

Automated Driving Cycle Simulation and Measurement System based on dSpace DS1103

T. Haubert, P. Mindl, Z. Cerovsky, P. Mnuk

Czech Technical University in Prague, Faculty of El. Engineering, Dept. of El. Drives and Traction, tomas.haubert@fel.cvut.cz

Abstract

This paper presents a system for automated driving cycle simulation and measurement for electric vehicles. The system is a part of a laboratory located in VTP Roztoky and is used for electric vehicle powertrain testing. The main part of the control system is dSpace DS1103.

1 Introduction

Increasing of living standards increased needs for people transport. Gas emissions and fossil fuel consumption are actual problems for global environment quality on the Earth. Gas emission of car internal combustion engines (ICE) brings many ecological problems in big cities specially. One way how to reduce gas emissions in big cities is to use electric vehicles. Nowadays it is planning in big cities (Paris, London) to limit an access to the cities for all vehicles with a combustion engine. The development of the electric vehicles is faster in recent years and now it is very progressive branch of industry. The most significant problems of recent electric vehicles are high purchase price and short driving distance. The driving distance depends on the capacity of the battery pack where all energy is stored in electrochemical cells. During the drive the energy flows from the battery to the powertrain where is converted into the mechanical energy. The capacity of the battery, efficiency of the powertrain and the drivers style determines maximum length of the driving distance.

2 Laboratory equipment

Science and Technology Park in Roztoky town has been founded in the year 2012. The place is situated on the Nord-west of Prague. Main field of research works is focused onto the automotive techniques based on advanced internal combustion engines (ICE) and progressive electric (ED) and hybrid drives (HD) technology. One of the special laboratories is focused on electric and hybrid-electric drives measurement and development. For this purposes is laboratory equipped by electric power supplies and by dynamometric testing bench.

Testing bench is designated for AC and DC supplied powertrains. For induction motor measurement special power source with secondary tap-changing transformer is used. This transformer enable to supply very spread spectrum of induction motors designated for automotive applications. Similarly for other types of motors, like synchronous permanent magnets motor or brush-less DC motors, supplied via DC/AC inverter, battery simulator for supplying of this assembly is designated. Like frequency converter supplying dynamometer so battery simulator operates in active front-endmode. Such arrangement enables bi-directional energy transmission (from mains to the machine and from the machine to the mains in case of recuperation regime). Dynamometer from VUES company is equipped by asynchronous machine with rotary stator. Mechanical torque is measured by means of strain-gauge-bridge.

Block diagram of the whole testing bench is shown in the figure 1. Dynamometer control system is equipped by basic controlling PC. This PC via RS 485 lines controls converters supplying dynamometer and testing induction motor. Torque and revolutions are measured on the dynamometer by means of separated sensors. Electrical parameters tested motor supply voltages, currents and powers are measured and calculated by YOKOGAWA precision power

analyser. Measured and calculated data are transmitted via Ethernet line to the control computer. For data collecting, calculating and plotting ASMOT programme is used. Application programme ASMOT is firmware specialized for routine measurement of broad spectrum of induction motors characteristics.

The simulation system has to simulate EV drive along a track with a known altitude according to a predefined speed of the vehicle. The vehicle speed profile and the profile of the angle of inclination are the main inputs for the control system. There has to be also a vehicle dynamic model which produces a load for the electric drive. As the control platform dSpace single board DS1103 was selected. The dSpace DS1103 (see Fig. 2) is a controller board used for rapid control prototyping. The control model is developed in MATLAB/Simulink programmable environment. Based on the developed model generates Simulink C code which is compiled and uploaded into the dSpace.

dSpace DS1103 is a part of the dSpace AutoBox system with Ethernet interface for the communication with PC. AutoBox needs only 12 V power supply and can be mounted direct in a vehicle if a testing in vehicle is needed. The communication schematic is shown in the Fig. 3. The main part is dSpace DS1103 which controls SEVCON via CAN bus and dynamometer via RS-232. On the CAN bus is connected also a PC using USB/CAN interface. The CAN bus is using a CANOpen protocol and RS-232 is using own developed protocol.

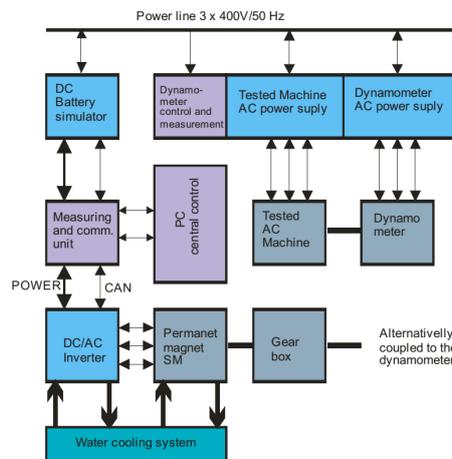


Figure 1: Block diagram of automotive electric powertrain testing bench



Figure 2: dSpace DS1103

3 Software in dSpace DS1103

The control algorithm is developed in MATLAB/Simulink programmable environment. The control system is using Stateflow toolbox which enables a development of the state machines in a graphical way. The schematic of the control system is show in the Fig. 4. Then main part is the Main Logic block. This block reads the statuses from other blocks and send the commands for these blocks. The other blocks are responsible for sub-coordination of SEVCON power converter, Dynamometer and Visualization. These blocks are also directly connected to the Interface block. These block is responsible for communication on the CAN bus and on the RS-232. Each block except interface block consists of a state machine and a sequence.

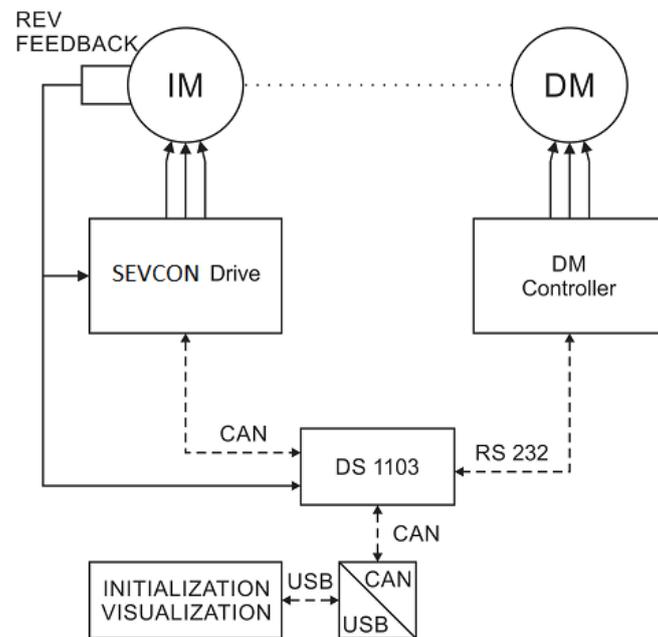


Figure 3: Schematic of communication

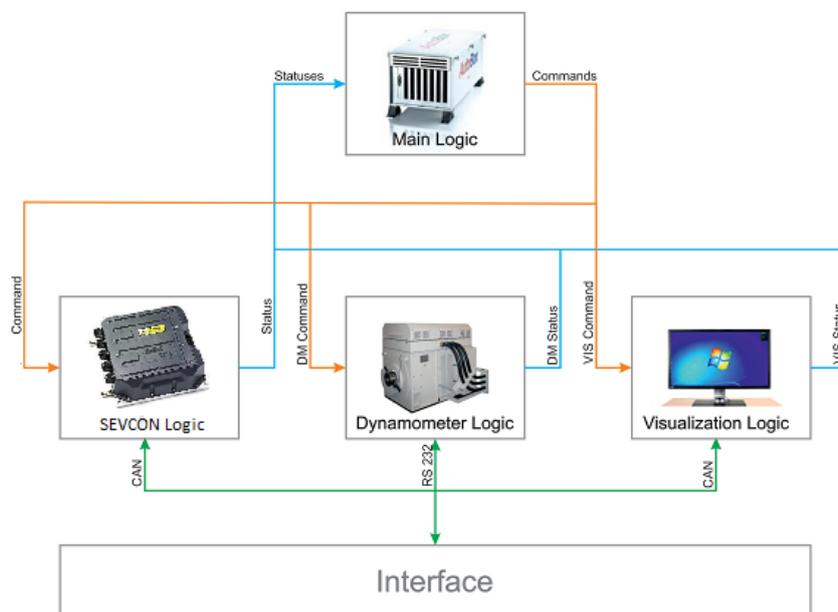


Figure 4: Schematic of control algorithm in dSpace

4 Electric Vehicle Optimizer Application

The Electric Vehicle Optimizer Application (EVO application) is an application developed in MATLAB environment. The application is divided into three main parts:

- Computer Part
- Graphical User Interface
- Data Gathering Part

The application is responsible for the track planning via Google maps or GPX measured data, for the vehicle parameters settings, for the optimization of the energy consumption calculation and for the visualization of the speed and position on the track during the driving cycle measurement of the test bench.

Computer Part

The computer part is responsible for all calculation on a given track and a vehicle. The computer part of the project has the following classes:

- @Drive: superclass defines the project and associates the other classes
- @Vehicle: defines the vehicle parameters and environment parameters
- @Track: gathers all data of the given track
- @Speed: specifying the speed profile and implements driver simulation controller
- @Energy: provides all calculation of the energy and time
- @Optim: implements an algorithm which is the main part of this thesis

As mentioned, the @Drive is the superclass of all computer classes, which represent the properties of this superclass. The vehicle, speed and track classes are defined as handle class, in order to have direct access to the vehicle, speed and track information in every class and keep the only instance of these objects. Handle represents a reference to an object and it is simply passed to every single class as a property to provide direct access and keep the object as unique.

The issues of graphical user interface (GUI) in Matlab is described in many publications, however, almost all of them use a built-in graphic editor GUIDE, which is not optimal for larger-scale projects. Hence, advanced GUI technique is introduced to create a well-arranged and easily expandable GUI. These two demands lead to the use of object oriented programming, as in the case of computer classes. In the GUI, each of the classes represents one window with its graphical objects. To explain the technique, the simplest window of the object is selected. The window in figure 5 welcomes the user when the application is started up. There are only four graphical objects in the window, three buttons and a title. This window represents class @startup of the project in interface package folder +interface.

Each of the classes of GUI includes a set of typical properties to simplify the handling of the window:

- The EVO property specifies the project where the window belong. This superclass connects the interface to the computer classes.
- Position determines where the windows appears when opened. Value of the position is either normalized or in pixels.

- Handles is a structure with handles of every graphical object of the window and their children. The handle structure allows the developer comfortable changes of any object or property.
- isOpened represents open flag to determine whether the window is opened or not.

As in the case of properties, the classes of GUI have also characteristic methods. The input of the class method is always the handle of the object to allow the method to manipulate and access properties and to call other methods. The most important method is the constructor of the class, where the object is assigned to the project. The class also includes callbacks of the graphical objects.

As we can see in figure 6, the web page includes a few objects, and the HTML script is simple as well, however, the JS code is far more complicated. Google maps service distinguishes several interfaces to obtain different geographical data. To gather data sufficient for the calculation, the application uses Google Maps Directions API to design the track and collect GPS coordinates, and Google Maps Elevation API to collect elevation level of the track. Both of these interfaces use JavaScript Object Notation (JSON) for input and output, and the use of the interface is quite comfortable. Moreover, complete documentation is provided, including examples of use. The use of Google Maps API is, for free users, limited in number and length of the request. Therefore, the JS must split the data obtained from Direction API to more requests for Elevation API and compose the result back.



Figure 5: Start-up EVO application window



Figure 6: Implemented web application for track design

5 Experimental Results

The result represents a comparing of driving cycle near Roztoky. The driving cycle starts in front of the building where is located the laboratory. Then it continues in direction to Velke Prilepy up to the small hill. After about 700 meter the track turns right in direction to Unetice. It goes a little bit up the hill and down the hill to Unetice village. The track continues straight on the main street. At the end of Unetice there is again up the hill and the track finishes about 100 m before roundabout where is a Lidl supermarket located. The map of this track together with altitude profile is shown in the figure 7. The figure is a screenshot from the EVO application.

The first drive was performed by Skoda Rapid EV prototype provided by Skoda auto a.s. The vehicle has an asynchronous motor as a main drive with DSG automatic transmission. The transmission allows to have constant motor revolution / vehicle speed ratio. The driving cycle data from CAN bus was logged via USB/CAN converter from Vector by Vehicle Network Toolbox included in MATLAB. The data from GPS was logged from built-in GPS receiver via serial interface. The most important information was the vehicle speed, which came from ESP unit of the vehicle. The driver was asked for a drive oriented to low energy consumption. The speed profile of the track is shown in the figure 8.

On the test bench is the same electric motor and power converter, but different battery model operating with different voltage and different vehicle model. The pure comparing of logged data from vehicle and from test bench would not be correct. I used the measured speed profile as the input to the test bench and used the same arrival time and the same track as the input to the optimization algorithm. I made three measurements:

- The first measurement uses as the input the measured speed profile in koda Rapid EV
- The second measurement uses the calculated speed profile from optimization algorithm
- The third measurement uses the calculated speed profile from reference drive algorithm

The all measurements use the same test bench with the same initial values, because we need the direct comparing of total energy consumption. The speed profiles were uploaded one

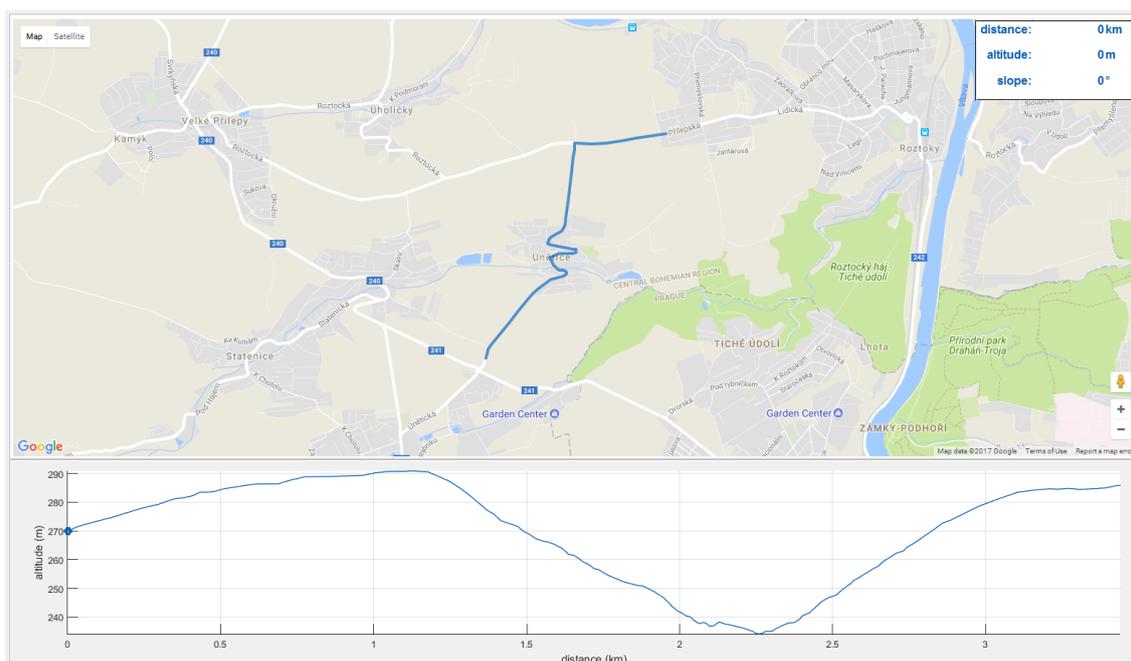


Figure 7: Map and altitude profile

by one into the dSpace. After uploading and starting the speed profile simulation is only the dSpace responsible for the simulation control. The PC with EVO application is using only for showing the actual state of simulation. The all results of the simulations were logged via ASMOT software.

Comparing

The total energy consumption was calculated from the battery simulator control software which calculates the total energy consumed and recuperated. The total energy consumption is shown in the figure bellow. The consumed energy of Skoda Rapid was 0.54 kWh, the consumed energy of optimization algorithm was 0.52 kWh and the consumed energy of the reference algorithm was 0.56 kWh.

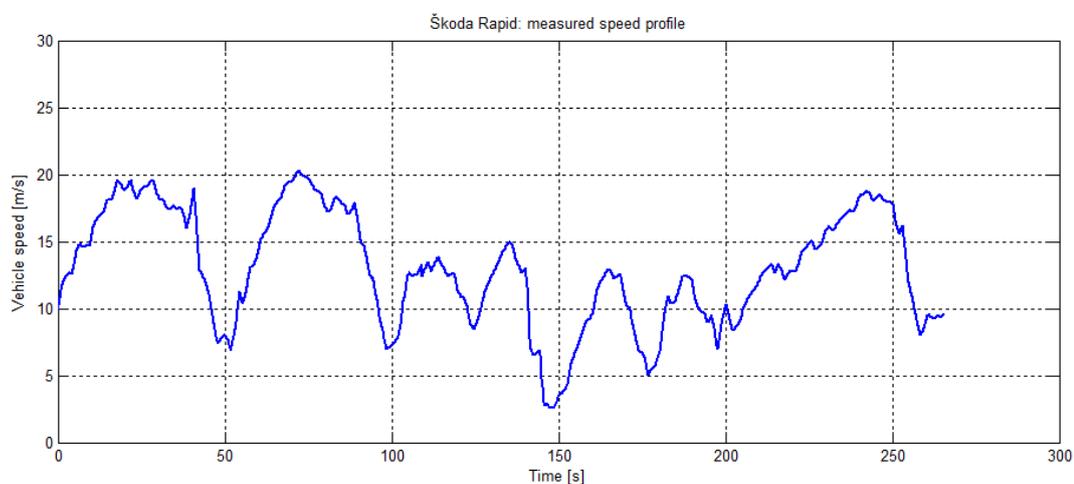


Figure 8: Skoda Rapid EV: measured speed profile

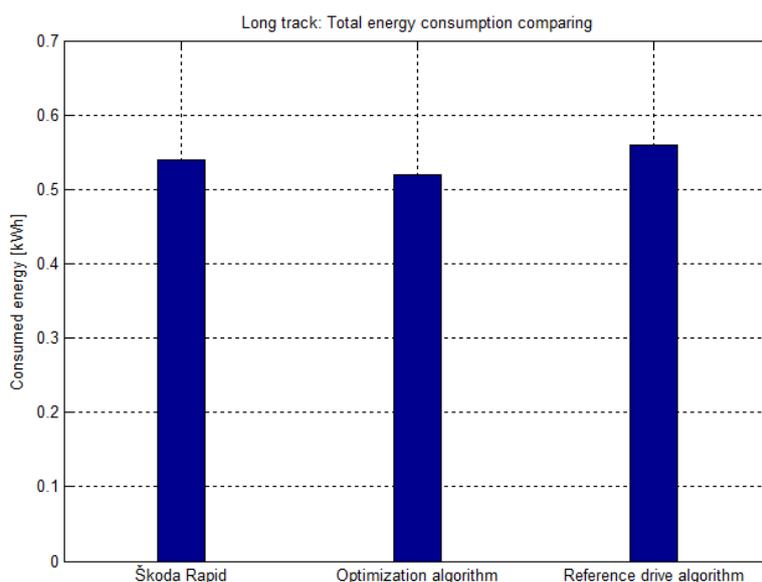


Figure 9: Total energy consumption comparing

6 Acknowledgement

This research has been realized using the support of Technological Agency, Czech Republic, programme Centres of Competence, project #TE01020020 Josef Boek Competence Centre for Automotive Industry and using the support of EU Regional Development Fund in OP R&D for Innovations (OP VaVpI) and Ministry for Education, Czech Republic, project #CZ.1.05/2.1.00/03.0125 - ED2.1.00/03.0125 Josef Boek Research Centre for Vehicles of Sustainable Mobility. This supports are gratefully acknowledged.

7 Conclusion

The main goal of this article was to present the dynamometer test bench in the laboratory in Roztoky. The main result of this laboratory is to design and verify an control algorithm for optimal speed profile calculation and simulation. The saved energy in comparing with the real drives is about 3 - 5 %.

Ing. Tomas Haubert

Czech Technical University in Prague, Faculty of Electrical Engineering, Department of Electric Drives and Traction, tomas.haubert@fel.cvut.cz

doc. Ing. Pavel Mindl, CSc.

Czech Technical University in Prague, Faculty of Electrical Engineering, Department of Electric Drives and Traction, mindl@fel.cvut.cz

prof. Ing. Zdenek Cerovsky, DrSc.

Czech Technical University in Prague, Faculty of Electrical Engineering, Department of Electric Drives and Traction, cerovsky@fel.cvut.cz

Ing. Pavel Mnuk, CSc.

Czech Technical University in Prague, Faculty of Electrical Engineering, Department of Electric Drives and Traction, mnuk@fel.cvut.cz