant topics and publishes each topic as a separate observable. The multi-domain powertrain model is composed of a collection of observers. Each observer subscribes to the observables needed to compute state variables using a set of operators. Since multiple required observable events will not necessarily occur synchronously, a lifetime parameter must be introduced. Events will last for a predetermined amount of time depending on the physical properties of the variables, which they represent. If all observable events (that an observer is subscribed to) overlap, the observer will react and propagate the data utilizing its operators. The resulting data stream will then be forwarded to a MQTT client, which will publish the data to the main MQTT broker. Eventually, the data stream will be transformed into another observable to which other observers can subscribe to.

To simulate the efficiency of a connected HEV powertrain in real-time, each of the observers would compute a specific state variable [4]. The closed feedback loop enables a nested component structure. Hence, a top-down modelling approach can be chosen, and simulation granularity could be increased over time. The higher the granularity, the more precise the outcome of the simulation.

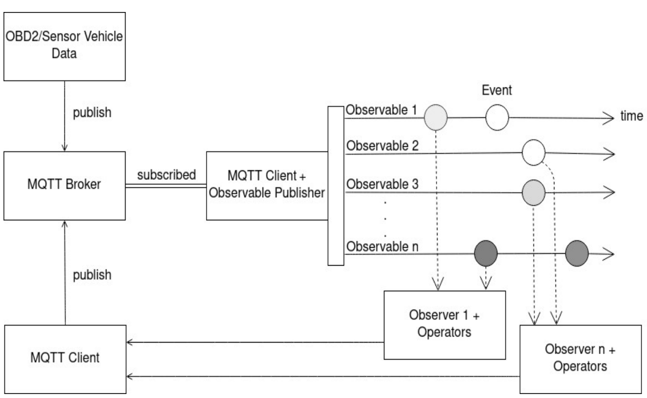

Figure 1. Proposed real-time HEV efficiency simulation engine using MQTT as a transport protocol. Computation of state variables in a closed feedback loop. All system variables are modelled as data streams (observables).

IV. Conclusion

A reactive simulation approach for HEV powertrain efficiency has been proposed. The main advantage of the discussed idea could be a fast and stable simulation process and the ability to partially compute system states as soon as the required corresponding data is available. Even if sensor data is faulty or temporarily not available, the simulation could still be capable of computing and publishing certain state variables and therefore approximate powertrain efficiency.

References

[1] D. W. Gao, C. Mi, and A. Emadi, “Modeling and Simulation of Electric and Hybrid Vehicles,” in Proceedings of the IEEE, vol. 95, no. 4, pp. 729–745, April 2007.

[2] V. Rallabandi, D. Lawhorn, D. M. Ionel, and X. Li, “Multi-Physics Modeling for Electric and Hybrid Vehicles with In-Wheel Electric Motors,” IEEE Transportation Electrification Conference and Expo (ITEC), pp. 146-151, 2018, doi: 10.1109/ITEC.2018.8450091.

[3] B. Finkbeiner, F. Klein, R. Piskac, and M. Santolucito, “Vehicle Platooning Simulations with Functional Reactive Programming,” Proceedings of the 1st International Workshop on Safe Control of Connected and Autonomous Vehicles, SCAV@CPSWeek, pp. 43-47, 2017, doi: 10.1145/3055378.3055385.

[4] F. Boussinot, B. Monasse, and J. Susini, “Reactive Programming of Simulations in Physics”, International Journal of Modern Physics C, vol. 26, no. 12, pp. 1-16, 2014, doi: 10.1142/S0129183115501326.