Cloud Aided Implementation of Energy Optimal Look-ahead Speed Control

Alexandros Soumelidis [*] **Péter Gáspár** [*] **Ádám Kisari** [*]
Ádám Bakos [*] **Balázs Németh** [*] **András Mihály** [*]
Zoltán Hankovszki [**]

[*] *Systems and Control Laboratory, Institute for Computer Science and Control, Hungarian Academy of Sciences,
Kende utca 13-17, 1111, Budapest, Hungary.
(e-mail: [alexandros.soumelidis; peter.gaspar; adam.kisari; adam.bakos; balazs.nemeth; andras.mihaly]@sztaki.mta.hu).*
[**] *Knorr-Bremse Hungary, Major u. 69., 1119 Budapest, Hungary.
(e-mail: zoltan.hankovszki@knorr-bremse.com)*

Abstract: The aim of the paper is to present the development, implementation and testing of a novel look-ahead cruise control system. The method strives to decrease fuel consumption while maintaining travel time limitations. In the development phase the control algorithm has been evaluated and fine tuned using state of the art software- and hardware in the loop simulations. The real-world implementation and tests were conducted on Hungarian highways with a Volvo FH13 truck. The test results confirmed the simulations and proved that a reduction in fuel consumption can be achieved this way.

Keywords: Automotive control, Optimal control, Look-ahead, ADAS, Cloud

1. INTRODUCTION

The main goal and motivation of the project presented in this paper is the never ending endeavour of decreasing the fuel consumption of all vehicles. Our main concern in this great task is to increase the efficiency of road freight vehicles using a look-ahead cruise control algorithm which maintains an energy-optimal speed along the way of the truck.

Several look-ahead control algorithms were already shown to be used by vehicular cruise control systems. These methods utilise known parameters of the environment, namely road elevation data, to improve fuel consumption. See Ivarsson et al. (2009); Nouveliere et al. (2008); Hellström et al. (2010); Passenberg et al. (2009); Sahlholm and Johansson (2009) The concept of look-ahead control can also be used in multi-vehicle scenarios such as platoons. See Alam et al. (2013). The specific algorithm that was used throughout this project was introduced in Németh and Gáspár (2013).

This method utilises advance information on road parameters downloaded from cloud services aggregating such road characteristics. In our case the main concerns are elevation data to obtain the slope of the road and speed limits to consider when calculating the optimal speed.

The use of simulators is a well known practice in the industry to reduce implementation costs and to speed up the development process. This is the reason why we used the following scheme of development. After the concept of the algorithm was created, simulations in the TruckSim software were started. These provided some improvements to the algorithm and showed the concept is valid. Then a hardware in the loop (HIL) simulation was carried out. This environment also used the TruckSim software as the simulated part, but the controller was implemented on the final embedded hardware. From the HIL simulations to the test vehicle minimal modifications were required.

We aimed to do the road tests representing real-life scenarios, that is why we chose a frequently used freight path in Hungary, the M1 highway as our main test route. Some shorter, preliminary tests were also conducted on the runway of Tököl airport and on the M6 highway. The time of the tests was chosen to be at night to minimize the traffic affecting the test and to maintain maximal safety of others as well.

The paper is organized as follows. Section 2 briefly summarizes the fundamentals of the proposed look-ahead cruise control method. Section 3 describes the systems used in the simulated tests. In Section 4 the physical implementation is discussed. Finally in Section 5 the test results are shown.

2. PRINCIPLES OF THE LOOK-AHEAD ALGORITHM

The algorithm assumes that the required parameters of the upcoming path such as speed limits and changes in elevation are known. When using this method the road ahead is divided into a predefined number of pieces, as depicted in Fig. 1. The proposed method, aims to calculate

[⋆] The research was supported by the National Research, Development and Innovation Fund through the project "SEPPAC: Safety and Economic Platform for Partially Automated Commercial vehicles" (VKSZ 14-1-2015-0125).

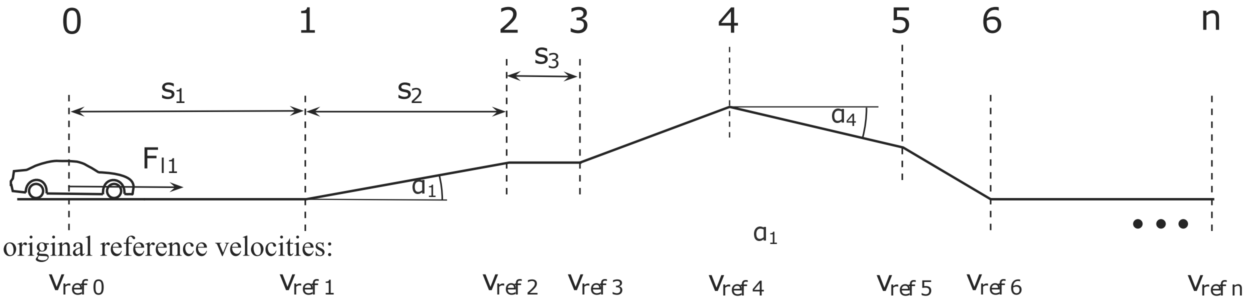

Fig. 1. Selected road segments

a reference velocity for every section in such a way that applying a constant traction force should accelerate or decelerate the vehicle to the desired speed by the end of each section. This way the force reference can be regarded as a piece-wise constant signal.

The speed of the truck at the n^{th} section end can be defined as follows:

$$\dot{\xi}_n^2 = \dot{\xi}_0^2 + \frac{2}{m} s_1 F_{l1} - \frac{2}{m} \sum_{i=1}^{n} s_i F_{di} \tag{1}$$

where s_i $i \in [1, n]$ is the distance to the section end, F_{l1} is the desired force acting on the vehicle, F_{di} $i \in [1, n]$ is the disturbance force, which can be broken down to the road slopes $F_{di,r}$ $i \in [1, n]$ and the rolling resistance and aerodynamic drag shown by $F_{di,o}$ $i \in [1, n]$.

Then, the current speed $v_{ref,0}$ is weighted with Q, while the desired velocities of the following road segments are weighted with $\gamma_1, \gamma_2, ..., \gamma_n$ such that

$$\gamma_1 + \gamma_2 + ... + \gamma_n + Q = 1. \tag{2}$$

The next formula can be given assuming the weights of Q and γ_i $i \in [1, n]$

$$\lambda = \sqrt{\vartheta - 2s_1(1 - Q)(\ddot{\xi}_0 + g sin\alpha)}. \tag{3}$$

where

$$\vartheta = Q v_{ref,0}^2 + \sum_{i=1}^{n} \gamma_i v_{ref,i}^2 + \frac{2}{m} \sum_{i=1}^{n} (s_i F_{di,r} \sum_{j=i}^{n} \gamma_j). \tag{4}$$

Two contradicting optimization problems rise that needs to be solved simultaneously. The control force needs to be minimised to provide fuel efficiency and the deviance from the speed limit must also be considered to ensure minimum travel time. To obtain the optimal control force a quadratic optimization problem must be solved; to minimize travel time the speed limit should be kept disregarding any other road parameters.

To obtain the optimal control force $F_{l1}^2 \to \min$ a quadratic optimization problem must be solved:

$$\bar{F}_{l1}^2 = (\beta_0(\bar{Q}) + \beta_1(\bar{Q})\bar{\gamma}_1 + \ldots + \beta_n(\bar{Q})\bar{\gamma}_n)^2 \to \min \tag{5}$$

considering the constraints:

$$0 \le \bar{Q}, \tag{6a}$$

$$\bar{\gamma}_i \le 1 \tag{6b}$$

$$\bar{Q} + \sum \bar{\gamma}_i = 1. \tag{6c}$$

The difference between the speed limit and the actual speed is minimized next

$$|v_{ref,0} - \dot{\xi}_0| \to \min \tag{7}$$

This brings us to selecting $\breve{Q} = 1$ and $\breve{\gamma}_i = 0, i \in [1, n]$ as the optimal solution, as in this case the speed limit is kept disregarding any other road parameters.

The two distinct and conflicting solutions can be united using the performance weights of R_1 and R_2 regarding the importance of the two objectives, where $R_1 + R_2 = 1$.

The weighting can be implemented as follows:

$$Q = R_1\bar{Q} + R_2\breve{Q} = 1 - R_1(1 - \bar{Q}) \tag{8a}$$

$$\gamma_i = R_1\bar{\gamma}_i + R_2\breve{\gamma}_i = R_1\bar{\gamma}_i, \quad i \in \{1, .., n\} \tag{8b}$$

This way the predicted solution is balanced between energy optimum and time optimum and the balance can be controlled via R_1 and R_2. See further details in Németh and Gáspár (2017).

3. SIMULATION ENVIRONMENT

Our work-flow was the following. In the beginning of the development, software-only simulations were conducted. The second phase utilised hardware in the loop tests, and finally we moved to the implementation on a real vehicle.

A simulation system has been created and extended over time to develop, implement, improve, and verify the previously described algorithm. The simulator is based on the TruckSim software package in combination with the Matlab/Simulink environment. The tests carried out on this system were fed with real-world data, captured on vehicle test runs or downloaded from open online databases such as the NASA JPL Shuttle Radar Topography Mission (SRTM) database or Google maps. The data consists of the physical parameters of the vehicle itself and the characteristics of the test route. More specifically consisting of the trajectory converted to a 2D projection, speed limits along the path and most importantly the road elevation. The test results obtained by simulations saved significant time and resources in the first phase of the development.

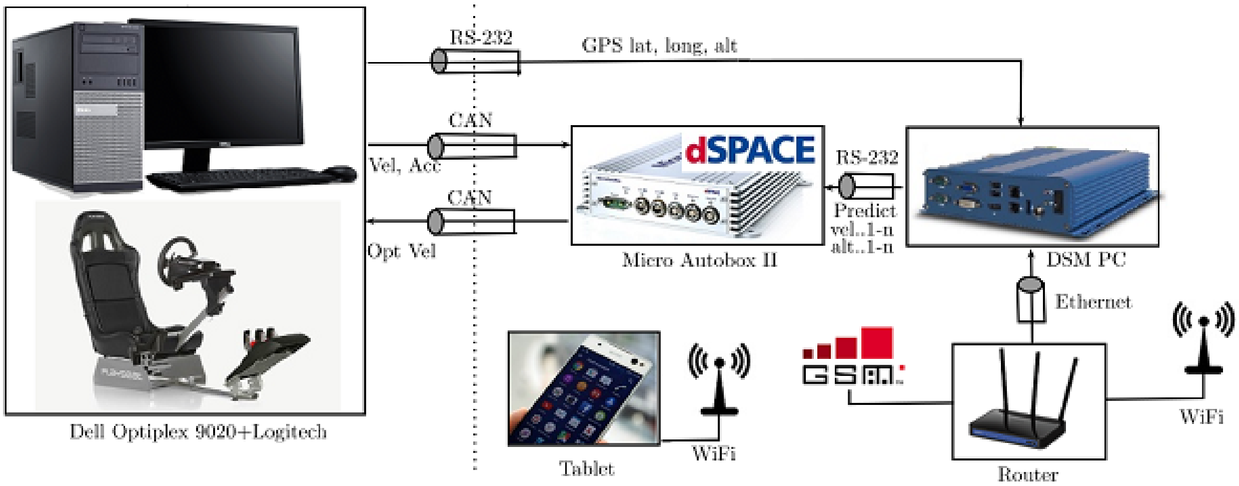

Fig. 2. Hardware in the loop architecture

3.1 Software simulations

Initially, the vehicle was simulated in TruckSim and the control algorithms were implemented in MATLAB/ Simulink. At this point the whole simulation environment is located on a single computer. This setup offers ease of implementation and fast tuning for the speed controller. Basic testing can be carried out without hardware limitations or feasibility problems. The dynamics of the truck and actuators can be considered in the models.

After this period, the now fully functioning look-ahead control algorithm was moved into a hardware in the loop environment. This is the separation of the simulated, TruckSim based vehicle dynamics and the close to final implementation of the control system on the hardware. Here the algorithm was further improved and the hardware parts were assembled to be usable on a real vehicle as it is. This also means that the simulation inputs and outputs were changed to match a real-world scenario.

The transition from the fully software based simulation to the HIL environment could be carried out step by step, one element at a time. This enabled the tests of each component separately. The final HIL architecture consists of a hardware assembly which can be fitted to a test truck in a plug-and-play manner, and a TruckSim simulation of the vehicle only. The final state of the HIL system can be seen in Figure 2.

3.2 HIL environment

As it can be seen in Figure 2, the computer running the simulation (marked Dell on the figure) is connected to the hardware components by an RS-232 and 2 CAN ports. The current speed and acceleration is sent to the MicroAutoBox II controller on the first CAN channel while the second channel is used to communicate the optimization output back to the computer. A local WiFi router is installed to connect the on-board computer (marked DSM PC on the figure) and the tablet which serves as a user interface. The on-board computer is responsible for the collection of GPS and geological data (locally and from the internet) which are sent on an RS-232 channel to the MicroAutoBox II controller for the calculation of the optimal speed.

This architecture only differs from the planned final implementation on the truck in the smallest possible manner. Only the vehicle and the GPS is simulated. The hardware components are unaware of the simulation and should be directly movable to the truck without any modification.

4. IMPLEMENTATION ON A REAL VEHICLE

After successful testing in the HIL environment the system is ready for road testing in the final phase of this project. The HIL system was designed to fully mimic a real truck thus making the complete system easily deployable on a real vehicle. Figure 5 shows that the sole difference from the HIL system is the replacement of the PC with an actual truck equipped with a regular GPS/GNSS receiver.

The road tests were conducted in cooperation with Knorr-Bremse Hungary. The test truck, courtesy of Knorr-Bremse, is a Volvo FH13 (see Figure 3), with a total mass of 16$tons$ including cargo. The truck has 3 axles, the total length is 9.33m and the engine power is 324kW

Fig. 3. The test vehicle

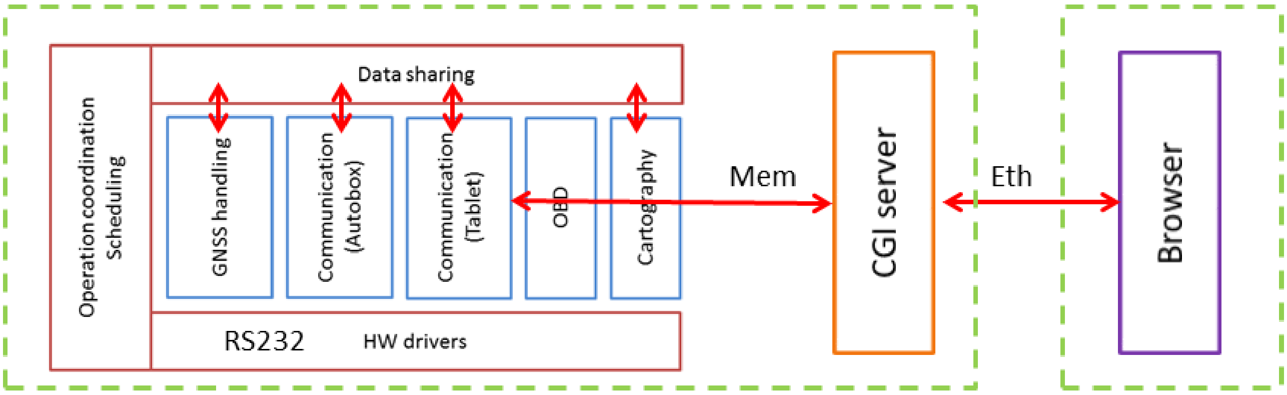

Fig. 4. Software framework of the onboard computer

The vehicle itself was modified at the fewest possible points. A lower level speed controller was installed to convert the speed output of the algorithm to acceleration and brake signals that the vehicle can process and execute. A means of measuring fuel consumption is also needed. This was implemented using the test vehicle's built in sensors and their precision was validated on previous test runs.

Due to safety reasons an emergency stop logic is integrated in the low-level speed controller. This logic automatically returns all control to the driver in case of any pedal press, when any component of the system is not ready or when someone presses the E-stop button installed on the dashboard of the vehicle.

The look-ahead controller system was installed behind the driver seat. This includes all of the above mentioned elements of the look-ahead controller, the low-level controller, safety features and development environment.

After installation, the low-level speed controller needed to be fine-tuned. This was necessary to achieve a controller conforming the following criteria:

- Speed tracking without steady state error.
- Do not reach saturation.
- Jerk free operation
- Minimal fuel consumption
- Dynamic tracking of the optimal speed

The main components of the system as seen in Figure 5 are the truck itself, the two dSpace MicroAutoBox II controllers the on-board computer the GPS/GNSS receiver and the tablet.

The dSpace MicroAutoBox II is a compact automotive prototyping system. It is fitted with several I/O ports of various kinds and it can perform real-time control tasks. The MicroAutoBox II is used in our system to run the optimization algorithm and also to connect to the vehicle systems (or PC in case of the simulations). The second MicroAutoBox II implements the low level speed control functionality described above.

The software code of the MicroAutoBox II is generated from the MATLAB/Simulink model used in the software simulation, with small modifications to use the actual hardware interfaces. The developer software package provided with the controller turns a suitable MATLAB code to an executable for the MicroAutoBox II. When loaded on the system the program automatically starts with the device.

The on-board computer is an industrial PC. Its purpose is connecting the MicroAutoBox II to the internet and providing a user interface for the driver. It connects to the GPS sensor in the final stage of the project. The PC also serves as a data logging device for offline tests and troubleshooting problems.

The tablet is the physical user interface. It only serves as a thin client connecting to the on-board PC. The tablet and the PC is connected by a WiFi router which provides internet access for both devices. The router is capable of connecting to 3G/4G mobile internet services which is necessary later, on the road tests.

The on-board computer serves multiple purposes in this system. For the sake of reliable and easy to maintain operation a modular software framework was developed for this PC. The framework itself provides a standard way to share information between the modules and to coordinate the whole system. Figure 4 demonstrates the design of this framework.

- One module handles the GPS/GNSS receiver;
- One module connects to the online databases;
- An other is responsible for the communication with the MicroAutoBox II device;

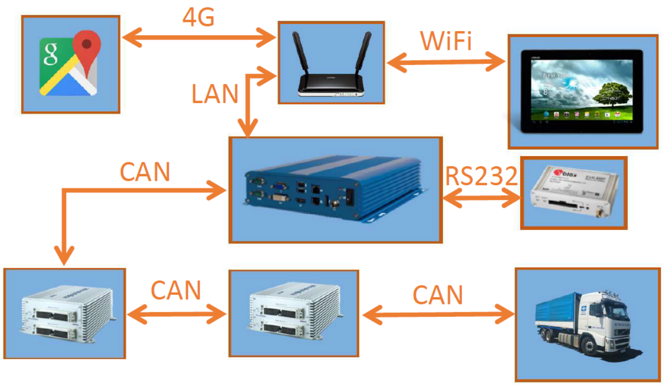

Fig. 5. Architecture of the final implementation

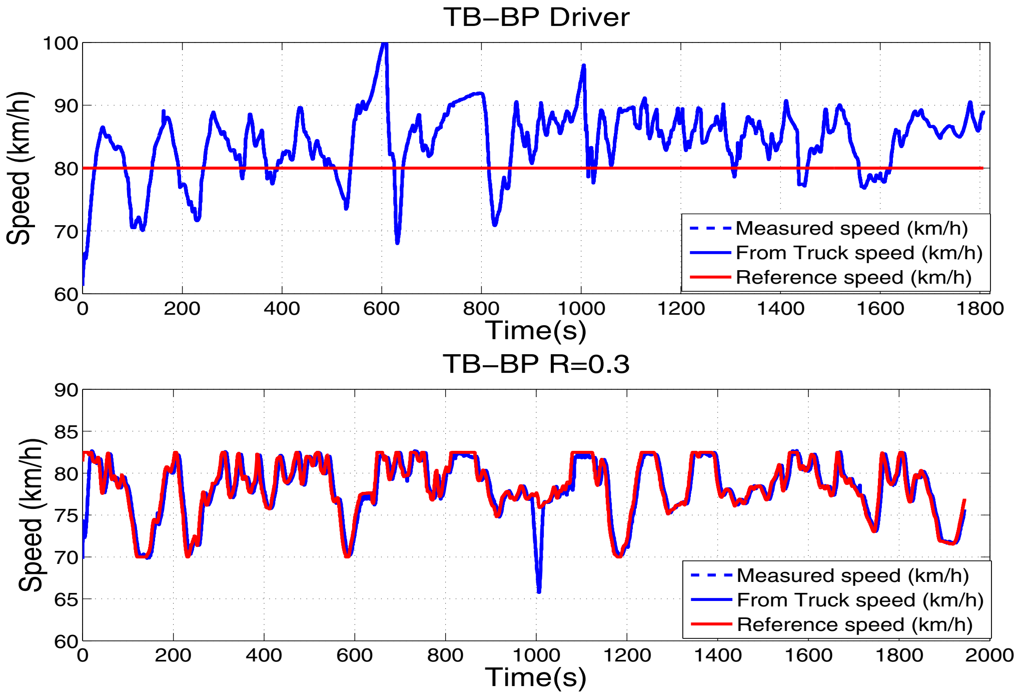

Fig. 6. Vehicle speed during the testruns

- Configurable data logging is also implemented in a different module;
- There is a module for the user interface.
  This last one is comprised of an internal communication part and a web server.

The user interface is web-based, this means it can be displayed locally or on any connected device with a browser capable of running JavaScript. In our case this device is a tablet.

When the driver navigates to the web address of the controller the interface is displayed. This screen differs in functionality from the previously described method of starting the system in a software simulation.

Most of the screen is occupied by a map which is rendered using the Google Maps API. The driver can plan a route to a desired location on this map, either by putting in an address in the search bar or by selecting a spot on the screen. The planned route is immediately transferred to the on-board PC and stored. After the press of the start button the system is engaged and the journey starts. The controller can be stopped and restarted at any time. The speed controller automatically turns off if the driver presses a pedal, in this case the system needs to be re-engaged to continue operation.

The current and calculated optimal speed is displayed beside the map with additional information such as expected travel time and current weather. The R_1 parameter of the algorithm can also be changed here.

5. RESULTS OF THE ROAD TESTS

Road testing of the implemented system was carried out in Hungary, on highway M1. A test route between Budapest and Tatabánya has been selected due to its characteristics. This road segment is approximately $50km$ long and it has inclinations on which the look-ahead algorithm can be demonstrated.

The test runs were made at night to minimize the effects of the traffic on the measurements, however as this is one of the busiest highways of the country the number of other vehicles on the road was still significant. The effects can be observed in Figure 6 as dips in the speed when the driver used the brakes due to congestions. There has been several round trips with and without the use of the controller and with different parameter settings.

The outcomes were matched with the reference runs without the controller. On these runs a professional truck driver drove all the way without specific orders, as he normally would.

The registered speed-profile and fuel-consumption diagrams of a driver's trip can be examined in Figures 6 (a) and 7 (a) respectively. It can be seen that the driver did not maintain a steady reference speed. The active speed limit at the time was $90km/h$ resulting in a total of $1807s$ travel time. The total fuel consumption was $10.23l$.

Figure 6 shows the Measured speed taken from the GPS sensor and the Truck speed taken from the diagnostic channel of the vehicle. The two measurements are identical.

The test trips using the look-ahead controller, as mentioned before, were carried out using several different settings. The specific parameters were chosen according to the TruckSim simulations done in the first two phases.

Out of the available parameters only R_1 was changed between the tests to find the optimum empirically. The measurements showed that the best setting is $R_1 = 0.3$. This setting resulted in the lowest consumption while maintaining the desired travel time.

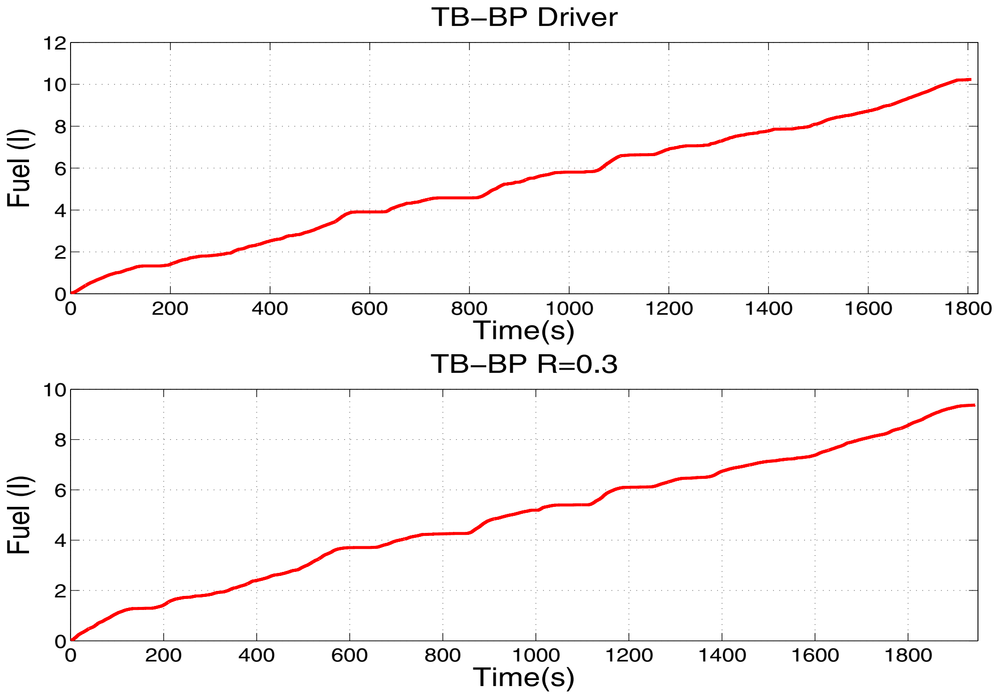

Fig. 7. Fuel consumption during the testruns

For the tests the speed of the truck was limited between 70 and $85 km/h$ to avoid bigger speed-swings which would cause disturbances for other drivers on the road.

The registered speed-profile and fuel-consumption diagrams of the speed controller tests can be seen in Figures 6 (b) and 7 (b) respectively. The test run took a total of $1944s$ to complete. The total fuel consumption was $9.37l$.

As it is shown from the measurements the presented pair of trips differ in total time by roughly 2 minutes. We choose this pair as these two were the closest amongst our measurements.

6. CONCLUSIONS

The conducted tests showed that the proposed look-ahead algorithm works as expected and the implemented system can perform in real-world conditions.

On our test runs the algorithm produced a reduction in consumed fuel measuring to 8.4%. The savings can be partially explained by the increase in travel time, but the saving is still considerable.

This system implemented at a large scale would amount to an improvement in fuel economy on any kind of road vehicles, including trucks and cars as well.

REFERENCES

Alam, A., Martensson, J., and Johansson, K.H. (2013). Look-ahead cruise control for heavy duty vehicle platooning. In *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, 928–935.

Hellström, E., Åslund, J., and Nielsen, L. (2010). Horizon length and fuel equivalents for fuel-optimal look-ahead control. In *Advances in Automotive Control*. Munich.

Ivarsson, M., Åslund, J., and Nielsen, L. (2009). Look ahead control - consequences of a non-linear fuel map on truck fuel consumption. *Proceedings of the Institution of Mechanical Engineers, Part D, Journal of Automobile Engineering*, 223, 1223–1238.

Németh, B. and Gáspár, P. (2013). Design of vehicle cruise control using road inclinations. *International Journal of Vehicle Autonomous Systems*, 11(4), 313–333.

Németh, B. and Gáspár, P. (2017). The relationship between the traffic flow and the look-ahead cruise control. *IEEE Transactions on Intelligent Transportation Systems*, 1154–1164.

Nouveliere, L., Braci, M., Menhour, L., and Luu, H. (2008). Fuel consumption optimization for a city bus. *UKACC Control Conference.*

Passenberg, B., Kock, P., and Stursberg, O. (2009). Combined time and fuel optimal driving of trucks based on a hybrid model. *European Control Conference, Budapest.*

Sahlholm, P. and Johansson, K. (2009). Road grade estimation for look-ahead vehicle control using multiple measurement runs. *Control Engineering Practice, article in press.*