Improving Vehicle Fleet Fuel Economy via Learning Fuel-Efficient Driving Behaviors

Ondrej Linda, *Student Member*, *IEEE*, Milos Manic, *Senior Member*, *IEEE*
Computer Science Department
University of Idaho
Idaho Falls, USA,
olinda@uidaho.edu, misko@ieee.org

Abstract — Reducing the fuel consumption of road vehicles has the potential to decrease environmental impact of transportation as well as achieve significant economical benefits. This paper proposes a novel methodology for improving the fuel economy of vehicle fleets via learning fuel-efficient driving behaviors. Vehicle fleets composed of large number of heavy vehicles routinely perform runs with different drivers over a set of fixed routes. While all drivers might achieve on-time and safe driving performance their actual driving behaviors and the subsequent fuel economy can vary substantially. The proposed Intelligent Driver System (IDS) utilizes vehicle performance data combined with GPS information on fixed routes to incrementally build a model of the historically most fuel efficient driving behavior. During driving, the calculated optimal velocity for specific location is compared to the current vehicle state and a fuzzy logic PD controller is used to compute the optimal control action. The control action can be projected to the drivers via a specialized HMI or used directly as a predictive cruise control to achieve overall fuel economy improvements. The method has been validated on a simulated heavy vehicle model, showing potential for substantial fuel economy improvements.

***Index Terms*— Driving Behaviors, Fuel Economy, Fuzzy Logic Control, Machine Learning, Vehicle Fleet**

I. INTRODUCTION

FUEL economy of road vehicles has been the scope of recent work of many researches due to several important reasons. Firstly, reducing the fuel consumption can lead to substantial economical benefits. Secondly, lowering the amount of spend fossil fuels reduces the carbon footprint of transportation and has significantly positive environmental impact. Finally, for many countries, which are dependent on imported petroleum, fuel efficiency can substantially decrease the dependency on foreign oil. For instance, according to the 30th edition of the Transportation Energy Data Book, the U.S. transportation petroleum usage itself surpasses the entire U.S. petroleum production by 72.5%, resulting in high dependency on foreign oil [1].

The recent advances in intelligent sensing and communications of transportation systems opened the door for many novel ways of achieving improved fuel economy of road vehicles. Several authors explored the possibility of optimizing the near-future vehicle acceleration profile with respect to fuel consumption based on the GPS information and known road topography information. Dijkstra's shortest path algorithm was used for optimizing the predicted driving behavior in [2]. A dynamic programming approach for calculating the look-ahead optimal strategy was presented in [3]. Passenberg et al. proposed to use a model predictive control framework combined with numerical shooting algorithm for achieving both time and fuel optimal driving in [4]. A cruise control reference shaping strategies to improve engine efficiency base on anticipated environmental changes was proposed in [5]. Kock et al. suggested a simple heuristic approach to fuel efficient predictive cruise control based on road topology information in [6]. There also exists significant research work in the area of predictive cruise control for hybrid electric vehicles [7]. Alternatively, the GPS information can be combined with the common vehicle sensors to obtain more accurate road slope estimate that is then used to optimize the fuel consumption of the vehicle [8]. A different approach via traffic signal state broadcasting for improved fuel economy and reduced trip time was presented by Asadi et al. [9].

This paper proposes a novel method for improved fuel economy of vehicle fleets via learning fuel-efficient driving behaviors. Vehicle fleets composed of large number of heavy vehicles frequently perform runs with different drivers over a set of fixed routes (e.g. transportation of employees to work using a park-and-ride bus system). While most drivers typically achieve on-time and safe driving performance their actual driving behaviors (e.g. acceleration profiles) and the subsequent fuel economy might vary substantially. The proposed Intelligent Driver System (IDS) combines vehicle performance data with GPS information to incrementally build a model of historically most fuel efficient driving behavior for a set of fixed routes. During driving, the current vehicle state is compared to the historically optimal vehicle velocity and a fuzzy logic PD controller is used to compute an optimal driving control action. The control action can be used as a guide for the drivers or it can be utilized directly as a predictive cruise control.

The proposed IDS algorithm was experimentally evaluated on an interactive driving simulator of an MCI D-Series bus. The software simulation calculates the longitudinal dynamics of a passenger bus considering engine, braking, gravitational, aerodynamic drag and rolling resistance forces. The collected preliminary results showed potential for significant fuel efficiency improvements of the vehicle fleet.

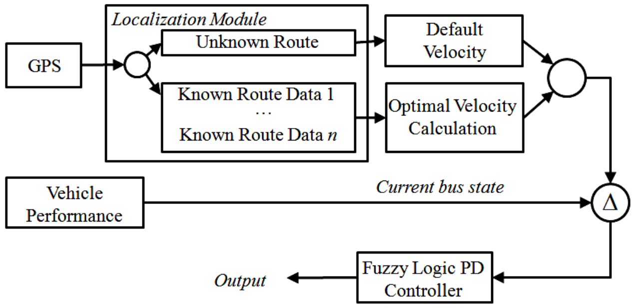


Fig. 1 Architecture of the proposed Intelligent Driver System.

The rest of the paper is organized as follows. Section II describes the proposed Intelligent Driver System. Section III discusses the optimal velocity learning and the prediction of the optimal driving action. Section VI describes the experimental test-bed and presents experimental results and the paper is concluded in Section V.

II. Intelligent Driver System

The fundamental idea of the proposed Intelligent Driver System (IDS) is to collect historical performance data from a pool of drivers for a fixed set of routes, learn from this data the historically optimal driving policy and then inform individual drivers about the optimal driving behavior during future runs. The overall architecture of the proposed IDS is depicted in Fig. 1. The two inputs into the system are the vehicle position information from the GPS receiver and the real-time vehicle performance data obtained from the vehicle's system diagnostic port. The procedure is composed of four main steps as follows: 1) vehicle localization, 2) optimum velocity calculation, 3) comparison of the calculated optimum and the current vehicle state, and 4) optimum control policy calculation.

The vehicle localization is performed by the localization module. The module must determine whether the vehicle is located on a previously visited route. In case of an unknown route, the localization module cannot provide any historical data. For a previously visited route, a set of n recorded data sets on that particular route can be retrieved from the database.

In the optimum velocity calculation step, the IDS computes the historically optimal velocity with respect to fuel economy and to the provided driving constraints (e.g. min/max allowed vehicle speed). For a previously unknown route, the algorithm simply outputs a default (e.g. constant) desired velocity. Hence, for an unknown route, the entire system degrades into a classical cruise control.

The calculated historically optimal vehicle velocity is then compared to the current vehicle velocity. The difference is then supplied to a fuzzy logic Proportional-Derivative (PD) controller. The fuzzy logic controller outputs a suggested change in the control signal (i.e. gas pedal position), which should be applied in order to maintain the historically optimal velocity. The following sections describe individual components of the IDS in more detail.

III. Optimum Velocity Learning

This Section first describes an algorithm for combining two sets of historical data for a fixed route into an optimal driving policy with respect to fuel consumption. Next, a heuristic method for combining multiple sets of historical data into a single fuel efficient driving policy is developed.

A. Segmenting and Merging of Two Vehicle Data Sets

The vehicle's diagnostic port allows for logging of various performance data. As an example, the velocity and the gas pedal position as a function of time recorded from a real MCI D-Series bus are depicted in Fig. 2. Assume that two drivers perform runs with the same type of vehicle over a fixed route. The state of the vehicle $x(t)$ at time t can be defined by its velocity $v(t)$ and gear $g(t)$ as follows:

$$x(t) = [v(t), g(t)] \quad (1)$$

The entire run can then be expressed as a data set $X = \{x(t), t = 0 \ldots T\}$, where T denotes the length of the run. The vehicle state in (1) is expressed as a function of time. In order to allow for synchronization between two different runs on a fixed route, the vehicle state must be transformed into position based form expressed as:

$$x(p) = [v(p), g(p)] \quad (2)$$

Here, p expresses the vehicle position along the fixed route. This transformation can be obtained by calculating vehicle position at specified time intervals and linearly interpolating between equidistantly spaced position steps. The entire run over a specific route can be expressed as $X = \{x(p), p = 0 \ldots L\}$, where L denotes the length of the route.

Given two recorded data sets, a naïve approach for calculating the optimal velocity over the road segment can be implemented by considering the average Miles per Gallon (MPG) indicator of both runs. The naïve method simply selects the driving policy that achieved higher overall MPG. This method considers the entire route as a single compact segment. Despite its simplicity, the major deficiency of this approach is ignoring potential differences of driving behaviors

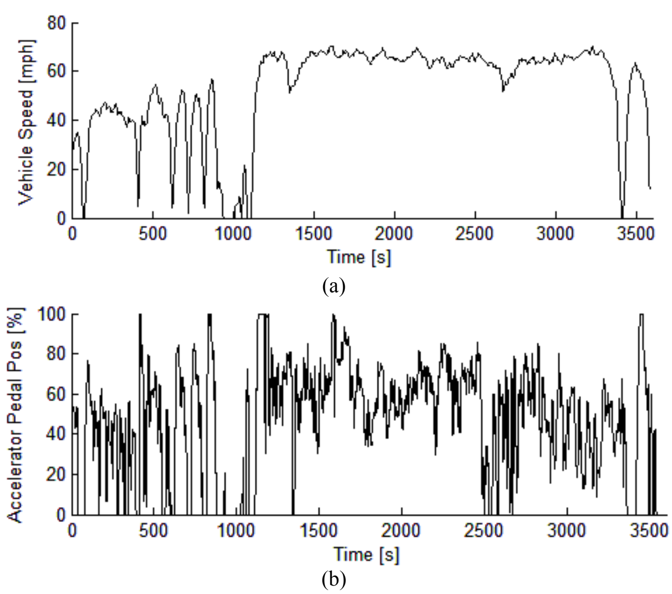


Fig. 2 Velocity (a) and gas pedal position (b) data recorder from a real bus.

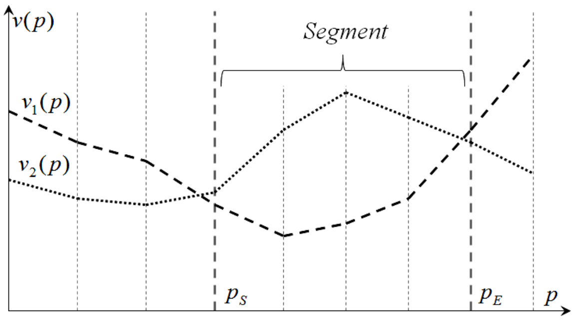


Fig. 3 Segmenting of two vehicle velocities.

in short segments on the road. For instance, some drivers might be better at driving uphill, while others are better at exploiting gravitational force and avoiding braking on a downhill.

The proposed method utilized by the IDS is based on the idea of segmenting the two runs and then choosing the appropriate driving policy for each individual segment. The final result is obtained as a composition of segments from both data sets, allowing for combining different fuel efficient behaviors from both drivers.

A segment $S(p_S, p_E)$ is defined by its starting position p_S and ending position p_E. In order to combine segments from two different runs, the state of the vehicle entering and leaving the segment must be identical in both data sets. Hence, in order for segment S to be located on runs X_1 and X_2 the following proposition must be true:

$$\exists S(p_S, p_E) \Rightarrow x_1(p_S) = x_2(p_S) \wedge x_1(p_E) = x_2(p_E) \qquad (3)$$

Here, $x_1(p) \in X_1$ and $x_2(p) \in X_2$. The definition of a road segment between two vehicle runs presented in (3) ensures that consecutive segments can be connected together without affecting the physical feasibility of the solution. A schematic diagram of a route segment is depicted in Fig. 3. In Fig. 3 it is assumed that both vehicles are in identical transmission gear.

Furthermore, it is assumed that the logged vehicle historical data also contain information regarding the vehicle's fuel rate $f(t)$ at every known vehicle state $x(t)$. The fuel rate $f(t)$ can be also transformed into position based form $f(p)$ via linear interpolation. The amount of consumed fuel $F(p_S, p_E)$ for a segment between positions p_S and p_E can be obtained as follows:

$$F(p_S, p_E) = \int_{p_S}^{p_E} f(p)\, dp \qquad (4)$$

The amount of fuel consumed for particular segment can then be used to guide the selection of the most fuel efficient driving policy.

The pseudo-code of the proposed segmenting and merging algorithm can be described in several steps as follows:

Input: Two vehicle data sets X_1 or X_2 on a fixed route.
Output: Combined data set X_O
Step 1: Transform both runes X_1 or X_2 into position dependent forms using linear interpolation:

$$x(t) \Rightarrow x(p) \qquad (5)$$

Step 2: Form set of common positions $P = \{\hat{p}_i\}$ such that:

$$\forall i \;\; x_1(\hat{p}_i) = x_2(\hat{p}_i),\;\; x_1(\hat{p}_i) \in X_1, x_2(\hat{p}_i) \in X_2 \qquad (6)$$

Note that $i = 1 \ldots N$, where N is the number of common points between runs X_1 and X_2 identified in Step 2.

Step 3: Create a binary vector $V = \{v_1, \ldots v_{N+1}\}$, such that:

$$v_i = \begin{cases} 0 & \text{if } F_1(\hat{p}_{i-1}, \hat{p}_i) \le F_2(\hat{p}_{i-1}, \hat{p}_i) \\ 1 & \text{if } F_1(\hat{p}_{i-1}, \hat{p}_i) > F_2(\hat{p}_{i-1}, \hat{p}_i) \end{cases} \qquad (7)$$

Note that the fuel consumptions F_1 and F_2 are computed based on the historical data from runs X_1 and X_2. Furthermore, it should be noted that points p_0 and p_{N+1} correspond to the beginning and the end of the route. Finally, the preference of X_1 to X_2 based on the consumed fuel can be subject to additional constraints, e.g. the minimum and maximum velocity that the vehicle should attain. Hence, if data set X_1 contains velocity which exceeds the specified allowed maximum velocity in the given segment, data set X_2 will be selected irrespective of the amount of consumed fuel.

Step 4: Create an optimal vehicle state sequence $X_O = \{x_O(p)\}$ as follows:

$$x_O(p) = \begin{cases} x_1(p) & \text{if } v_{g(p)} = 0 \\ x_2(p) & \text{if } v_{g(p)} = 1 \end{cases} \qquad (8)$$

Here, function $g(p)$ finds the pair of common switch points $\hat{p}_{i-1}$ and $\hat{p}_i$ that fulfill the following condition:

$$\hat{p}_{i-1} \le p < \hat{p}_i \qquad (9)$$

The above presented algorithm takes a pair of vehicle data sets on a fixed route and outputs a merged optimized sequence of vehicle states. Hence the algorithm can be described as a function with two inputs and one output, $X_O = Merge(X_1, X_2)$. Over time, the knowledge base of the IDS can contain many recorded runs over a specific fixed route. In order to compute the optimum driving sequence X_O based on $n > 2$ runs, an order of applying the merging algorithm *Merge* must be defined. The final fuel consumption of the resulting driving trajectory X_O will depend on the order in which individual data sets have been merged. This is due to the fact that unique sequence of common points $P = \{\hat{p}_i\}$ likely exists for all pairs of data sets.

B. Heuristic for Merging Multiple Data Sets

For a set of $n > 2$ recorded data sets on a fixed route, there exists a specific order of applying the *Merge* algorithm to individual data set that would yield the most fuel efficient driving policy. This merged most fuel efficient driving policy minimizes the total amount of consumed fuel over all possible ordering of applying the *Merge* operation. Unfortunately, this optimization problem can be only solved by enumerating all permutations of data set indexes which accounts for $n!$ combinations. This number is prohibitively large even for modest number of historical runs.

To alleviate this problem an approximate heuristic approach is proposed, which does not guarantee global optimal solution, but it can yield a solution with high fuel efficiency in a reasonable time. The pseudo-code of this heuristic method can be described as follows:

Input: n vehicle data sets on a fixed route $X_1,...,X_n$.

Output: Optimized run X_O

Step 1: Construct matrix Ω where each element i, j stores fuel consumption F of the merged vehicle trajectory computed using the $Merge(X_i, X_j)$ method:

$$\Omega = \begin{vmatrix} F(X_1) & F(Merge(X_1,X_2)) & \cdots & F(Merge(X_1,X_n)) \\ F(Merge(X_1,X_2)) & F(X_2) & \cdots & F(Merge(X_1,X_n)) \\ \vdots & \cdots & \ddots & \vdots \\ F(Merge(X_1,X_n)) & F(Merge(X_2,X_n)) & \cdots & F(X_n) \end{vmatrix} \tag{10}$$

Note that in order to obtain matrix Ω only elements under (or above) diagonal must be computed, since the merging operation constitutes s symmetric relation.

Step 2: Find element $\Omega_{i,j}$ such that:

$$\Omega_{i,j} = \arg\min_{i,j}(F(X_i, X_j)) \tag{11}$$

Step 3: Store pair X_i and X_j for merging and remove the i^{th} row and j^{th} column from matrix Ω.

Step 4: Repeat **Step 2** and **Step 3** until matrix Ω becomes empty.

Step 5: Use the *Merge* algorithm to merge all selected pairs of data sets. The result will be a new set of merged data sets $X'_1,...,X'_m$, where $m = \lceil n/2 \rceil$.

Step 6: Start from **Step 1** with the new set of m data sets.

Step 7: Repeat the above procedure until single data set X_O remains.

Note the there will be total of $\lceil \log_2 n \rceil$ repetitions of the procedure in **Step 1** – **Step 6**, since each procedure reduces the number of remaining vehicle trajectories by factor of 2. While the presented heuristic method is not optimal, it calculates a fuel efficient solution by sequentially paring vehicle data sets, which are likely to yield the best fuel efficiency.

IV. Predicting Optimal Driving Action

The proposed Intelligent Driver System was designed for learning the most fuel efficient driving behaviors and computing the optimal driving policy on a fixed route. In the described system, the calculated control action can be projected to the drivers via a specialized HMI or used directly as a predictive cruise control to achieve overall fuel economy improvements. This Section describes the calculation of the optimal control action using a fuzzy PD controller.

A. Fuzzy Logic Control

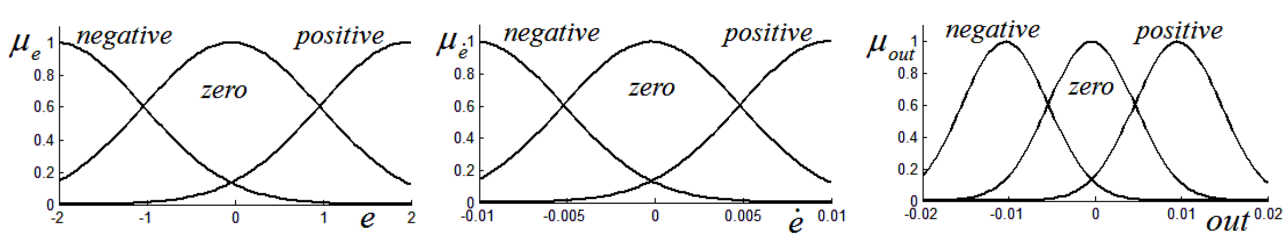
Fuzzy Logic Controllers (FLCs) have been successfully applied to many engineering problems [10], [11]. One of the most widely recognized advantages of FLCs is their capability to codify human knowledge in the form linguistic fuzzy rules. These linguistic fuzzy rules can be intuitively understood and constructed by humans. In addition, FLCs can cope with ambiguity, imprecision and uncertainty in linguistic expressions via appropriate modeling of linguistic terms using fuzzy sets [11].

In general, an FLC can be decomposed into four major parts – input fuzzification, fuzzy inference engine, fuzzy rule base and output defuzzification [11]. In this work, the Mamdani-type FLC is considered. Mamdani type FLC maintains a fuzzy rule base populated with fuzzy linguistic rules in an implicative form. Rule R_k can be described as follows:

$$\textbf{IF } x_1 \text{ is } A_1^k \textbf{ AND } \ldots \textbf{ AND } x_n \text{ is } A_n^k \textbf{ THEN } y_k \text{ is } B^k \tag{12}$$

Here, symbol A_j^k and B^k denote the j^{th} input fuzzy set and the output fuzzy set, n is the dimensionality of the input vector $\vec{x}$, and y_k is the associated output variable. The input vector is first fuzzified using the fuzzy membership function (e.g. Gaussian, triangular or trapezoidal). The fuzzification of input x_i into fuzzy set A_i^k results in a fuzzy membership grade $\mu_{A_i^k}(x_i)$. Using the minimum t-norm operator, the degree of firing of rule R_k can be computed as:

$$\mu_{R_k}(\vec{x}) = \arg\min_{i=1..n}\{\mu_{A_i^k}(x_i)\} \tag{13}$$



(a) (b) (c)

Fig. 4 Input and output fuzzy sets of the fuzzy PD controller

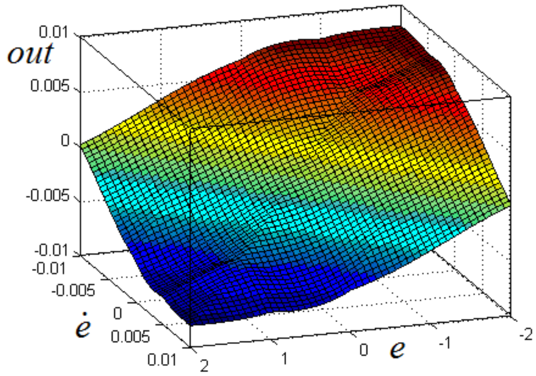


Fig. 5 Output control surface of the fuzzy PD controller.

TABLE I
FUZZY RULE TABLE

	$\dot{e}_{neg}$	$\dot{e}_{zero}$	$\dot{e}_{pos}$
e_{neg}	*Positive*	*Positive*	*Zero*
e_{zero}	*Positive*	*Zero*	*Negative*
e_{pos}	*Zero*	*Negative*	*Negative*

The output of each fuzzy rule is computed by applying the rule firing strength via the t-norm operator (e.g. minimum or product) to the associated rule consequent. The fired output fuzzy sets from all rules are aggregated using the t-conorm operator (e.g. the maximum operator), resulting in an output fuzzy set *B*. For further details on the fuzzy inference process please refer to [10], [11].

Finally, the defuzzification of the output fuzzy set *B* yields a output value *y*. Here, the centroid defuzzifier was used [11]. For an output domain discretized into *N* samples the crisp output value *y* is obtained as:

$$y = \frac{\sum_{i=1}^{N} y_i \mu_B(y_i)}{\sum_{i=1}^{N} \mu_B(y_i)} \tag{14}$$

B. Fuzzy PD Controller for Driving Action Calculation

For the optimal driving action calculation, the fuzzy Proportional-Derivative (PD) controller was utilized. Here, the control signal *out*(*t*) at time *t* is proportional to the measured error signal *e*(*t*) and its time derivative $\dot{e}(t)$. The error is calculated as the difference between the actual vehicle velocity and the calculated optimal velocity $v(t)$ and $v_O(t)$.

The fuzzy PD controller has been adopted in the proposed system for its several advantages that can be summarized as follows: i) it can be constructed using linguistic knowledge about the problem domain, ii) it features more design degrees of freedom than classical PD controllers, and iii) it was shown to produce smoother and more robust control behaviors [12].

In this paper, the fuzzy controller was designed using three Gaussian input fuzzy sets {*negative*, *zero*, *positive*} for describing the input signals error *e* and error derivative $\dot{e}$, as shown in Fig. 4(a) and 4Fig. (b) The controller's output signal *out* was modeled using three Gaussian output fuzzy sets {*negative*, *zero*, *positive*}, shown in Fig. 4(c). The fuzzy rule base presented in Table I provides nine linguistic fuzzy rules for the optimal driving action control. This rule table constitutes a commonly adopted set of rules for fuzzy PD controllers. The fuzzy PD controller can be understood as a composition of multiple classical PD controllers with variable gains [12]. The resulting output control surface is visualized in Fig. 5.

C. Driver's Input

The calculated optimal control action constitutes the suggested change from the current control input. The fuzzy PD controller compares the current vehicle velocity with the calculated historically optimal velocity for the specific location of the vehicle and computes the desired control action, i.e. gas pedal position. Hence, the fuzzy PD controller smoothly guides the driver towards the historically optimal velocity profile.

The calculated control action can be used in two different ways. First, it can be used as a part of a predictive cruise control, which directly controls the acceleration profile of the vehicle instead of the driver. Second, it can be projected to the drivers via a dedicated HMI to guide them towards the optimum driving policy. The design and testing of the HMI is a scope of an ongoing research effort. An example of an HMI prototype can be seen in Fig. 6. This simple HMI informs the drivers about the current gas pedal position and about the suggested gas pedal position.

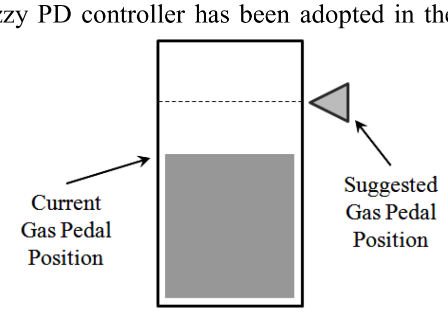


Fig. 6 Intelligent Driver System HMI.

V. EXPERIMENTAL RESULTS

The feasibility of the proposed Intelligent Driver System was evaluated on an interactive simulation of the MCI D-Series bus. This Section describes the design of the experimental test bed and the preliminary experimental testing.

A. Vehicle Modeling

The implemented model simulates the longitudinal dynamics of MCI D-Series bus. The model is based on Newton's second law stated as:

$$m\dot{v} = F \tag{15}$$

The implemented simulation considers several forces that affect the vehicle's state. The considered forces are the engine force F_e, the aerodynamic drag force F_a, the rolling resistance force F_r, and the gravitational force F_g. After substituting the individual forces into (15) the following expression can be obtained:

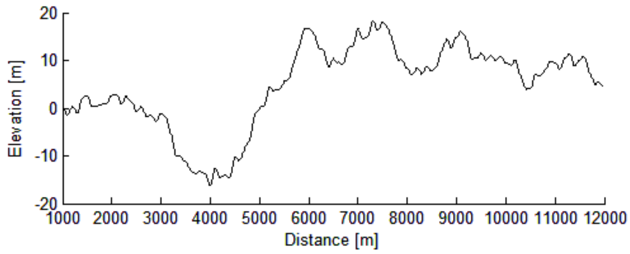


Fig. 7 Testing route topography.

TABLE III
MPG MATRIX

	Driver 1	Driver 2	Driver 3	Driver 4
Driver 1	*N/A*	*10.37*	*9.48*	*9.87*
Driver 2	10.37	*N/A*	*10.29*	*10.36*
Driver 3	*9.48*	*10..29*	*N/A*	*9.27*
Driver 4	*9.87*	*10.36*	*9.27*	*N/A*

$$m\dot{v} = F_e - F_a - F_r - F_g \tag{16}$$

The engine force F_e is approximated using the torque function $T(P, G)$ as follows:

$$F_e = \frac{i_t \, i_f \, T(P,G)}{r_w}, \tag{17}$$

Here, P is the acceleration pedal position, G is the current gear, r_w is the wheel radius and i_t and i_f are the current gear and the final drive conversion ratios. The torque function has been modeled based on the technical specification of the vehicle's engine.

The aerodynamic force F_a is estimated based on the following equation:

$$F_a = \frac{1}{2} c_w A_a \rho_a v^2, \tag{18}$$

TABLE II
DRIVER PERFORMANCE

	Time [s]	MPG
Driver 1	*356*	*9.48*
Driver 2	*360*	*10.25*
Driver 3	*341*	*8.50*
Driver 4	*356*	*9.27*
Optimum	*358*	*10.44*

Here, c_w is the air drag coefficient of the vehicle, A_a is the front cross section area of the vehicle and v denotes the velocity of the vehicle.

The rolling resistance force F_r is approximated using a friction force as:

$$F_r = c_r m g \cos\alpha, \tag{19}$$

Here, c_r is the rolling resistance coefficient, m is the vehicle mass and α denotes the road slope.

Finally, the gravitational force affects the vehicle according to:

$$F_g = m g \sin\alpha, \tag{20}$$

Here again, symbol α denotes the slope of the road.

In order to compute accurate fuel consumption of the simulated vehicle a data-driven approach was preferred to analytical approaches. The gathered historical data from real MCI D-Series bus runs were used to create a functional mapping between the vehicles engine rpm, the gas pedal position and the fuel rate of the engine. To create a mathematical model based on the data, the Adaptive-Network-Based Fuzzy Inference Systems (ANFIS) was used [13]. The ANFIS was trained with 3 Gaussian fuzzy sets for each input and with grid partition of the input space resulting in nine fuzzy rules. A detailed description of ANFIS can be found in [13].

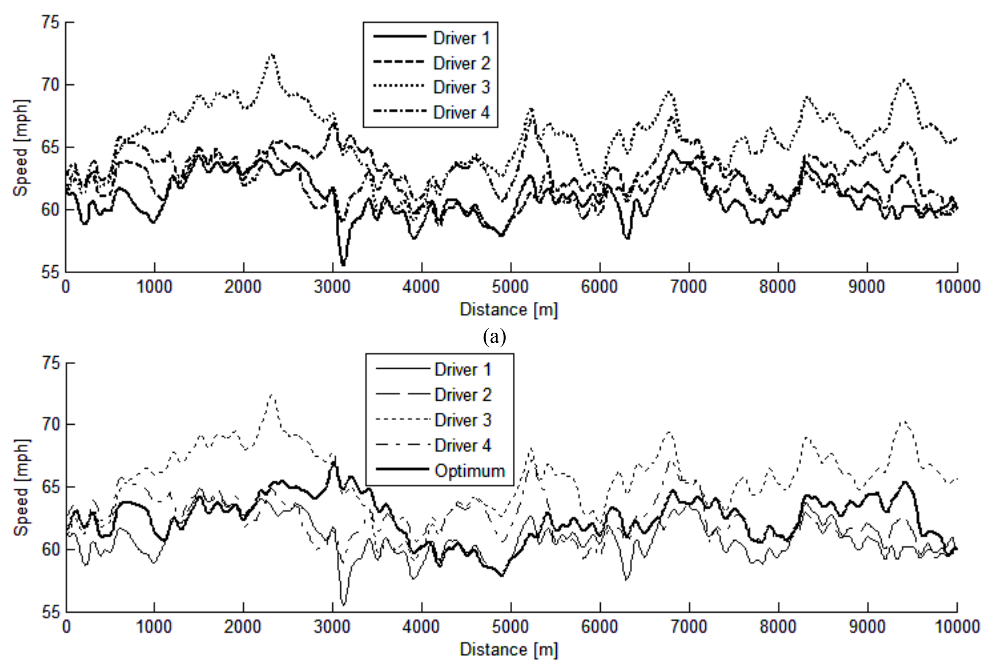


Fig. 8 Recorded data from 4 drivers (a) and the calculated optimal velocity profile (b).

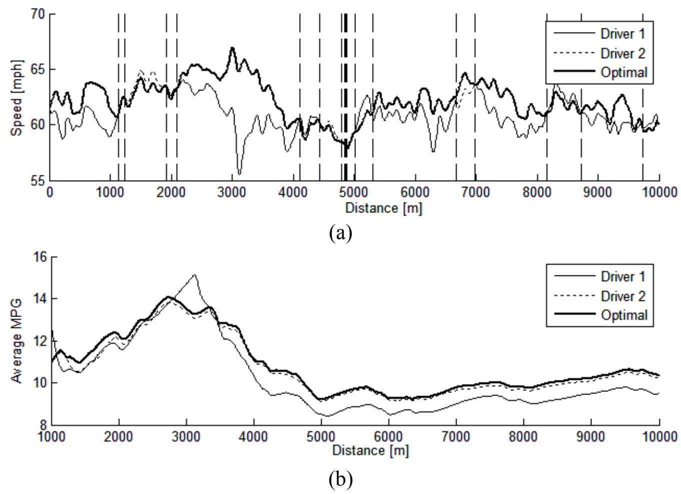


Fig. 9 Merged velocity profile with depicted switch points (dashed vertical lines) (a) and the resulting optimal MPG (b).

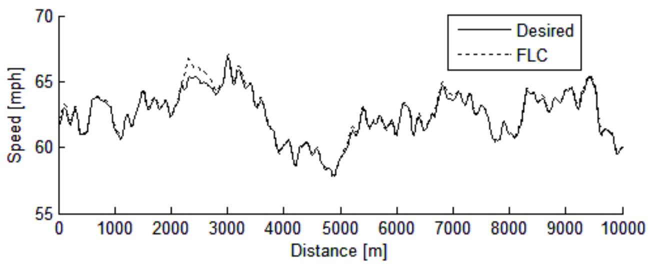


Fig. 10 Comparison of the desired optimal velocity and the FLC controlled vehicle velocity.

B. Experimental Testing

The developed IDS was evaluated in the implemented driving simulator. The driving performance of four different drivers was recorded on a 10km long road segment. The used randomly generated route topography is depicted in Fig. 7. The recorded drivers' velocity profiles are shown in Fig. 8(a). Table II lists the achieved time and the average MPG of each run.

The optimum velocity learning algorithm described in Section III was applied to the four recorded driving data sets. In the first stage of the algorithm, the MPG of merging each possible pair of data sets was calculated. The resulting matrix is shown in Table III. It can be observed that merging data sets from Driver 1 and Driver 2 yields the highest MPG. As an example the switch points and the merged velocity data sets for Driver 1 and Driver 2 are depicted in Fig. 9.

The proposed heuristic method was used to sequentially merge the velocity trajectories until the historically most fuel efficient driving velocity profile for the specific route is obtained. Hence, in the first stage of the method the algorithm merges data from Driver 1 and Driver 2 and data from Driver 3 and Driver 4. In the next stage the two new data sets are merged together, yielding the final result. The resulting velocity is depicted in Fig. 8(b). The predicted historically optimal MPG for the given route is 10.44, which shows 0.19 MPG improvement over the historically most fuel efficient driver, as shown in Table II.

The fuzzy PD controller was then used to calculate the optimal driving action and directly control the vehicle's acceleration. Comparison of the desired and the actual velocity is shown in Fig. 10. It can be observed that the designed fuzzy PD controller can achieve good tracking of the desired velocity. The actual MPG of the final run was 10.65. This final MPG further exceeds the expected optimal MPG, which is likely due to smoothing of the driving behavior using the fuzzy logic PD controller as opposed to real human driver.

VI. Conclusion

This paper presented a novel methodology for improving the fuel consumption of vehicle fleets via learning fuel-efficient driving behaviors. The main assumption used by the proposed system was that vehicle fleets are composed of a number of vehicles that routinely perform runs with different drivers over a set of fixed routes. The proposed Intelligent Driver System utilized vehicle performance data combined with GPS information on fixed routes to incrementally build a model of the historically most fuel efficient driving behavior. The calculated optimal velocity was then projected to the drivers or used as predictive cruise control to achieve overall fuel economy improvements.

In order to evaluate the performance of the proposed method, a simulated heavy vehicle model was created. The initial test showed potential for substantial fuel economy improvements of the vehicle fleet.

References

[1] S. C. Davis, S. W. Diegel, R. G. Boundy, "30th Edition Transportation Energy Data Book", Oak Ridge National Laboratory report, 2011.

[2] S. Park, H Rakha, K. Ahn, K. Moran, "Predictive Eco-Cruise Control: Algorithm and Potential Benefits," in *Proc. IEEE Forum on Integrated and Sustainable Transportation Systems*, pp. 394-399, June 2011.

[3] E. Hellstrom, M. Ivarsson, J. Aslund, L. Nielsen, "Look-Ahead Control For Heavy Trucks to Minimize Trip Time and Fuel Consumption," in *Control Engineering Practice*, vol. 17, issue: 2, pp. 245-254, Feb. 2009.

[4] B. Passenberg, P. Kock, O. Stursberg, "Combined Time and Fuel Optimal Driving of Trucks based on a Hybrid Model, " in *Proc. of European Control Conference*, pp. 4955-4960, Aug. 2009.

[5] A. M. Ari, S. Koskie, Y. Chen, "Novel Optimization Strategy to Improve Fuel Economy for Heavy-Duty Trucks," in *Proc. 2010 IEEE International Conference on Vehicular Electronics and Safety*, pp. 122-127, July 2010.

[6] P. Kock, S. Gnatzig, B. Passenberg, O. Stursberg, A. W. Ordys, "Improved Cruise Control for Heavy Trucks using combined Heuristic and Predictive Control," in *Proc. of 2008 IEEE Int. Conf. on Control Applications*, 2008.

[7] T. Van Keulen, G. Naus, B. de Jager, R. van de Molengraft, M. Steinbuch, E. Aneke, "Predictive Cruise Control in Hybrid Electric Vehicles," in *World Electric Vehicle Journal*, vol. 3, 2009.

[8] P. Sahlholm, K. H. Johansson, "Segmented road grade estimation for fuel efficient heavy duty vehicles," in *Proc. 49th IEEE Conf. on Decision and Control*, pp. 1045-1050, Dec. 2010.

[9] B. Asadi, A. Vahidi, "Predictive Cruise Control: Utilizing Upcoming Traffic Signal Information for Improving Fuel Economy and Reducing Trip Time," in *IEEE Trans. on Control Systems Technology*, vol. 19, issue: 3, pp. 707-714, May 2011.

[10] G. Klir, B. Yuan, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice Hall, Upper Saddle River, NJ, 1995.

[11] J. M. Mendel, *Uncertain Rule-Based Fuzzy Logic Systems: Introduction and New Directions*, Prentice-Hall, Upper Saddle River, NJ, 2001.

[12] X. Du, H. Ying, "Derivation and Analysis of the Analytical Structures of the Interval Type-2 Fuzzy-PI and PD Controllers, " in *IEEE Trans. on Fuzzy Systems*, vol. 18, no. 4, pp. 802-814, August 2010.

[13] J.-S. R. Jang, "ANFIS: Adaptive-Network-Based Fuzzy Inference System," *IEEE Trans. Syst., Man, and Cybern.*, vol. 23, no. 3, 1993.