Educational Frameworks for Vehicle Mechatronics

Tamás Bécsi, *Member, IEEE,* Szilárd Aradi, *Member, IEEE,* and Péter Gáspár, *Member, IEEE*

***Abstract*—Teaching mechatronics is a complex challenge because of the multidisciplinary nature of the topic. On the other hand the requirements of the smart city or the smart mobility of the future demand complex solutions from next generation of engineers. The topics include electrical, mechanical engineering, control theory and ICT in which fields the students must deepen their knowledge. Tasks and solving problems in mechatronics require cognitive and operational knowledge and practical experience in systems design and analysis.**

There is a strong demand for project-based teaching using certain kinds of practical material. The paper presents two educational frameworks designed for students with a specialization in vehicle mechatronics. Both are aimed at the emulation of real vehicle behaviors, the first is on ECU (Electric control Unit) level while the second is on vehicle level.

I. Introduction and Motivation

Except for the recession in 2008 worldwide vehicle market has been growing steadily. According to the figures of The International Organization of Motor Vehicle Manufacturers (OICA) the vehicle industry requires the employment of more than eight million people directly in manufacturing the vehicles and the parts that are built into them. This accounts for over five percent of the world's total manufacturing employment [1]. Competition in the industry motivates the manufacturers to introduce new technologies, comfort and safety solutions resulting in the high employment level of their research and development departments.

Despite the continuous research activity in this field, the automotive industry seemed to have fallen behind the so-called "Connected Revolution" passenger cars could not follow the rapid growth of the ICT (Information and Communication Technology) sector. This lag can be attributed to several factors:

- Both the development and the life cycle of the vehicles are rather long.
- The standardization and migration of new solutions are difficult.
- The property and life safety issues and threats are greater than in other industries (e.g. mobile phones)

Despite these difficulties there is no doubt that the vehicle of the future will be the integral part of global communication. Without this, smart city and smart mobility is unimaginable. Communication, intelligence and cooperative techniques and systems will characterise the cars of the future.

T. Bécsi and Sz. Aradi is with the Department of Control for Transportation and Vehicle Systems, Budapest University of Technology and Economics, Budapest, Hungary, e-mail: becsi.tamas@mail.bme.hu; aradi.szilard@mail.bme.hu .

P. Gáspár is with the Systems and Control Laboratory, Computer and Automation Research Institute, Hungarian Academy of Sciences, Budapest, Hungary Email: gaspar.peter@sztaki.mta.hu

Therefore there is a strong demand for well-trained vehicle engineers on the labor market. These engineers need diverse knowledge in the field of electronics, mechanics, hardware and software design, and also in industry-related technologies, such as standardised intra-vehicle communication. Understanding these needs the Department of Control for Transportation and Vehicle Systems Budapest University of Technology and Economics introduced a specialization programme covering most of these fields.

Teaching mechatronics is a great challenge since the subject is a multidisciplinary area of electrical, mechanical and IT knowledge. Students must deepen their knowledge in these slightly related fields of engineering.

TABLE I. Subjects of Mechatronics Engineering [2]

Mechanical Engineering	
Design and Production	System Dynamics
Control Theory	
Control System Design	Real-Time Systems
Electric/Electronic Engineering	
Sensors	Actuators
Computer Science and Technology	
Algorithm Application	AI and Communication

Tasks and solving problems in mechatronics require cognitive and operational knowledge and practical experience in systems design and analysis [3]. All mechatronics-related curricular papers state that besides the formal teaching of basic techniques, such as sensors and actuators, software and hardware design or dynamics, certain kinds of practical material is needed in this field [4] [5] [6]. There is a clear consensus amongst the stakeholders in education and industry about the strong need for project-based teaching in the education of mechatronic systems [7] [8] [9]. Although methodologies may differ a university student project only simulates the real industrial development conditions. However, giving individual or group assignments to students with a related problem to solve is the best practice to acquire skills in this field. Didactic questions, such as what kind of hierarchical or development structure should they use, emerge during the organization of the students into groups. Grimheden [10] proposed the agile method while Scrum [11] proposed other classic development methods such as the V-model, see also [12] [13].

The paper presents two educational frameworks designed for students with a specialization in vehicle mechatronics. Both are aimed at the emulation of real vehicle behaviors, the first is on ECU level while the second is on vehicle level.

The paper is organized as follows: The next section introduces an educational framework for automotive ECU design, while Section III presents an autonomous vehicle framework and finally Section IV introduces a Hardware-in-the-Loop environment for designing and testing multiple solutions.

II. The ECU Design framework

The first presented framework serves the purpose of aiding the students throughout a complex, yet simplified functional development of an electronic control unit of a vehicle.

The goal is for the students to understand and become familiar with the basics of the development process of an automotive ECU. For this purpose each student is assigned an individual project which represents the - naturally limited - functionality of an embedded controller of a vehicle. The specification of this task describes the sensors and actuators, analogue and digital I/Os and also the communication roles of the unit.

Based on this specification students must analyse the problem and design a system capable of fulfilling these tasks.

In order to simulate the real-world development process as much as possible several aspects have been taken into consideration. The areas and skills involved in the student project are:

- Understanding and analysing functional specifications;
- Knowledge of standard vehicular network description formats, such as *DBC* (Communication Database for CAN) or *LDF* (LIN description file);
- CAN/LIN/FlexRay communication via a specific microcontroller;
- Software and algorithm testing;
- Failure mode and effect analysis and error handling;
- Test case design;
- Documentation.

Testing is necessary during the development process. In some cases it could be feasible to provide the specific vehicle part as a test bed for the controllers. Furthermore in an emulated environment the modelling of erroneous behaviour and failure modes is much easier.

These are the main reasons why the decision to develop a complex framework providing all functionalities of the modules to be driven has been made. The paper presents the hardware and software-related solutions of this emulation environment.

A. Design tasks

Each student task represents a specific vehicle ECU with a well-defined task. They share the feature of CAN communication, and each has to interact with at least one of the other units by providing status information, or sending/receiving commands. The ECU functionalities involved in the tasks are the following [14]:

1) Electric Window Lift (EWL): The EWL module must control an electric window of a vehicle with two direction inputs and a proper PWM signal. With a short (less than half a second) input it must drive the window to the end position, otherwise it drives the window until the input button is pressed. Failures may occur when the limit switches are malfunctioning: the window becomes stuck or the drive motor fails.
2) Data Acquisition Unit (DAU)
3) Trip Computer Unit (TCU): The TCU is a simple display module for the driver, which collects information from other ECUs via CAN and calculates average speed and trip distance on a daily and full lifetime basis. The display shows warning on opened doors.
4) Blindspot Assistant (BSA): The BSA must implement simple functionalities of a blindspot assistant. The ECU gives warning based on the measured side distances of objects, the steering wheel angle, turn signal state and vehicle speed. Some information is gathered through the I/Os, and other is received via CAN communication. Based on the severity level, the ECU actuates warning LEDs, or shakes the steering wheel.
5) Turn Signal (TS): The TS unit drives four turn signal lamps of the vehicle with the information provided by their control switches and also the switch of the hazard warning. The ECU also considers the angle of the steering wheel, which is provided as a simple analogue input. The system must monitor the state of the lamps and malfunctions can be emulated in the framework.
6) Headlight Control (HLC): The HLC implements the functionalities of a headlight control unit, where the standard low or full-beam functions are driven by digital IOs, although cornering light is controlled by information acquired via the CAN bus. For error handling purposes the measurement of lamp states are given as a feedback.
7) Central Lock (CL): The CL unit controls the central locking system of the vehicle based on the state of the doors and the control inputs.
8) Electronic Cruise Control (ECC): The ECC module implements the Electric Cruise Control function, with the driver inputs of on/off and acceleration/coast. It also uses the information of the wheel speed sensor. The ECC gives the throttle command via CAN, and the emulation framework simulates simple vehicle dynamics and external forces such as wind or inclinations.

B. Framework Overview

In order to design such framework the behaviour of several subsystems must be determined to adequately simulate the working environment of the control unit. These subsystems and as well as their relations are presented in Fig.1.

The core of the emulation is the dynamics, mechanical and electrical behaviour. In addition, error injection capabilities are also necessary. These functionalities may differ depending on the individual task.

Connection to the simulated environment is achieved by the behavioural model of the sensors and actuators. The dynamics of these subsystems must be separated from the core, and in this way separate sensor/actuator failure and degraded modes can be also emulated.

CAN communication is integrated into the framework which serves the following purposes. There are design tasks whose main functionality is operated via CAN and there are some which only broadcast state messages to the vehicular network. The complete system can only be operated without emulated CAN messages if all tasks have been completed by all students. In the individual development phase a student with a

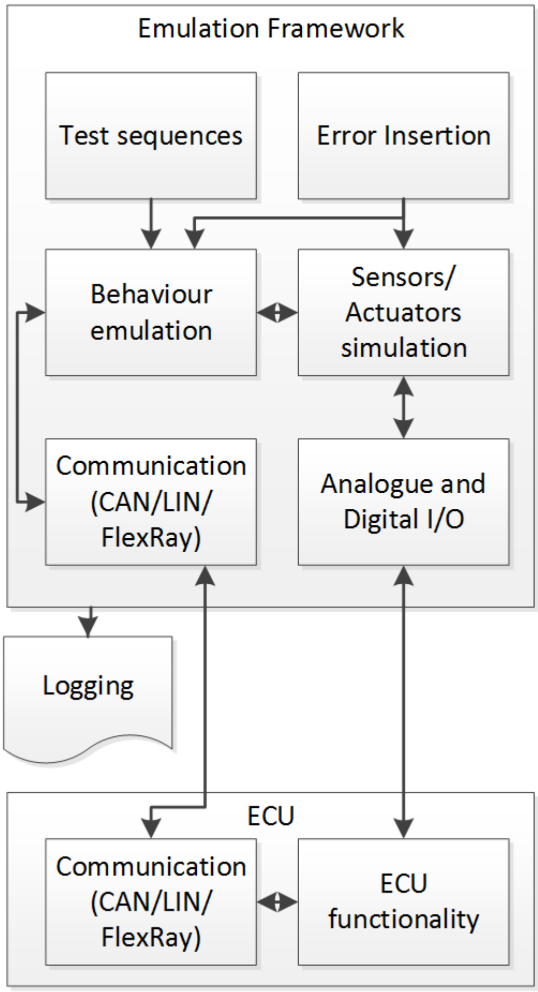

Fig. 1. Framework overview

specific ECU needs the emulated CAN communication of the connected ECUs designed by other students.

In most cases the primary functionality of the given ECU is simple and its implementation is trivial. However, when the unit must detect, log and handle errors originating in either behaviour dynamics or the sensors, even the simplest function could become a complex problem. The possibility of error injection gives the opportunity for the students to examine the failure modes since their effects must be evaluated and an adequate action plan must be outlined with different states of normal, degraded, or erroneous operation.

The emulation environment has two operational states. One is manual operation, where all interactions and system state modifications are carried out manually in the control window of the system. In most cases and during the initial development process this feature is sufficient for the students for testing purposes, although the handling of complex, sequential and repeatable system states or error situations are inconvenient. For this case the playback of previously designed test sequences is possible. An automatic logging feature also helps the design process and it is also useful at the final evaluation of the project.

C. The educational vehicle network

As it was mentioned before, all control units have the requirement to communicate via CAN network. One type of message that must be provided by the units is the ”state message”, sending all signals and operational states periodically. The interaction between diferent units is done through previously defined ”command” messages where one ECU directs an action of an other control unit. In this way the individual projects are connected in an imaginary vehicle network on a shared CAN bus, which is extended with an additional supervisor device as shown in Fig.2.

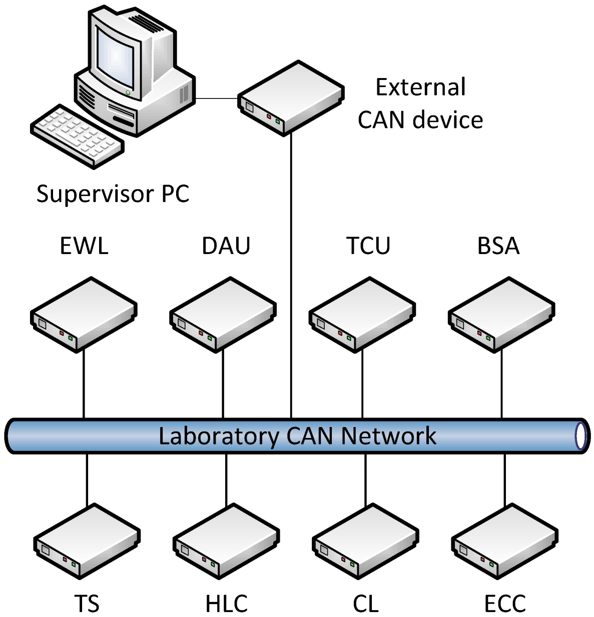

Fig. 2. Laboratory vehicular CAN network

This common CAN network ensures the cross communication of the units just as in any commercial vehicle. The simple illustration of this behaviour is, when the CL ECU receives a ”*lock doors*” command from its operator, it sends out a message to the EWL to close windows and also notifies the TS to blink two with all turn signals. If the locking of doors can not be completed it sends a single-blink command to the TS instead. Table II shows the cross-interaction chart of the control units.

The dbc file which describes the signals and messages is given to the students along with the specification.

D. Hardware Framework and Implementation

The framework presented is independent of the platform used by the students during the ECU design process since it

TABLE II. COMMUNICATION CHART OF THE LABORATORY NETWORK (RST: READ STATUS; CMD: GET COMMAND)

	BSA	CL	DAU	ECC	EWL	HLC	TCU	TS
BSA	X			RST		RST		RST
CL		X						
DAU			X				CMD	
ECC				X				
EWL		CMD			X			
HLC				RST		X		
TCU		RST	RST	RST			X	
TS		CMD						X

interacts through analogue and digital IOs and standard CAN interfaces. Since the controller education at our department uses the Atmel AVR family the primary platform of the ECU design is the same. The students receive a BIGAVR6 (see Fig.3.) development board with a AT90CAN128 [15] microcontroller to prepare their project. The advantage of this is that it is a full-feature development board for AVR controllers [16] having all necessary analogue and digital I/Os, displays, an onboard programmer and also a CAN transceiver.

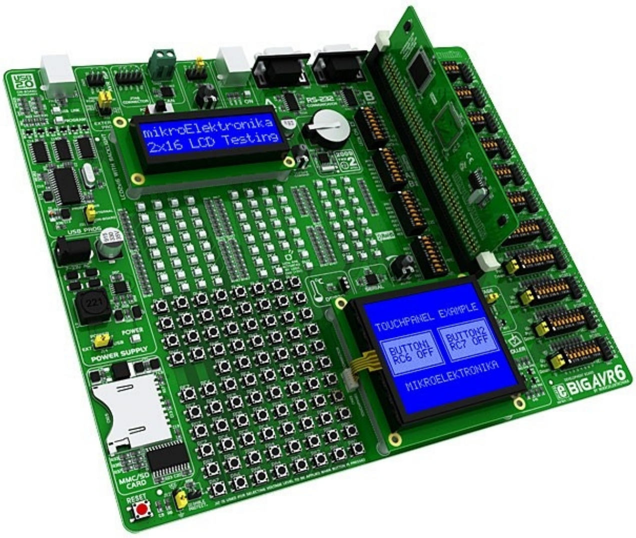

Fig. 3. BigAVR development board

A general purpose CAN interface with USB connectivity called WeCAN USB [17] is used for communication.

The framework utilizes NI myDAQ, a low-cost portable data acquisition (DAQ) device as its analogue and digital I/O interface (see Fig. 4.). It comes with two analogue inputs and outputs and also 8 digital I/O channels. [18]

E. Example: the EWL unit

The electric window functionality in the imaginary vehicle is realized by the EWL unit. In the scenario provided by the emulation framework an electric motor litfs the window as a response to two different inputs: one digital signal for direction and a PWM signal for speed command. Two buttons provide passenger input while two limit switches sense the end position

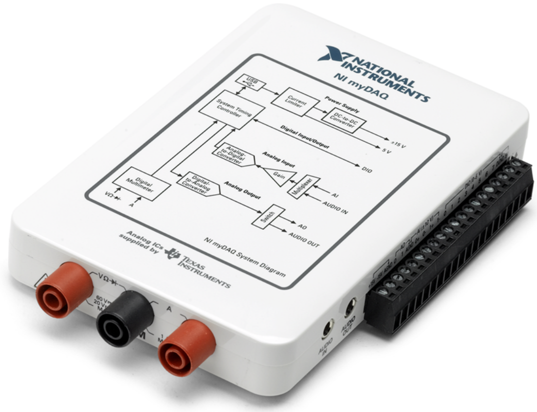

Fig. 4. National Instruments MyDAQ

alongside with a redundant analogue sensor. The current of the electric motor is also provided as an analogue signal.

Failure modes of the emulated environment may include the following: short circuit may occur in the motor while increased current can be measured when the windows is stuck. Also the limit switches can fail in both ways. The examination of the system is necessary by the students and they must identify the possible redundancies and error handling solutions. An example screen-shot of the emulation software of the EWL unit can be seen in Fig. 5.

The EWL unit must handle two CAN messages according to the provided dbc. The state message with the identifier "MSG_ST_EWL" provides the position of the two buttons, the two end position sensors, the analogue position, operability information and motor current. The unit understands the message called "MSG_CMD_EWL" coming from other ECUs that command the window to a specific position.

The provided dbc formally describes the two messages as:

BO_ 2149187380 MSG_ST_EWL: 8 EWL_ECU
SG_ St_Wnd_Sw_Up : 27—1@1+ (1,0) [0—0]
SG_ St_Wnd_Sw_Down : 26—1@1+ (1,0) [0—0]
SG_ St_Wnd_EP_Low : 25—1@1+ (1,0) [0—0]
SG_ St_Wnd_EP_High : 24—1@1+ (1,0) [0—0]
SG_ St_Wnd_Position : 16—8@1+ (0.4,0) [0—102] "%"
SG_ St_Wnd_Operability : 8—8@1+ (1,0) [0—0]
SG_ St_Wnd_Current : 0—8@1+ (0.04,0) [0—0] "A"

BO_ 2148138804 MSG_CMD_EWL: 8 CL_ECU
SG_ Cmd_Wnd_Pos : 0—8@1+ (0.4,0) [0—100]

III. EXPERIMENTAL VEHICLE FRAMEWORK

A. The Vehicle Frame

The second presented educational framework is a vehicle for implementing and testing advanced driver assistance and

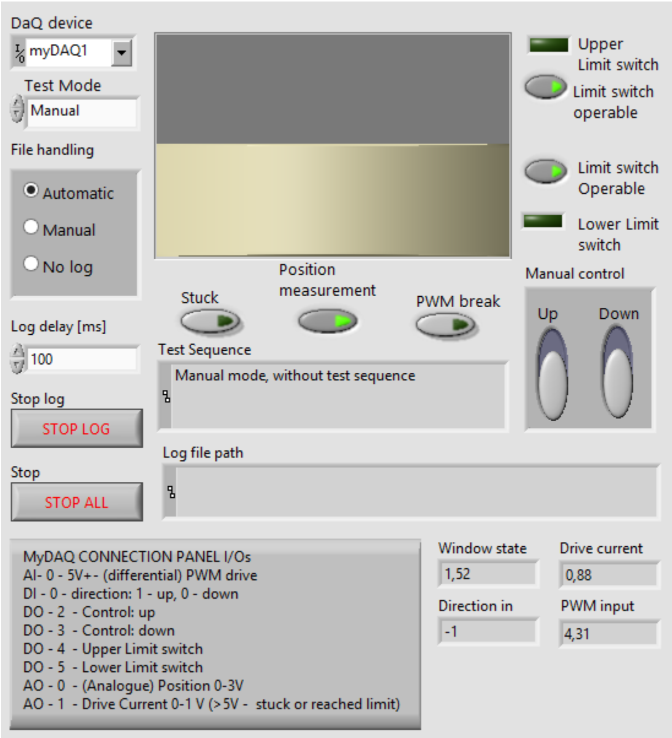

Fig. 5. The EWL emulator

autonomous driving functions. Hence, the purpose of the development was not the building of a real 1:1 scale vehicle, which would raise several questions and needless extra design problems. For the purpose of rapid framework development a much simpler vehicle basis has been chosen: a simple commercial go-cart frame (see Fig. 6.) with hydraulic disc brakes, rigid rear axles and a chain drive.

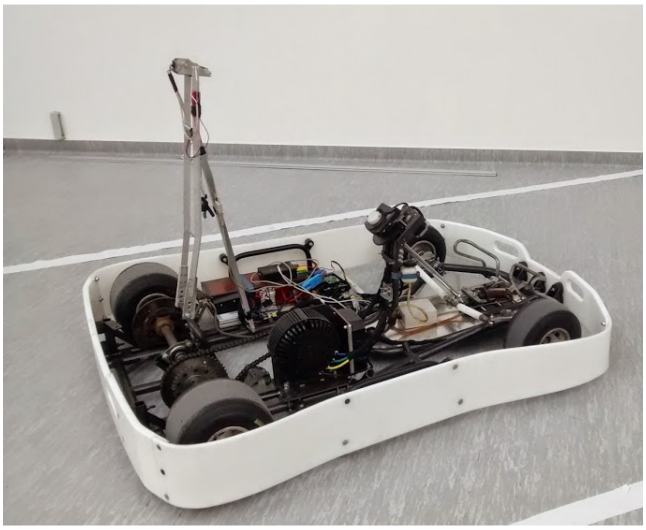

Fig. 6. The modified go-cart frame used for the project

B. Sensors

Autonomous vehicles rely on their ability to sense the environment, which is a key feature in achieving better performance and functionality. These systems generally use ultrasonic sensors, radars and different vision systems, although information provided by these sensors is limited. With LIDAR systems one can receive detailed depth-of-field maps, which is the reason why it has gained popularity amongst the self-driving car projects [19][20]. Besides, vision-based techniques are also common [21][22]. Additional sensors, such as GPS or any other satellite navigation techniques and additional inertial sensors may further enhance performance. Due to the wide range of data sources, sensor fusion plays a key role in these projects. [23]

Since LIDAR is not installed on the developed vehicle framework its primary environmental sensor is the camera and its image processing algorithms. Having an independent controller unit the camera provides lane detection, road sign or object information on a low level through a CAN interface. The framework has the capability of mounting radar or ultrasonic sensors on the cart.

C. Actuators

Three main actuators are applied in the system: the driving motor, the steering actuator and the brakes.

An electric motor has been chosen to drive the cart, where three main alternatives seem to be a potential choice [24]:

- Series wound brushed DC motor, with easy controllability, but high wear and maintenance needs;
- Asynchronous motors are widely used in industrial applications, yet the generation of the three-phase power supply needed to drive the motor, control problems and the high investment costs are significant disadvantages of this choice.
- Brushless DC motors have all the advantages of the brushed motors with a slight inconvenience in their control, but the maintenance needs, the reasonable costs and the good power-density ratio makes them the best choice amongst the three.

From the three possibilities the BLDC solution has been chosen: the HPB5000B motor with a compact design, a water resistant body, a stainless steel shaft, and a self-cooling fan [25]. The main technical parameters of the motor are detailed in Table III.

TABLE III. TECHNICAL PARAMETERS OF THE CHOSEN HPM5000B BLDC MOTOR

Voltage	48V/72V/96V/120V
Rated Power	3KW-7.5KW
Efficiency	91%
Speed	2000-6000rpm (customizable)
Weight	11Kg
Height	126mm
Diameter	206mm

A chain drive (H428 type and a gear ratio of 43/18) has been chosen to transfer the torque of the motor to the rear axle.

This drive train is able to accelerate the cart at $2.5m/s^2$ up to $40km/h$ with the maximum obtainable velocity of $60km/h$, which is satisfactory for our experimental needs.

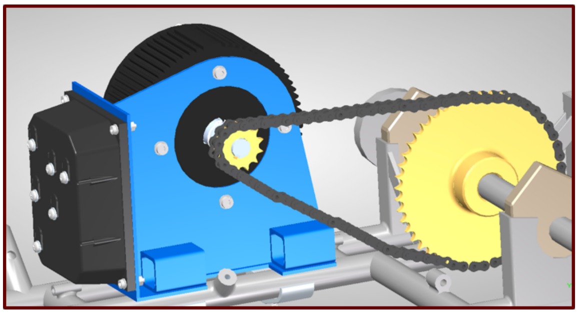

Fig. 7. Drive Chain instalment design

The BLDC motor and its control unit has the capability of regenerative braking. This way the original brake system of the cart is only used as a redundant stopping method for enhancing safety and position fixing.

There are many possibilities to realize steering functionality. During the design of the vehicle two possible motor types were examined: a stepping motor and a DC servo. The torque requirement of steering in this cart-type vehicle is approximately $10Nms$ and the maximum rotation angle is $45°$. Accuracy of the position control is necessary, and because of the external physical impacts overload protection is also important. This requirement is fulfilled by using an adjustable torque limiter. A SanMotion 103H7823-1740 stepping motor from SANYO-DENKI [26] has been chosen for the steering actuator, having a torque of $2.7Nm$ with the rated current of $2A$. The required torque is achieved with a gear drive having 4.2 transmission value. Instalment design of the subsystem is shown in Fig. 8.

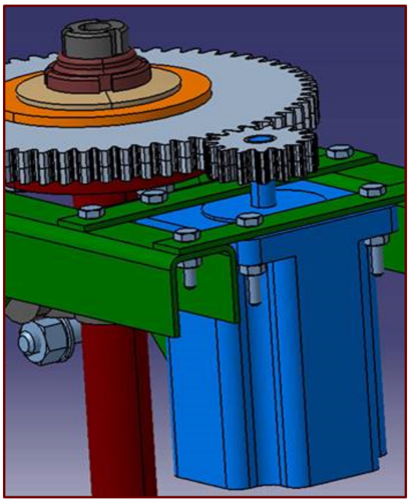

Fig. 8. Steering Actuator instalment design

D. Control architecture

The design of the control is based on the requirement that the architecture must be as modular as possible, so that the cart could be easily reconfigured for different projects. This approach resulted in the separation of the actuator handling and the high-level control logic. For this purpose two main ECUs have been designed to operate the vehicle:

- the so called "central ECU" handles sensor information and calculates driving logic,
- the "actuator ECU" controls the actuators.

These units are connected via CAN network where the filtered and processed sensor data are transferred along with the control commands. The control architecture and its logical specification are shown in Fig. 9.

As it was mentioned before drive, steering and brake actuators are handled by the Actuator ECU, which generates the pulse-width modulated and analogue control signals for them in order to keep the reference value received from the central control.

These well separated tasks do not require high computational power, so the micro-controller chosen for this task is an Atmel AT90CAN128 [15], a low-power 8-bit AVR RISC-based unit.

The tasks of the central ECU are the collection of sensor data and the calculation of the vehicle control. Sensor data is collected through a CAN network, which also serves the purpose of keeping parts of the design as interchangeable as possible. The determination of the control outputs is based on the fused sensor data.

This way the implemented functions are:

- Lane following at a fix speed,
- Object avoidance.

The control unit has high computational needs even when performing the simplest autonomous driving tasks. This leads to the choice of the AT32UC3C2512C microcontroller[27] a complete System-On-Chip controller based on the AVR32UC RISC processor. The maximum clock frequency of 66 MHz is sufficient for the defined tasks or any simple student project focusing on the basic driver assistance functionalities although for more complex projects the replacement of this unit will be necessary.

IV. Hardware in the Loop Testing

Providing development and test environment for the design of autonomous vehicle function is not trivial since the process of the design of a relatively small scale vehicle poses feasibility questions regarding the iterative process of function design and testing. The project has numerous subtasks whose progress can not be handled in a linear way, thus parallel development is needed to achieve advance. However, the developed function must be tested, first at model level and then at the implementation level with the final embedded system.

A. Theoretical control loop

For creating a convenient development environment system functions must be divided into four well-defined areas:

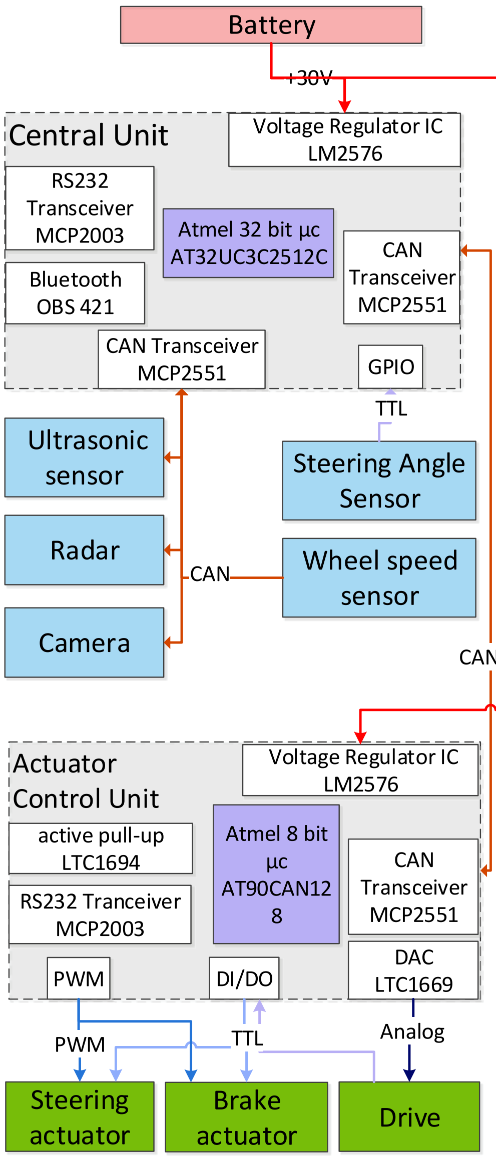

Fig. 9. Control architecture logical specification

1) Vehicle properties and dynamics with the appropriate modelling of actuators and sensors.
2) Environment possessing the road, signs, other vehicles and objects. Because of the vision system, the environment model must also implement visualization.
3) Sensors, with special attention to the camera since it has an important role and a significant number of development requirements.
4) The controller with the implemented logic.

On the first level of testing and development the performance of any subsystem can be evaluated by using previously defined or recorded static raw data as input. However, on the next level, where more than one subsystem must interact and even on the system integration phase this is not feasible because of the dynamic behaviour of the interaction between the units. On the one hand the different development progress of the individual subtasks prevents overall testing, on the other hand for full system evaluation a track is needed.

These are the reasons that lead to the development and application of a Hardware-in-the-Loop (HIL) [28][29] system framework to ensure the smooth development process where all subsystem parts can be easily interchanged by any of its three representation:

- Recorded, previously-defined raw data or in some cases a simple computer model. (Static data)
- Detailed computational model representation. (Simulation)
- Real system/reality.

The usage of static test datasets and the real application on the other end are easy to imagine. The problem emerges when one must implement dynamic simulated behavior. Fig. 10 presents the simplified control loop and the possible simulation replacements. Naturally not all combinations of these choices at all four subsystems result in practical test cases.

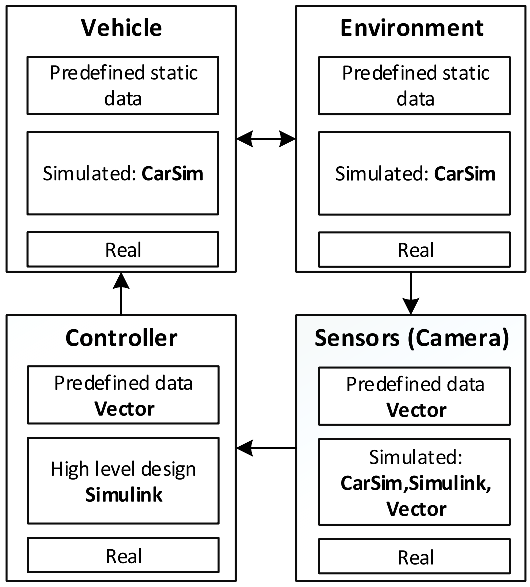

Fig. 10. The modular possibilities of Harware-in-the-Loop development

B. Simulation elements of the HIL process

The simulation elements of the Hardware-in-the-Loop testing and development are shown in Fig. 11.

There are a few complex vehicle modelling solutions on the market from which CarSim has been chosen to perform the

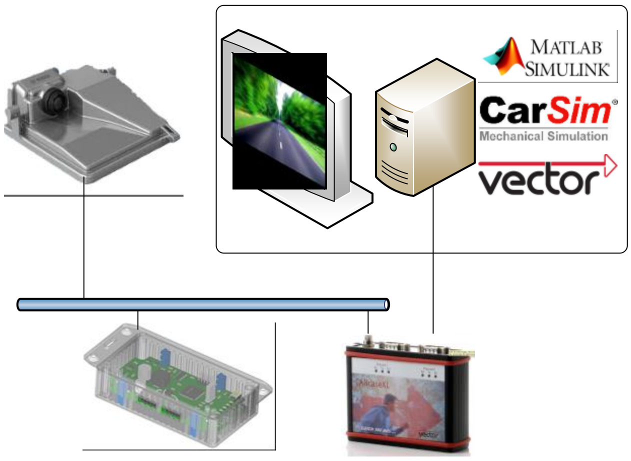

Fig. 11. Simulation elements of the HIL process

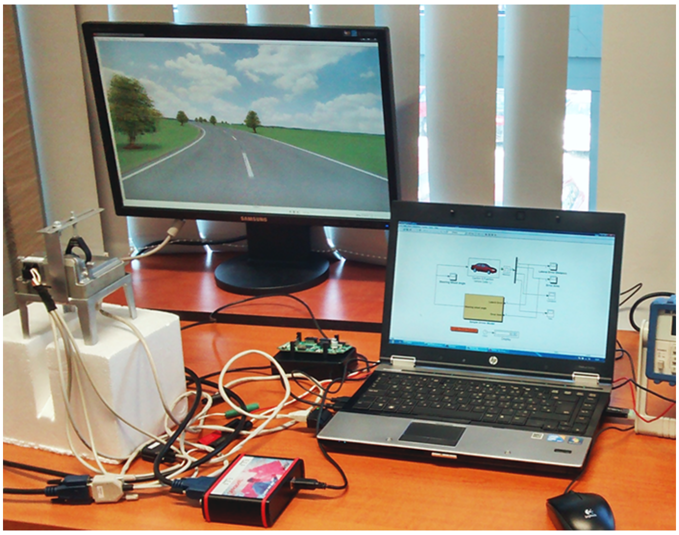

Fig. 12. Example Hardware-in-the-Loop configuration

simulation of dynamics, actuators, track and environment. The system is capable of providing 3D visualization from multiple viewpoints with different focal lengths. CarSim simulates the dynamic behaviour of passenger cars, race cars, light trucks, and utility vehicles. It animates simulated tests and its outputs are the calculated variables in real-time [30].

Any tool capable of being interfaced through CAN can be used for high level control design purposes, although the original solution uses Matlab/Simulink because of its ease of use and high interfacing capabilities to CarSim.

CAN communication is managed by the CANCase XL, a professional CAN/LIN/J1708 bus interface with USB 2.0 interface from Vector GmbH.

Finally an example setup can be examined in Fig. 12 where vehicle dynamics and 3D visualization are performed by CarSim, but its image is processed with the real camera (with the help of a focus correction lens), which transfers the detected features via CAN to the control ECU. The calculated actuator commands are interpreted by a CANCaseXL module and a simple Simulink model closes the loop towards CarSim.

V. Conclusions and Future Expansion Possibilities

The paper has presented two frameworks that are intended to aid project based education of vehicle mechatronics students. Working on connected yet individual tasks has both competitive and cooperative aspects. The students working on the same tasks exchange ideas and compare their results during the design process. The individuals working on ECUs that must work together spend hours testing the interaction and communication of their units. The framework inherently encourages team work since at the end of the semester all units are connected together to perform joint operation just as they would do in a real vehicular environment. Naturally a high amount of educational effort is needed in this kind of development since continuous consultations with the developer students is indispensable.

The paper has also presented an experimental vehicle framework based on a laboratory vehicle. The design of the system aims to develop a modular structure in which sensors and actuators can be easily interchanged when the scope of the high level function development changes. Several driver assistance solutions can be developed and tested with the usage of this relatively small-scale vehicle, while our future plans are aimed at the development of autonomous functions. These envisioned goals can be achieved on different levels with the framework. Engineering students can work on comparatively lower complexity level problems, such as lane detection and keeping, while in the research intelligent vehicle solutions and self driving capabilities are targeted. A vehicle framework by itself is not sufficient for the overall development process therefore a Hardware-in-the-Loop system is also introduced to aid the entire process on all design levels.

Universities need to provide the industry with engineers for the interdisciplinary field of mechatronics, especially in the field of vehicle electronics and mechatronics, since it is a mix of electrical, mechanical and software engineering.

References

[1] "Organisation internationale des constructeurs d'automobiles (oica)," online:www.oica.net.

[2] O. Yilmaz and K. Tunçalp, "A mixed learning approach in mechatronics education," *Education, IEEE Transactions on*, vol. 54, no. 2, pp. 294–301, 2011.

[3] H.-H. Erbe and F. W. Bruns, "Didactical aspects of mechatronics education," in *5th IFAC International Symposium on Intelligent Components and Instruments for Control Applications (SISACA). Aveiro-Portugal*, 2003.

[4] Y. Wang, B. Schwarz, Y. Pan, B. Chen, H. Wiedmann, C. Xie, H. Wang, Y. Yu, and X. Feng, "Mechatronics education at {CDHAW} of tongji university: Structure, orientation and curriculum," *Mechatronics*, vol. 18, no. 3, pp. 172 – 177, 2008.

[5] S. Meek, S. Field, and S. Devasia, "Mechatronics education in the department of mechanical engineering at the university of utah," *Mechatronics*, vol. 13, no. 1, pp. 1 – 11, 2003.

[6] J. B. Hargrove, "Curriculum, equipment and student project outcomes for mechatronics education in the core mechanical engineering program at kettering university," *Mechatronics*, vol. 12, no. 2, pp. 343 – 356, 2002, mechatronics Education in Europe and the United States.

[7] K. Tan, K. Tang, and T. Lee, "Foreword for special issue on design-centric mechatronics education," *Mechatronics*, vol. 23, no. 8, pp. 917 –, 2013.

[8] Y. Wang, Y. Yu, H. Wiedmann, N. Xie, C. Xie, W. Jiang, and X. Feng, "Project based learning in mechatronics education in close collaboration with industrial: Methodologies, examples and experiences," *Mechatronics*, vol. 22, no. 6, pp. 862 – 869, 2012, special Issue on Intelligent Mechatronics (LSMS2010, ICSEE2010).

[9] Y. Wang, Y. Yu, C. Xie, X. Zhang, and W. Jiang, "A proposed approach to mechatronics design education: Integrating design methodology, simulation with projects," *Mechatronics*, vol. 23, no. 8, pp. 942 – 948, 2013.

[10] M. E. Grimheden, "Can agile methods enhance mechatronics design education?" *Mechatronics*, vol. 23, no. 8, pp. 967 – 973, 2013.

[11] L. Rising and N. Janoff, "The scrum software development process for small teams," *Software, IEEE*, vol. 17, no. 4, pp. 26–32, Jul 2000.

[12] H. Komoto and T. Tomiyama, "A system architecting tool for mechatronic systems design," *CIRP Annals-Manufacturing Technology*, vol. 59, no. 1, pp. 171–174, 2010.

[13] K. Forsberg and H. Moozhor, "The relationship of system engineering to the project cycle," in *The 12th INTERNET World Congress on Project Management*.

[14] T. Becsi, S. Aradi, and P. Gaspar, "Design of an educational emulation framework for mechatronics control unit development," in *Mechatronic and Embedded Systems and Applications (MESA), 2014 IEEE/ASME 10th International Conference on*, Sept 2014, pp. 1–5.

[15] *AT90CAN32/64/128 Complete Datasheet*, Atmel Corporation, 2008.

[16] *BIGAVR User's Manual*, Mikroelektronika, 2006.

[17] *WECAN USB PC ADAPTER*, Inventure Automotive Electronics.

[18] *NI myDAQ User Guide and Specifications*, National Instruments Corporation, 2011.

[19] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *Intelligent Vehicles Symposium (IV), 2011 IEEE*. IEEE, 2011, pp. 163–168.

[20] T. Luettel, M. Himmelsbach, and H.-J. Wuensche, "Autonomous ground vehicles, concepts and a path to the future," *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1831–1839, 2012.

[21] M. Bertozzi, A. Broggi, and A. Fascioli, "Vision-based intelligent vehicles: State of the art and perspectives," *Robotics and Autonomous Systems*, vol. 32, no. 1, pp. 1–16, 2000.

[22] C. J. Taylor, J. Košecká, R. Blasi, and J. Malik, "A comparative study of vision-based lateral control strategies for autonomous highway driving," *The International Journal of Robotics Research*, vol. 18, no. 5, pp. 442–453, 1999.

[23] D. Gohring, M. Wang, M. Schnurmacher, and T. Ganjineh, "Radar/lidar sensor fusion for car-following on highways," in *Automation, Robotics and Applications (ICARA), 2011 5th International Conference on*. IEEE, 2011, pp. 407–412.

[24] S. Aradi, T. Becsi, and P. Gaspar, "Experimental vehicle development for testing autonomous vehicle functions," in *Mechatronic and Embedded Systems and Applications (MESA), 2014 IEEE/ASME 10th International Conference on*, Sept 2014, pp. 1–5.

[25] High power bldc motors, hpm5000b. Golden Motor Technology Co Ltd. [Online]. Available: http://www.goldenmotor.com/hubmotors/

[26] *Stepping Motors data sheet, SanMotion 103H7823-1740*, SANYO DENKI CO.,LTD.

[27] *Atmel AVR 32-bit Architecture Manual Complete*, Atmel Corporation, 2012.

[28] W. Deng, Y. Lee, and A. Zhao, "Hardware-in-the-loop simulation for autonomous driving," in *Industrial Electronics, 2008. IECON 2008. 34th Annual Conference of IEEE*, Nov 2008, pp. 1742–1747.

[29] D. J. Verburg, A. C. Van der Knaap, and J. Ploeg, "Vehil: developing and testing intelligent vehicles," in *Intelligent Vehicle Symposium, 2002. IEEE*, vol. 2. IEEE, 2002, pp. 537–544.

[30] *CarSim*, Mechanical Simulation Corporation, 2013.

Tamás Bécsi received both the M.Sc. and Ph.D. degrees from the Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics in 2002 and 2008, respectively. Since 2002 he has been a PhD student, since 2005 he has been an assistant lecturer and since 2014 he has been an associate professor at the Department of Control for Transportation and Vehicle Systems, Budapest University of Technology and Economics. His research interests include linear systems, embedded systems, traffic modeling and simulation. His research and industrial works have involved railway information systems and vehicle control.

Szilárd Aradi received the M.Sc. degrees from the Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics in 2005. Since 2005 he has been a PhD student and since 2009 he has been an assistant lecturer at the Department of Control for Transportation and Vehicle Systems, Budapest University of Technology and Economics. His research interests include embedded systems, communication networks, vehicle mechatronics and predictive control. His research and industrial works have involved railway information systems, vehicle on-board networks and vehicle control.

Péter Gáspár received both the M.Sc. and Ph.D. degrees from the Faculty of Transportation Engineering and Vehicle Engineering, Budapest University of Technology and Economics in 1985 and 1997, respectively, and the D.Sc. degree in control from the Hungarian Academy of Sciences (HAS) in 2007. Since 1990 he has been a senior research fellow at the Systems and Control Laboratory (SCL), Computer and Automation Research Institute, HAS and since 2007 he has been a research advisor. He is Head of the Vehicle Dynamics and Control Research Group within SCL. He is the head of the Department of Control for Transportation and Vehicle Systems, Budapest University of Technology and Economics. He is a member of IFAC Technical Committee on both the Automotive Control and the Transportation Systems. His research interests include linear and nonlinear systems, robust control, multi-objective control, system identification and identification for control. His research and industrial works have involved mechanical systems, vehicle structures and vehicle control.