49. Internationales Wissenschaftliches Kolloquium
Technische Universität Ilmenau
27.-30. September 2004

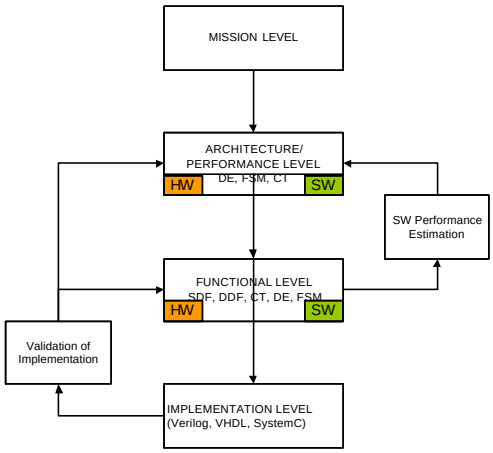
Horst Salzwedel / Matthias Zens

DEVELOPMENT OF EMBEDDED AUTOMOTIVE ELECTRONICS AT ARCHITECTURAL/PERFORMANCE LEVEL

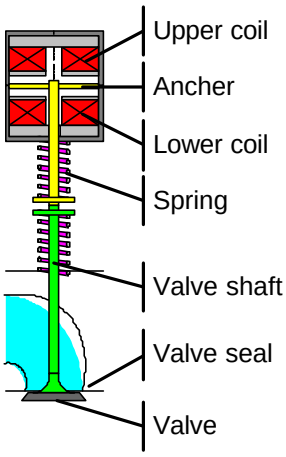
INTRODUCTION

During research on solving multidisciplinary problems in flight control design [1], functional level design tools like Ctrl-C®, MatriX® and later Matlab® were developed. These generic tools permitted to easily combine models from different engineering disciplines and e.g. permitted to improve the accuracy of navigation alignment filters by more than a factor 100 and reduced the alignment time by more than a factor 100 [2]. In the mid 1980th these design tools were adopted for functional level design processes in automotive industry that are being used today. In the meantime, the complexity of electronics increased by nearly a factor 10000. Today's automobiles include networked embedded systems with up to 100 electronic control units (ECUs) and cooperating hardware and software components. Since functional level design does not permit to simulate the overall system and the interaction of hardware and software, it does not provide validated executible specifications for the sub system designer. Neither the subsystems nor the overall system can be sufficiently validated. Subsystems like ECUs exhibit failure rates up to two orders of magnitude and more above those desired. Despite exponentially increasing validation cost, large scale recalls of automobiles accelerate because of problems with electronics, causing billions of € of losses for the automotive industry.

During the last 10 years, the Mission Level Design flow [3] and Mission Level Design software [4] have been developed, that integrate the design flow for hardware and software from mission level to implementation. In this design methodology, most design decisions are done at the architectural/performance level early in the design process, permitting to to include requirements of the overall system and the ineraction of hardware and software. Development times for hardware and software were reduced by up to a factor 10 and system reliability has been increased. During a diploma thesis at Infineon [5], a Mission



Level Design approach was investigated for the design of architectures for electronic valve applications.



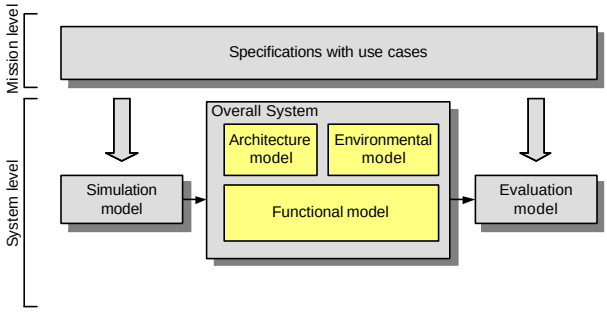
Camless engines using electromagnetic valve actuators have demonstrated fuel savings of up to 20% [6]. Despite the large amount of electronics required to implement electronic vales, they are not permitted to be more expensive than current generation valves, driven by camshafts. The reliability also must be as high as that of mechanical valves. Optimizing cost and reliability of electronics means optimizing it's architecture for the use cases, missions, of electromagnetic valve applications. This includes,

- Analysis of actuator models
- Development of control algorithms
- Analysis of sampling frequency, accuracy and fault tolerance
- Determination of required performance of processor and busses
- Choice of the hardware architecture and hardware components to minimize cost

Some of the models and strategies developed during this investigation are shown below.

MODELING STRATEGY

In order to achieve confidence in the design process, it is embedded in a validation environment, consisting of exchangible components for architectural models, environmental models and functional models of hardware and software. Use cases are driving simulation model and evaluation model, in order to asure that all design decisions are validated against the use or mission of the overall system.



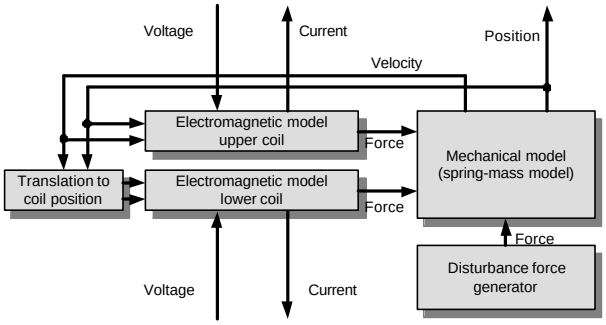
Architecture, functional and environmental components of the model are mapped into models of different execution domains of the software system MLDesigner[4]:

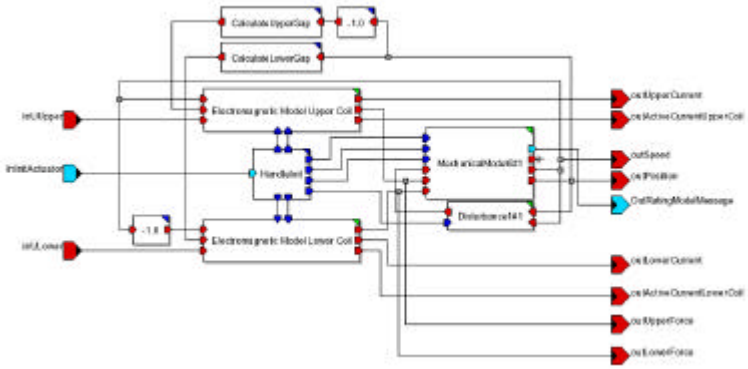
- Architectural components like resource contention of CPU and bus, network topology, protocols of traffic flow, arbitration, buffers, RTOS and software execution are modeled in the decrete event domain (DE)
- Sychronous and Dynamic Data Flow (SDF, DDF) are used to model functional behavior and algorithms

- The Finite State Machine (FSM) is used to model software, controllers and protocols
- The Continuous Time Discrete Event Domain (CTDE) is used to model analog components like the mechatronic valve

Because of the big difference between execution times for simulati ng architectural components and functional components, the strategy used here was to perform analysis and design as much as possible at the architectural/performance level and include functional components only when necessary. Decisions that can be made through performance evaluation are, optimization of the bus architecture, memory architecture, and transport protocols. Results we get from these simulations are usage of resources, delays, and throughput, permitting to perform hardware/software partitioning and sizing of resources like busses, CPUs and memories. A resource instruction model was developed to map functions into architectures.

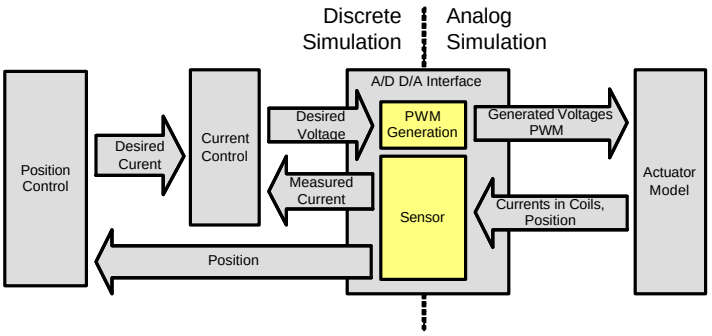
MODELING





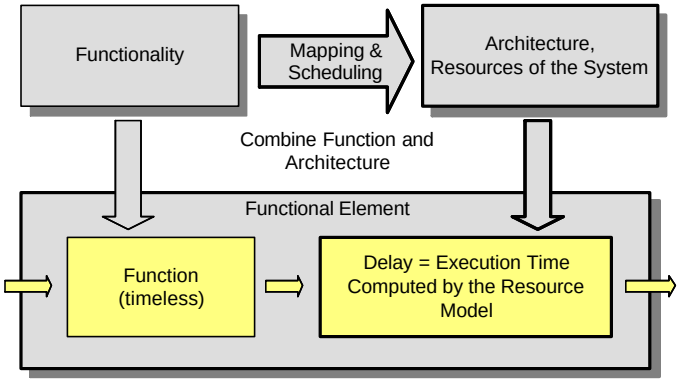
The electromechanically actuated valve includes upper and lower magnetic coils, that move an anchor plate up and down. The anchor plate has a pre-set tension by the spring. A spring also surrounds the valve, attached to the anchor shaft. The model therefore consists of an electromagnetic model and a mass-spring model. The spring-mass dynamics as well as the electromagnetic actuation of the valve are modeled in the CTDE domain of MLDesigner. The primitive *HandleInit* receives the initialization message and distributes the discrete events to the continuous components of the model. The module *Disturbance* belongs to the simulation model and inserts disturbance forces on the valve.

The functional model consists of discrete controllers and an analog section describing the physics.



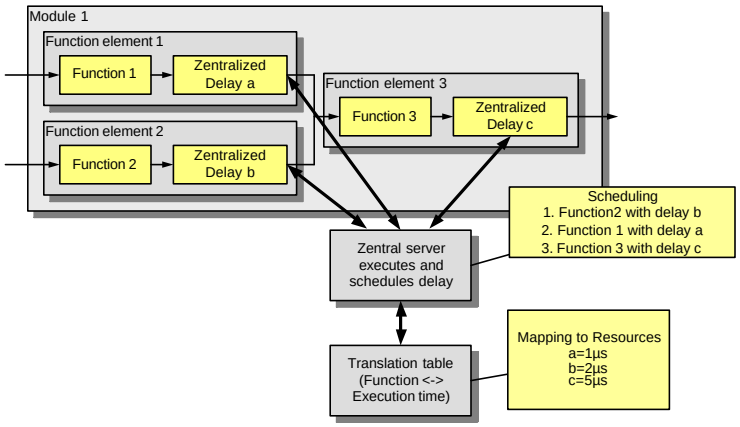
For the inner loop current control, a PI controller was developed. A Tustin transformation was used to convert the analog control law into the z domain, in order to maintain phase information. For the out loop position controller, again a PI controller was developed. Critical is that the controller achieves a “soft landing” of the valve. The anchor speed must be as close as possible to 0 *m/sec* when it reaches the upper coil. A hold current is activated in the upper coil well before the anchor reaches its upper limit. The speed of the anchor is controlled by managing the valve energy. Losses due to friction and electromagnetic disturbances are compensated for. Both controllers are synchronized and are executed at the same sampling rate. For the functional model, the algorithms are timeless and are activated only at discrete times. Therefore both the functional model, as well as the architectural model is implemented in the DE domain.

The functionality of a subcomponent of a system consists of the functional algorithm and the resources and time delays it needs for execution. The algorithm is mapped into a timeless function and into the architecture of the system and its resources.



INSTRUCTION RESOURCE MODEL

The critical element for the simulation of real time embedded systems is the Instruction Resource Model (IRM). It links (maps) the function of a data flow model to the resource that will execute it and schedules it, in order that it will execute at the right time. Execution requests of a data flow model are send to a central server, which



sends an achnoledgement to the function when the resource has executed the function. The elgorithmic execution of the function is executed in the function of the data flow graph itself. The IRM therefore performes the function of a real time operating system (RTOS).

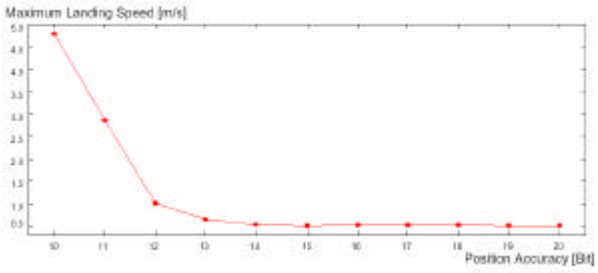
The execution times of the different functions may be determines from,

1. Known values of a previously use of the algorithm on the target processor
2. Software performance estimation from the C-code. In Ref. 7 an instruction set simulator is added to MLDesigner, which executes the compiled code at the desired optimization level and returnes the execution time to the event loop of the DE scheduler.
3. Execution of the VHDL code of the hardware
4. Measuring the execution time with a scope on the real hardware

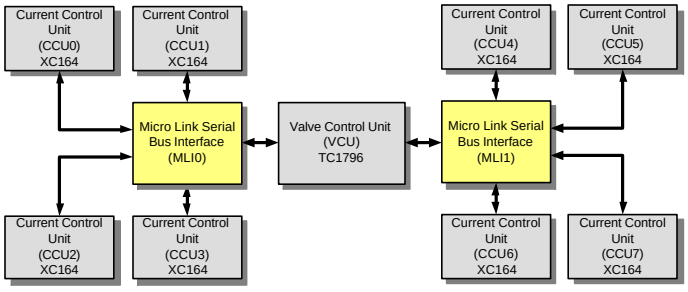
In this development, methods 3 and 4 have been used.

SIMULATION AND ANALYSIS

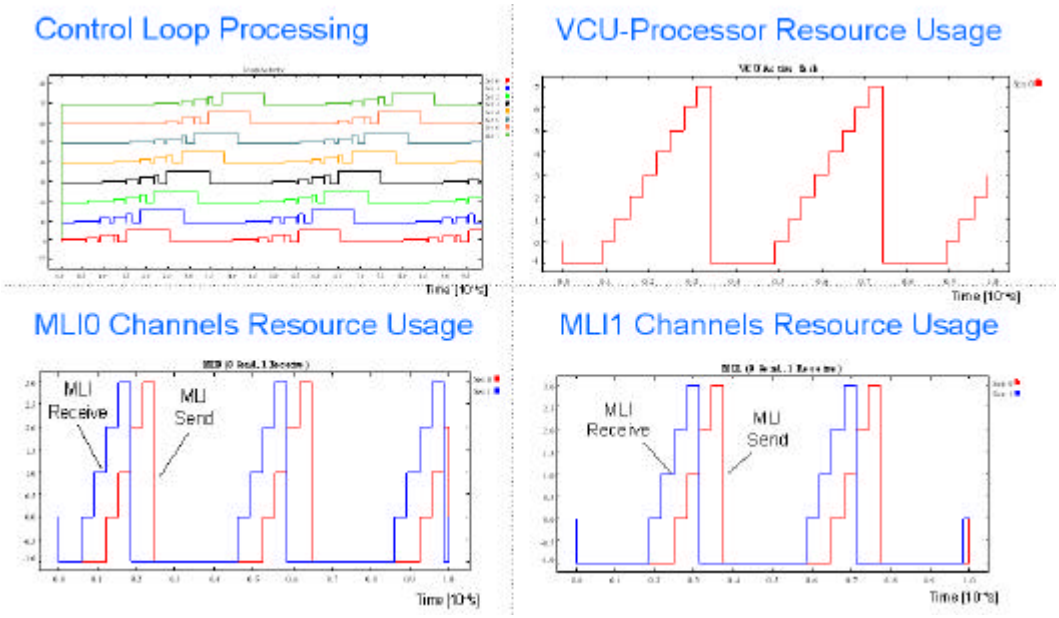
For the functional evaluation of the electronic valve control system, the effects of sampling time and time delays on the system were investigated. The main measure of acceptance was the maximum landing speed of the valve. The accuracy for current, position and PWM and ist impact on maximum landing speed were determined. For reliability analysis, the error rates of data transport on the busses were investigated. The graph shows that at position has to be measured to at least 14 bits.



For the architectural evaluation, the timing characteristics of the electronics are analyzed and validated. This included the analysis of all the control loops, determination of the level of usage of the resources, and analysis of the distribution of tasks of individual control loops over the resources of centralized and decentralized architectures. For decentralized partitioning, 1 Infineon XC164 processor was selected for voltage control, A/D, and D/A conversion of 1 valve, and an Infineon TC1796 processor for outer loop position control of 8 valves. The simulation was performed only with the architectural performance model since the functional evaluation was done before, significantly shortening the



simulation times. The simulation results of this architecture show the usage of the resources: control loop processing, VCU, MLI0 channel and MLI1 channel [5].



CONCLUSION

We have shown that complex automotive electronics can be developed, analyzed and optimized byan integrated simulation containing mission, architecture, and function of hardware and software. This methodology automatically validates the design against mission level requirements and significantly reduces requirements for redesign after test and hence reduces development time. The Instruction Resource Model can separate function from architecture and model RTOSs.

References:

[1] Horst Salzwedel, James H. Vincent Modeling, Identification and Control of Flexible Aircraft, AFWAL-TR-84-3032, 1984.
[2] Horst Salzwedel, R. Calhoun, and Lt. P. Murdock, Unbiased Transfer Alignment Filter Design for Air Launched Weapons, National Aerospace and Electronics Conference, Dayton, Ohio, May 1985.
[3] Gunar Schorcht, Design of Integrated Mobile Communicatio n Systems at Mission Level, Dissertation at Ilmenau Technical University, July 2000.
[4] MLDesigner® Manual v2.4, http://www.mldesigner.com
[5] Matthias Zens, Development of Methods for Analysis of Electron ic Valve Control Architectures (German), Diploma Thesis, Ilmenau Technical University, 2003, http://www-sst.theoinf.tu-ilmenau.de/studentsWorks/works/documents/2003/Diplom2003_MatthiasZens.pdf
[6] K. Manon McNair, Matthias Zens, Horst Salzwedel, System-Level Partitioning Using Mission-Level Design Tool for Electronic Valve Application, SAE World Congress, Detroit, March 2003.
[7] Thomas Lohfelder, Modeling Operating Systems in MLDesigner, Diploma Thesis, Ilmenau Technical University, 2004.

Authors:

Horst Salzwedel, Ilmenau Technical University, Helmholtzring 1, D-98693 Ilmenau
Phone: +49-3677-691316, Fax: +49-3677-691285, Email: horst.salzwedel@tu-ilmenau.de
Matthias Zens, Lauterbach Datentechnik GmbH, Fichtenstr. 27, D-85649 Hofolding
Phone : Tel ++49 8104 8943-166, Fax ++49 8104 8943-170, Email : matthias.zens@lauterbach.com