EcoDrive: A Mobile Sensing and Control System for Fuel Efficient Driving

Lei Kang, Bozhao Qi, Dan Janecek and Suman Banerjee
University of Wisconsin-Madison
{lkang, bozhao, suman}@cs.wisc.edu , janecek@wisc.edu

ABSTRACT

This paper introduces EcoDrive, a fuel consumption sensing and control system for modern vehicles, implemented in an embedded platform, to improve fuel efficiency and reduce carbon emissions. EcoDrive senses vehicle dynamics through the standard vehicle On-board diagnostics (OBD) port and models various vehicle forces, i.e., propulsion, drivetrain loss, wind resistance and grade resistance, as functions of instant fuel consumption. By sensing vehicular speed and controlling air/fuel injection rate in real time, EcoDrive can adjust speed carefully to improve fuel efficiency.

We have collected more than 10,000 miles of driving traces from 12 different vehicles to build models of vehicle dynamics. Based on the models, a prototype of EcoDrive is implemented in an off-the-shelf embedded platform. The prototype is installed on a regular vehicle and evaluated through test drives of more than 100 miles across both urban and highway environments. In comparison with human drivers, EcoDrive achieves an average of 20% higher fuel efficiency in urban road segments and 30% higher fuel efficiency on highways.

1. INTRODUCTION

Modern vehicles are the ultimate mobile computing platforms. They are often on the move at significant speeds and are equipped with significant embedded computing systems that control and manage different functions including providing various types of assistance to its driving function [1, 2, 3, 4, 5]. The computational capabilities of automobile systems provide opportunities and challenges to manage and control every aspects of a vehicle's drive. The focus of this work has been in the design of a system to improve the fuel efficiency of a vehicle's drive by sensing, computing, and actuating the acceleration behavior of the vehicle in an autonomous manner, by modeling properties of the vehicle, road conditions, and driving actions. The benefits of such a system can be profound — Morgan Stanley reported that there could be $158 billion in annual savings in the US, if all cars adopted a smoother driving style [6]. With the global push for improving fuel efficiency of vehicles to reduce consumptions and carbon emissions, we believe solution such as ours can be one of many important mechanisms to meet such a goal.

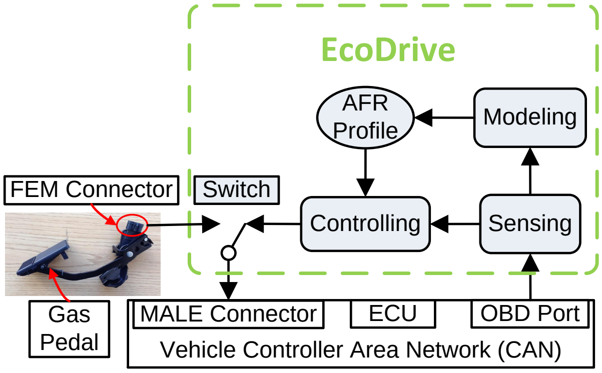

Figure 1: EcoDrive Architecture.

System and Assumptions. This paper introduces EcoDrive, a fuel consumption sensing and control system that improves fuel efficiency and reduces carbon emissions. EcoDrive estimates instant fuel consumptions of different driving behaviors based on sensed vehicle parameters from the On-board diagnostics (OBD) port [7, 8]. It can adjust vehicular speed in real time according to individual vehicle properties and road conditions to achieve higher fuel efficiency measured by Kilometer Per Liter (KPL) [1]. EcoDrive is an independent system that can be installed on or removed from regular vehicles easily. This system controls the vehicle's acceleration and speed to provide a fuel efficient drive on its path. In our work and current implementation, we design this system assuming there is no other factors that would contribute to a choice of acceleration and speed, e.g., other vehicles, pedestrians, etc. or other obstacles in vicinity. Clearly in a practical system, this knowledge would be critical in modifying the acceleration behavior. Currently, we adopt the approach followed by other equivalent systems, such as cruise control [9], which allows the driver to instantly disable cruise control by actively pressing the brake pedal. In an analogous way, in our current implementation, we provide the driver a switch which can be pressed to instantly

MobiCom'15, September 7–11, 2015, Paris, France.

DOI: http://dx.doi.org/10.1145/2789168.2790111.

---

[1] KPL refers to the distance travelled per unit volume of fuel consumed. It interchangeable with Mile Per Gallon (MPG), i.e., 1 KPL = 2.35214583 MPG.

disable EcoDrive, if its acceleration behavior is perceived to be unsafe for nearby vehicles or obstacles.

EcoDrive Components. EcoDrive delivers its design via three components including an OBD sensing component, a vehicle dynamics modeling component and an acceleration controlling component. The architecture is illustrated in Fig. 1. The sensing component reads real-time OBD parameters through the OBD port. The modeling component models various vehicle forces as functions of instant fuel consumption and produces a fuel consumption profile, called Air/Fuel Rate (AFR) profile [2]. The controlling component utilizes the AFR profile to calculate fuel efficient driving strategies according to speed limit and road conditions. EcoDrive emulates the gas pedal by sending voltage values to the connector through an Arduino board. The vehicular Electronic Control Unit (ECU) controls air/fuel injection rate according to the voltage inputs.

EcoDrive addresses two main challenges.

a) Model Vehicle Dynamics based on OBD Parameters. EcoDrive uses the OBD parameters delivered by the sensing component to build an AFR profile, which records instant fuel consumptions of various accelerations under different speeds. To this end, EcoDrive models various vehicle forces, including propulsion, drivetrain loss, wind resistance and grade resistance, as functions of instant fuel consumption. First, we model propulsion (or output torque) as a function of engine torque and gear ratio [10, 11]. We use AFR to model engine torque, and use the ratio between RPM and vehicular speed to model gear ratio. Second, we represent drivetrain loss and wind resistance as a function of vehicular speed [12]. The coefficients are estimated from recorded data traces where the car is driving at constant speeds. The basic idea is that the sum of resistances is equal to propulsion when the car is driving at a steady-state speed. Third, we model grade resistance by altitude changes over road segments. The altitudes of the locations are obtained from National Elevation Dataset [13]. Based on the three models, AFR is modeled as a function of speed, acceleration and road conditions.

b) Control Air/Fuel Rate and Vehicular Speed to Improve Fuel Efficiency. EcoDrive controls vehicular speed by emulating gas pedal. It sends the emulated gas pedal position values to the Arduino board which then converts the position values to corresponding output voltages. The output voltages are delivered to the ECU through a 6 Pin Connector. The problem is how to adjust vehicular speeds to travel through a certain distance with the lowest fuel consumption. We solve this problem by using dynamic programming. Each state of the dynamic programming model records the minimum air/fuel cost that allows the car to achieve the current speed at the current location. In this model, speed can only increase and the last state of each speed records the minimum fuel consumption if the car reaches the pre-assigned distance at that speed. We call the speed with minimum fuel consumption the target speed. By backtracking the state matrix from the last state of target speed, EcoDrive can obtain the desired AFR at each speed. EcoDrive adjusts air/fuel injection rate based on real-time sensed vehicular speed. Once the vehicular speed reaches the target speed, EcoDrive enters a cruising state and commands a constant air/fuel injection rate until the car reaches the pre-assigned distance.

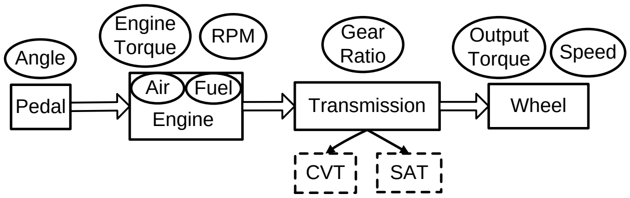

Figure 2: Vehicle Drivetrain.

EcoDrive Prototype. We build a prototype of EcoDrive in an off-the-shelf mobile embedded platform. The prototype is installed and tested on a 2011 Chevrolet Impala. We test EcoDrive on the Impala for more than 100 miles in both urban and highway environments. We evaluate the fuel consumption of EcoDrive and human drivers on different road segments. In urban areas, EcoDrive achieves 10%-40% higher fuel efficiency than four recruited human drivers. On highway, we evaluate EcoDrive on two highway segments with different target speeds. In our user tests, EcoDrive has over 30% improvements compared to different human drivers. In comparison with cruise control, which is more fuel efficient than human drivers in the traces we collected, EcoDrive achieves an average of 10% higher fuel efficiency. We evaluate the performance of EcoDrive on other vehicles by using trace-driven simulation based on the 10,000 miles data collected from 12 different vehicles. We further find that instant fuel economy display on regular vehicles is misleading and cruise control is fuel consuming during speed changes (either requested by user or affected by road conditions).

Contributions. EcoDrive is an independent fuel consumption sensing and control system that can improve fuel efficiency. The system is implemented on an embedded platform that can be easily installed on regular vehicles. We model various vehicle forces as functions of instant fuel consumption and the models are evaluated by utilizing 10,000 miles of driving traces collected from 12 vehicles. EcoDrive is installed on a regular vehicle and evaluated by more than 100 miles of driving in both urban and highway environments, and demonstrated to improve fuel efficiency compared to cruise control system of the vehicle and to human drivers.

2. PRELIMINARY

To illustrate how vehicle system works and how each components are related to different OBD parameters, we present a high-level vehicle system control flow in Fig. 2. Drivers accelerate vehicle by pressing the gas pedal, and then the gas pedal position will be sent to the Electronic Control Unit (ECU). The ECU controls air/fuel injection rate to produce engine torque to drive the vehicle. Transmission is used in this process to transit power from engine to wheel and match engine rotational speed with wheel rotational speed.

2.1 Power Production and Transition

In a car engine, the explosion of compressed fuel and air produces the power to drive a car. The air is injected through an intake port and expelled from an exhaust port. The volume of oxygen in the exhaust air is monitored by fuel trim sensors. Based on fuel trim sensor readings, the

---

[2] AFR refers to the volume of air/fuel cost per unit time.

Electronic Control Unit (ECU) calculate the desired fuel injection rate to maintain the ideal air/fuel ratio [3].

The burning of air and fuel mixture triggers the revolution of crankshaft, which carries piston power out of the engine to the transmission. The transmission carries power to wheels. Transmission is also known as gearbox that uses gears and gear trains to provide speed and torque conversion. The transmission reduces the higher engine speed to the slower wheel speed and increases torque in the process of accelerating the vehicle. The transmission converts engine revolutions to driveshaft revolutions, and then to axle revolutions. A Continuously Variable Transmission (CVT) is a transmission that can change seamlessly through an infinite number of effective gear ratios between maximum and minimum values. This contrasts with Step Automatic Transmissions (SAT) that offer a fixed number of gear ratios.

2.2 OBD Parameters

Table 1: Collected OBD Data

Name	PID	Unit
Fuel Level (FL)	2F	%
Mass Air Flow Rate (MAF)	10	g/s
Fuel System Status (FSS)	03	Bitmask
Long Term Fuel Trim (LTFT)	07	%
Short Term Fuel Trim (STFT)	06	%
Vehicular Speed (VS)	0D	km/h
Engine RPM (RPM)	0C	rpm
Accelerator Position (AP)	5A	%

On-board diagnostics (OBD) is an automotive term referring to a vehicle's self-diagnostic and reporting capability. It is the interface between car Controller Area Network (or CAN bus) and external devices, e.g., a OBD scan tool that connects the OBD port and a laptop. Car CAN bus allows vehicular components to communicate with each other. For example, a position value is sent to the Electronic Control Unit (ECU) after human driver press the gas pedal, and the ECU adjusts air/fuel injections according to the position value. Therefore, we can read this position message transmitted over CAN bus from the OBD port.

The data we collected from the OBD port are summarized in Table 1. Fuel Level (FL) of the vehicle is usually measured by a float sensor, which is usually visualized on the fuel gauge in the car [14]. The FL is calculated according to the height of the float. The Fuel System Status (FSS) is used to indicate current engine mode, open loop mode or closed loop mode. In open loop mode, the engine calculate the fuel injection based on the pre-calculated table and there is no feedback to the engine to adjust the fuel injection rate. The engine runs in open loop mode during short warm up time and runs in closed loop mode at most of the time. In closed loop mode, the engine adjusts fuel injection rate based on fuel trims and air flow rate. Mass Air Flow Rate (MAF) is used to measure the air intake rate of the engine, and therefore an effective indicator of instant fuel consumption [15, 16, 17]. Both Long Term Fuel Trim (LTFT) and Short Term Fuel Trim (STFT) are used to adjust the air/fuel rate injected into engine. LTFT changes less frequent than STFT. In this paper, we use AFR as a metric of instant fuel consumption and/or carbon emission.

Table 2: Vehicles (distance in miles)

No.	Car Model	Urban	Highway
1	Chevrolet Impala 2011	1051	852
2	Nissan Rogue 2011	1198	1063
3	Subaru Forester 2011	651	757
4	Buick LaCrosse 2006	599	649
5	Volkswagen Tiguan 2014	600	347
6	Honda Accord 2013	173	840
7	Toyota Camry 2011	35	338
8	Volkswagen Touareg 2014	21	156
9	Nissan Altima 2014	193	271
10	Nissan Rogue 2011	105	0
11	Subaru Legacy 2015	119	30
12	Mazda CX5 2014	202	89

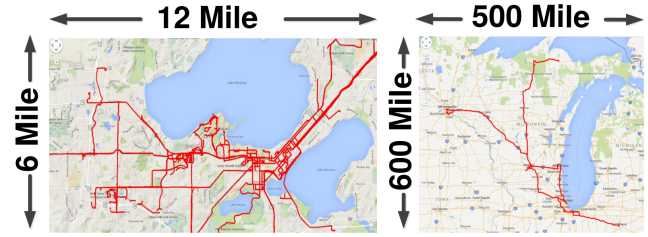

Figure 3: Urban driving data (left) is collected from a US mid-size city. Highway driving data (right) is collected from both local highways and cross-state highways.

$$AFR = C_f * MAF * (1 + LTFT) * (1 + STFT). \qquad (1)$$

where C_f is a constant value for unit conversion, e.g., from air flow rate to fuel rate or to carbon emission rate.

Vehicular Speed is measured by the perimeter of the wheel and number of rotations. Accelerator Position is measured by the angle of the gas pedal. It controls the air/fuel rate injected into the Engine. Engine RPM is measured by the rotation speed of the engine motor.

2.3 Dataset

We use a customized Android application to collect OBD and GPS data from different car makes and models. The dataset is summarized in table 2. The transmission of car 2,6,9,10 and 11 is CVT and the rest is SAT. The driving data of car 1-6 is collected over a course of three months, and the rest is collected from random events, e.g., three cross-state highway trips are collected by car 7, 8 and 9, respectively. Here we separate urban driving and highway driving, as urban driving patterns are dominated by frequent accelerations and decelerations while highway driving patterns are dominated by constant speed cruising. The dataset covers car makes from three different countries, urban data over 5,000 miles, highway data across 5 US states and over 5,000 miles, two common types of transmissions. We use the first

---

[3] Air/fuel ratio is a constant value around 14.67, while air/fuel rate is the volume of air/fuel injected per unit time.

car to install a prototype of EcoDrive and tested it in both urban and highway environments.

2.4 EcoDrive Architecture

EcoDrive reads real-time OBD parameters through OBD port and controls air/fuel injection rate by emulating the gas pedal. We illustrate the architecture of EcoDrive in Fig. 1. The sensed parameters from OBD port are passed to modeling component to train the models. The modeling component builds an AFR profile to record the instant fuel consumptions of various accelerations at different speeds. Based on pre-calculated driving strategy and real-time sensed vehicular speed, the controlling component controls air/fuel injection rate to adjust vehicular speed by sending the gas pedal position values to the ECU. To emulate the gas pedal, we utilize the drive-by-wire technology, where the gas pedal and throttle are connected by electronic messaging instead of mechanical linkage. It enables the position of gas pedal can be sent as a message to control throttle position. The throttle controls the volume of air flow injected into the engine.

2.5 Applications

EcoDrive is used as an independent system that can be installed on regular vehicles. It can be used as a control system in a way similar to cruise control [9, 18, 19]. In this application, it is used as a EcoDrive mode that a human driver can switch it on or off. Drivers are required to press the brake pedal to turn this mode off to keep safe distance to front car or traffic lights/signs accordingly. This mode can be integrated with intelligent front object or traffic light detection systems [18, 19] to further enhance driving experience. EcoDrive requires two inputs, the speed limit and road length. The speed limit can be specified by human driver or obtained from online database [20]. The road length can be calculated by a navigation software. How to obtain these information is beyond the scope of this paper.

It can also be used as a subsystem of autonomous driving systems [1, 4, 5, 21]. We envision a highly autonomous and intelligent system that all the route information are pre-calculated, e.g., speed limit, traffic conditions and distances of each road segment etc. EcoDrive can be used as a subsystem to predict fuel consumption of possible routes [22] and calculate the optimal driving strategy.

3. VEHICLE DYNAMICS AND FUEL CONSUMPTION

EcoDrive models various forces as functions of instant fuel consumption and produces fuel consumption profile as output. Different from Existing vehicle dynamics models [23], we only use parameters available from OBD instead of assuming the engine parameters are known in advance. We consider several factors that affect car fuel consumption, e.g., engine torque, drivetrain loss, wind resistance and grade resistance. The vehicle force can be modeled as follows.

$$F = F_p - F_l - F_g - F_w. \tag{2}$$

where F_p refers to the propulsion caused by car engine, F_l refers to the drivetrain loss caused by transmission and various gears connecting engine and wheels, F_w refers to the wind resistance, F_g refers to the grade resistance. Since we are only interested in fuel consumption on acceleration and cruise, we exclude the forces caused by brake.

We model the forces in the following steps. First, we use RPM and vehicular speed to model gear ratio and use AFR to model engine torque. Second, the drivetrain loss and wind resistance is modeled as the counterforce of propulsion when the car is driven in constant speeds. Finally, we use driving data extracted from flat road to train the parameters of our propulsion and loss models, and then use the propulsion model to model grade resistance.

3.1 Propulsion Modeling

The propulsion (or output torque) of vehicle is closely related to transmission gear ratios and engine torque [10, 11]. The propulsion produced by vehicle engine can be represented by

$$F_p = a_p * \tau_e * R_G. \tag{3}$$

where τ_e refers to the engine torque, R_G refers to transmission gear ratio and a_p is the coefficient used for unit conversion.

3.1.1 Gear Ratio Modeling

Modern vehicles are usually using automatic transmission and freeing the driver from shifting gears manually. There are mainly two types of transmissions, Step Automatic Transmission (SAT) and Continuously Variable Transmission (CVT). SAT uses discrete transmission gear ratios while CVT has an infinite number of effective transmission gear ratios. Since different transmission types show different properties of gear ratio changes over different speeds, we use gear ratio changes to identify different vehicle transmission types. We estimate gear ratios by using RPM and speeds when the car is driving in constant speeds.

In a SAT vehicle, R_G are discrete values.

$$R_G = R_i = \frac{RPM}{v} \qquad i = 1, ..., n \tag{4}$$

where v represents the vehicular speed and n is the number of gears of a transmission, e.g., $n = 4$ for a 4-speed transmission.

In a CVT vehicle, R_G changes continuously over time and always approach the optimal RPM.

$$R_G = \begin{cases} \frac{RPM_a}{v} & \text{if } \ v < v_T, \\ R_b & \text{if } \ v \geq v_T. \end{cases} \tag{5}$$

where RPM_a is the optimal RPM of a given engine, and v_T is the speed threshold. If the vehicular speed is lower than v_T, the engine RPM converges to a constant value RPM_a under different speeds. If the vehicular speed is higher than or equal to v_T, the gear ratio is a constant value R_b.

3.1.2 Engine Torque Modeling

Engine torque is produced by the explosion of air and fuel, therefore, we can use AFR to model engine torque.

$$\tau_e = AFR^{f(v)}. \tag{6}$$

where $f(v)$ is a parameter function that is monotonically increasing with vehicular speed. Based on our empirical

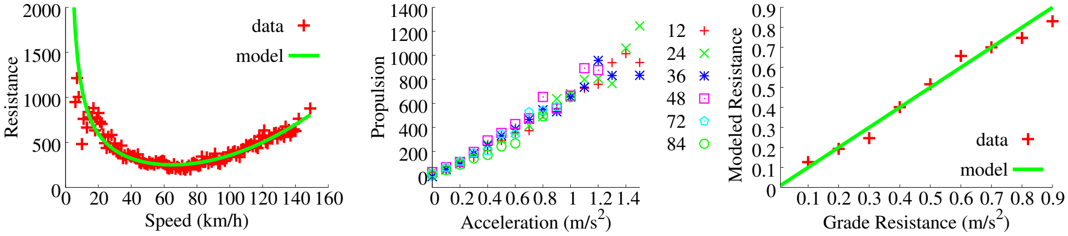

Figure 4: Vehicle Dynamics Modeling. The figures are about drivetrain loss and wind resistance under each speed (left), relation between acceleration and propulsion (minus wind resistance and drivetrain loss) under each speed (middle), modeled grade resistance with groundtruth (right).

observation, it starts from 0.3 at speed 0 and converges to 1 when the speed is larger than $20km/h$.

Therefore, the propulsion of a car on the wheel can be written as

$$F_p = a_p * \tau_e * R_G = a_p \frac{AFR^{f(v)} * RPM}{v}. \tag{7}$$

3.2 Drivetrain Loss and Wind Resistance Modeling

Instead of modeling each loss individually, we model all the power losses as a whole. There are mainly two power losses in automotive systems, drivetrain loss and wind resistance. Drivetrain loss consists of transmission loss and other mechanical system losses. Transmission connects car engine and wheels by various gears and power loss occurs when transmit the power from engine to wheels. We model drivetrain loss in the following form.

$$F_l = \frac{a_l}{v} + b_l * v + c_l. \tag{8}$$

where v is the vehicular speed and the rest are coefficients. The drivetrain loss drops to minimum when the car is driving in moderate speed.

The force from the air drag [12] can be represented by following equation,

$$F_w = 0.5 * \rho_a * c_d * A_a * v^2 = a_w * v^2. \tag{9}$$

where ρ_a is the air mass density, c_d is the air drag coefficient and A_a is the effective area of the vehicle. Since the effective area of vehicle is different from vehicle to vehicle, we model air drag as a function of vehicle speed with unknown coefficient a_w.

After we put drivetrain loss and wind resistance together, the modeled loss and actual loss are shown in the Fig. 4. In low speed, the main loss comes from drivetrain loss due to mechanical frictions. In high speed, the main loss comes from wind resistance due to air compression. After we have drivetrain loss and wind resistance, we use propulsion minus the loss $(F_p - F_l - F_w)$ to model the relation between acceleration and propulsion, as shown in the middle of Fig. 4. For accelerations, we use extra MAF instead of raw MAF readings from the OBD port. Raw MAF minus the MAF required to maintain a certain speed is the extra MAF for this speed.

3.3 Grade Resistance Modeling

Different road types, i.e., flat road, uphill and downhill, have significant different impacts on vehicle movements and fuel consumptions. Similar to [12], the grade resistance can be modeled as a combination of forces caused by grade and rolling resistance.

$$F_r = mgc_r \cos\theta. \tag{10}$$

where c_r is the grade resistance coefficient, m is the vehicle mass, g is the gravity of earth , θ is the road grade and v is vehicular speed. The road elevation information is obtained from National Elevation Dataset [13].

3.4 AFR Profile

Vehicle needs different air/fuel injection rate to achieve different accelerations under different speeds. Based on the vehicle dynamics models we built in previous sections, we can build a fuel consumption profile. In this profile, we can lookup the AFR required to accelerate a vehicle with an arbitrary acceleration under any speed. Let $AFR(v, a)$ denotes the air/fuel rate is required to accelerate the vehicle at acceleration a when the vehicle is driving at speed v. The profile can be represented as the following equation.

$$AFR(v, a) = e^{\frac{log(a/R_G)}{f(v)}} + AFR_c(v) \tag{11}$$

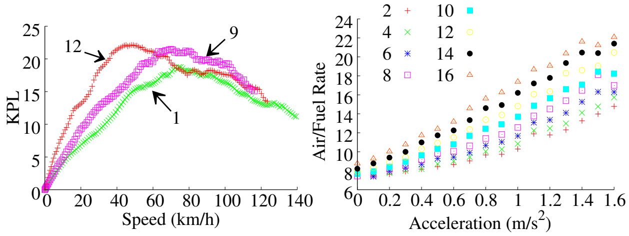

Figure 5: AFR profile.

An illustration of AFR profile of car 1 is shown in the right subfigure of Fig. 5. The profile provides the AFR required to accelerate the car at a certain acceleration under a certain speed. We can infer a lookup table for each $AFR(v, a)$ based on the profile. The left subfigure in Fig. 5 illustrates different vehicles have different speed-KPL matching, which indicates AFR profile needs to be built based on individual vehicles.

4. ACCELERATION CONTROL

EcoDrive controls air/fuel injection rate to control vehicular speed by emulating gas pedal. EocDrive calculates accelerations of each speed and leaves brake decision to drivers or other systems. It needs two inputs, road speed limit and segment length. The speed limit can be set by human drivers or queried from online database [20]. The road segment length can be estimated by human drivers or by a third-party navigation software. Road segment length is not necessary for long distance roads, as the driving strategy will be the same if the distance is longer than a threshold. For example, if the speed limit is $25mph$, the fuel efficient acceleration strategies for $300m$ and $500m$ are the same. In this paper, we assume both speed limit and road segment length are known. The output of EcoDrive is a matrix and each element in the matrix is the minimum fuel consumption when the car arrives at a certain distance with a certain speed. By traversing all the possibilities, EcoDrive is able to find all reasonable driving strategies with different fuel consumption and travel time.

4.1 Problem Statement

Driving pattern can be abstracted as accelerate-cruise-brake. EcoDrive focuses on the accelerate-cruise part and leaves brake decision to human drivers. For example, in a $300m$ road segment with speed limit $25mph$ (or $40km/h$), the EcoDrive accelerates and cruises to $200m$ and the driver uses the rest $100m$ to stop the car as there is a stop sign at the end of the road. Similar to cruise control, we assume there is enough distance to front obstacles for drivers to brake.

EcoDrive needs two key inputs: 1) The length of the road segment, i.e., from current location to the point driver will make a brake, 2) The lower and upper speed limits of the road segment. Such inputs can be given by user or extracted from online databases and navigation softwares. We focus on how to utilize such inputs to calculate fuel efficient strategies. Therefore, the problem that EcoDrive is going to address can be stated as follows: *Given a road segment length with lower and upper speed limits, how to control air/fuel injection rate in real time to achieve the most fuel efficient driving strategy.*

4.2 Driving Strategy by Dynamic Programming

We use dynamic programming to traverse all the possibilities to find all reasonable driving strategies. This is inspired by the fact that the fuel efficiency and travel time are related to the speed and distance. We divided the road segment into smaller equal length road segments. We use v_i and d_k to represent the ith speed and the length of kth segment, respectively. We use $s(i,k)$ to represent the state when the car is driving at speed v_i at the start location of road segment d_k. For any state $s(i,k)$, there are two sets of possible transitions. First, it transits from $s(i,k-1)$, which means that the car is driving in constant speed v_i in last road segment. Second, it transits from $s(i-1,k-1-m)$, which means the car is accelerating from speed v_{i-1} to v_i under acceleration a_j. The lower speed bound is used to eliminate some unreasonable cruising speed, e.g. it is obviously not fuel efficient and reasonable if the car is cruising at 1 km/h.

In the first case, the accumulated fuel consumption is the sum of the fuel consumption used to reach state $s(i,k-1)$ and the cruising fuel consumption from d_{k-1} to d_k.

$$FC_c(i,k) = FC(i,k-1) + AFR(v_i, 0.0) * t_i. \tag{12}$$

where t_i means the time cost when the car is driving through road segment d_{k-1} and $t_i = \frac{d_{k-1}}{v_i - v_{i-1}}$.

In the second case, the accumulated fuel consumption is the sum of the fuel consumption used to reach state $s(i-1,k-1-m)$, the cruising fuel consumption FC_c in part of road segment $k-1-m$, and the acceleration fuel consumption FC_j. The car may start to accelerate at any location in road segment $k-1-m$. The road segment is split into two parts, the car cruises to a point within the road segment $k-1-m$ and then starts to accelerate at acceleration a_j. Therefore, the fuel consumption of this case can be represented as follows.

$$FC_a(i,k) = FC(i,k-1-m) + FC_c + FC_j. \tag{13}$$

where $FC_c = AFR(v_i, 0.0) * t_c$ and t_c is the cruising time cost within road segment $k-1-m$. To calculate t_c, we need to calculate the distance used to accelerate from v_{i-1} to v_i. Before that, we present the acceleration fuel consumption cost as

$$FC_j = AFR(v_{i-1}, a_j) * t_a. \tag{14}$$

The acceleration time is $t_a = \frac{v_i - v_{i-1}}{a_j}$ and the acceleration distance is $d_a = v_{i-1} * t_a + 0.5 * a_j * t_a^2$. The cruising distance is simply $d_c = \sum_{x=k-m-1}^{k-1} d_x - d_a$ and the cruising time is $t_c = \frac{d_c}{v_{i-1}}$.

Therefore, the fuel consumption at the start of state $s(i,k)$ is

$$FC(i,k) = min\{FC_c(i,k), min\{FC_a(i,k)\}\}. \tag{15}$$

After we traverse all the states, the last state of each speed is the minimum fuel consumption when the car is finally cruising in that speed. The most economy driving strategy can be traced back from the state with minimum fuel consumption. Different travel time is used when the car is cruising in different speeds. By using this property, we can find a tradeoff between travel time and fuel consumption.

Assume that we split the road segment into n smaller equal length road segments , there are m different speeds and l different accelerations, the time complexity of the dynamic programming algorithm is $O(n*m*l)$ and the space complexity is $O(n*m)$. In practice, the most road segment length in urban area is less than $1km$ and we set each smaller road segment to be $1m$. The unit of the speed sensed from the OBD port is km/h and the speeds are integers, so we have $0-60$ different speeds in urban area. The acceleration is normally from $0.1m/s^2$ to $1.4m/s^2$. Therefore, the time complexity of the algorithm grows linearly with the total road length.

Different road conditions affect air/fuel rate as the car needs lower (or higher) air/fuel injection rate to achieve the same acceleration in downhill (or uphill) conditions. The air/fuel rate for a specific road segment d_k becomes $AFR(v_i, a_j - a_k)$, where a_k refers to the acceleration/deceleration caused by the kth road segment. The driving strategy can be calculated based on the dynamic programming solution as well.

4.3 Fuel Economy and Travel Time

In general, fuel efficient driving strategies require longer travel time in urban environments. Therefore, there is a trade-off between fuel efficiency and travel time. EcoDrive provides different travel time by selecting different target speed. The higher the target speed, the less the travel time.

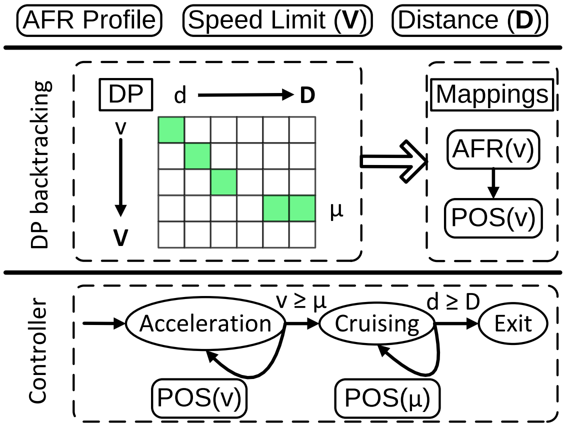

Figure 6: EcoDrive control flow.

The user can select the most fuel efficient driving strategy with fixed target speed, or select desired travel time with corresponding target speed. The selection of target speed also depends on the speed limit, i.e., the target speed should not be higher than the speed limit. The most fuel efficient cruising speed ranges from 30mph to 60mph for different types of vehicles. In urban environments, the most fuel efficient target speed is around the speed limit. However, in short road segments, it is generally more fuel efficient to select a target speed that is lower than speed limit due to extra cost on acceleration.

4.4 Highway Strategy

The most fuel efficient highway speeds range from 40km/h to 80km/h for the vehicles we observed. Given the highway speed limit ranging from 80km/h to 110km/h in US, drivers can find a trade-off between fuel efficiency and travel time by selecting different cruising speeds. Generally speaking, constant speed cruising generally has higher KPL than frequent accelerations and decelerations. Therefore, traditional cruise control is considered a good option for economic driving. However, traditional cruise control sticks to one target speed. If the driver manually increases target speed, the car will aggressively approach to the target speed, which will bring extra fuel consumptions. Similarly, when the car is cruising uphill, the road will slow down the car and cruise control will aggressively approach the target speed. In downhill condition, cruise control will reduce air/fuel injection to maintain the target speed. EcoDrive increases speed gradually and utilizes the acceleration caused by downhill to achieve higher KPL.

4.5 Acceleration Controller

We summarize the control flow of EcoDrive in Fig. 6. The AFR profile is calculated offline by the modeling component. Given the travel distance D and speed limit V, EcoDrive uses dynamic programming model to calculate the speed to acceleration mapping of the most economic driving strategy. This process is done by backtracking the last state of target speed μ. The speed to acceleration mapping records the desired acceleration under that speed. This mapping is converted into speed to air/fuel injection rate mapping $AFR(v)$ by querying the AFR profile. The acceleration controller retrieves the air/fuel injection rate based on the mapping and real-time sensed vehicular speed. A air/fuel rate to gas pedal position mapping and a gas pedal position to voltage mapping are calculated in advance. The controller gets the speed to gas pedal position mapping $POS(v)$. Based on the air/fuel injection rate, EcoDrive sends corresponding voltage values to the ECU. In this process, EcoDrive maintains three states, acceleration, cruising and exit. EcoDrive is in acceleration state by default and enters cruising state when sensed OBD speed is not less than target speed. After it enters cruising state, it remains constant air/fuel rate injection. Different from traditional cruising control, EcoDrive may increase/decrease speed in cruising state by adapting to various road conditions. If the car reaches the distance or EcoDrive is turned off by switch of brake, it enters exit state. EcoDrive releases all the resources in exit state and aborts to wait for next inputs.

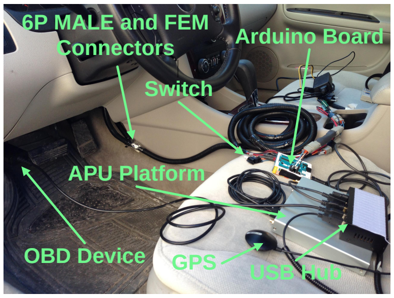

Figure 7: EcoDrive in-vehicle setup.

5. IN-VEHICLE SETUP AND IMPLEMENTATION

We explain the in-vehicle setup and implementation of EcoDrive.

5.1 In-vehicle Setup

The in-vehicle setup of EcoDrive is shown in Fig. 7. EcoDrive uses Arduino Uno microcontroller to emulate gas pedal by delivering signals to the ECU through a 6P FEM Connector. The microcontroller outputs a 16-bit resolution digital pulse-width modulation (PWM) signal which is smoothed out by an RC filter. A 6P MALE Connector is used to read signal outputs from the original gas pedal. A wiring harness and switch button are built that allows the user to switch the signal read by the ECU between the gas pedal and the microcontroller. Therefore, the setup supports two mode, human driving mode and EcoDrive mode. In human driving mode, the gas pedal and air/fuel injection rate is controlled by human driver. In EcoDrive mode, the gas pedal is unusable and air/fuel injection rate is controlled by the system. In EcoDrive mode, The APU platform [24] sends pedal positions to the microcontroller through serial communication and the microcontroller converts the pedal position values into analog voltages outputs. To do this, we construct a mapping between sensor analog voltage outputs and pedal positions. By using the mapping between gas pedal position and air/fuel rate built from driving traces, we can send position values from laptop to the car to control air/fuel injection rate.

5.2 Implementation

We implement EcoDrive data sensing module and air/fuel injection control module in C++. We operate EcoDrive on the Ubuntu 14.04 32bit distribution (with linux kernel version 3.13.0-34-generic), that runs on PC Engines APU platform [24]. APU platform is a mobile embedded platform that is equipped with 1GHz dual core CPU and 4G DDR3 DRAM. EcoDrive uses two threads to query and read OBD messages, respectively. One thread sends OBD query message to the OBD port in every 250ms. Higher frequency OBD query will cause CAN read error. Another thread listens on the OBD port for echo messages and sends voltage control commands to Arduino board. The Arduino board initiates a loop to listen on the USB serial and sets the voltage output of two pins based on received command. The two pins represent the voltage output pins of gas pedal's two position sensors.

6. EVALUATION

Our evaluation consists of four parts, EcoDrive road testing results in both urban and highway environments, vehicle dynamics modeling accuracy, driving data statistics and trace-driven simulation.

6.1 Fuel Efficiency Test Results

6.1.1 Urban

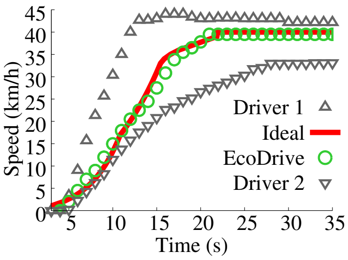

Figure 8: Driving behaviors of EcoDrive and Human Drivers on a 300m length road segment.

In this experiment, we compare EcoDrive with human drivers in urban road segments. EcoDrive uses drive-by-wire technology to control air/fuel injection rate. It accelerates the vehicle by sending gas pedal position values to ECU. EcoDrive provides driver a switch button which can disable EcoDrive immediately. In our evaluation, we did not experience any bugs or out of control situations, but we made our risk management as follows. First, we control the input parameters. i.e., the gas pedal position value, on both the controller and the Arduino board. Second, we can shift the transmission to neutral or park to disconnect the engine and wheel if the engine is out of control due to unexpected gas pedal position input. Third, the brake dominates the gas pedal and we can brake even when the engine is out of control.

Test Drive Example. A EcoDrive prototype test drive and human drive tests are conducted in a $300m$ length road segment with speed limit of 25mph (or 40km/h). The vehicular speed traces are shown in Fig. 8. The ideal curve is plotted from the traces obtained from dynamic programming model. In our prototype, the air/fuel injection rate is controlled based on sensed speed. The speed may change at any point during the query interval, so that it is challenge to control the air/fuel injection rate precisely at each speed. The differences between estimated speed traces and actual speed traces are mainly caused by the impreciseness of speed sensing and road conditions. Some other factors affect the preciseness of vehicular speed control include wind speed and passenger weights etc. Two human drivers are asked to drive on the same road segment. It is shown that the driver 1 is more aggressive by fast accelerations and driver 2 is more conservative by slow accelerations. The driving trace of EcoDrive prototype falls in between. In this comparison, the prototype shows 23% and 19% fuel efficiency improvements than driver 1 and 2, respectively. Driver can reach a fuel efficient speed faster by fast acceleration, but it takes more fuel during the acceleration process. If a driver uses lower accelerations to reduce fuel consumption during acceleration process, overall fuel efficiency will not be increased due to driving in low (less fuel efficient) speeds and longer travel time. EcoDrive chooses the best strategy for economic driving.

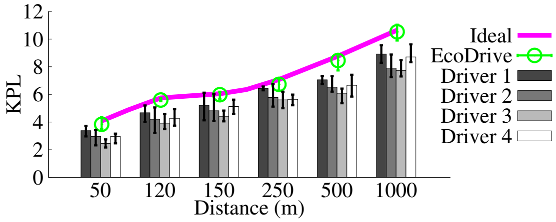

Figure 9: Fuel efficiency comparison between EcoDrive and human drivers on urban road segments.

Urban Road Segments. We select six different road segments and let EcoDrive drive the car 10 times on each road segment. We recruited four drivers to drive on the same road segments 10 times as well. The speed limit of the $500m$ and $1000m$ road segments is $30mph$ (or $50km/h$) and the that of the $150m$ and $250m$ road segments is $25mph$ (or $40km/h$). There is no speed limit for the $50m$ and $120m$ road segments. The length is the driving length of EcoDrive and the overall road segment length is longer so that drivers have enough time to stop the car. The results are shown in Fig. 9. The ideal fuel consumption is the theoretical fuel consumption that is calculated based on AFR profile and dynamic programming model. The actual fuel consumption of EcoDrive on certain road segment length is very stable. EcoDrive shows 10%-40% fuel efficiency improvement compared to different drivers with different road segment lengths. The fuel efficiency improvement comes from the smooth driving behaviors of EcoDrive. EcoDrive has an average of 20% more travel time than human drivers. EcoDrive can have 5%-10% improvements than human drivers if not sacrificing travel time. The plots of tradeoff between fuel consumption and travel time are omitted due to the space limit.

6.1.2 Highway

In this experiment, we compare EcoDrive with cruise control and human drivers on highway.

Compare to Cruise Control. Fig. 10 plots the different acceleration patterns between cruise control and EcoDrive. Cruise control is a vehicle built-in feature that can keep car driving at a certain speed while human drivers can manually increase or decrease the cruising speed. The cruise control pattern is extracted when the gas pedal position is 0 but the air/fuel injection rate is higher than the idle state

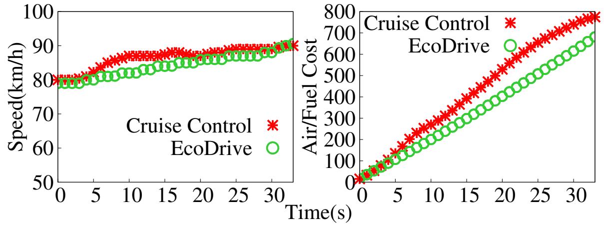

Figure 10: Acceleration comparison between Cruise Control and EcoDrive.

rate. As shown in the figure, cruise control consumes more fuels than EcoDrive due to the aggressive acceleration strategy. The travel distance is similar for both cases, so EcoDrive is more fuel efficient than cruise control. Similarly, driving uphill by using cruise control will consume more fuels as well. When the car is driving uphill, the vehicular speed drops due to grade resistance, so cruise control will aggressively accelerate to the setting speed. In downhill conditions, cruise control will reduce air/fuel injection rate, while accelerating by maintaining the same air/fuel injection rate has higher fuel efficiency.

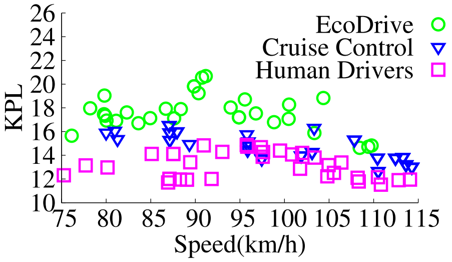

Figure 11: Highway experiments.

Compare to Human Drivers. We evaluate EcoDrive by driving on two highway segments, one is a local highway segment and another is a cross state highway segment. The highway segments are selected based on historical driving traces of car 1. EcoDrive is evaluated by two-way driving in each highway segment, e.g., if EcoDrive enters highway at A and exits at B in one experiment and it will enter at B and exits at A in next experiment. The two-way road testing is used to evaluate EcoDrive in both uphill and downhill conditions. EcoDrive is evaluated between exit 256 and 262 on Madison's West Betline Highway, and between exit 132 and 136 on US 14 Highway.

The cruise control and human driving traces are extracted from historical driving traces collected on the same highway segments. In each driving trace, the car is traveling at a constant speed(it may have some small speed variations). And we calculate the average speed of each trace as the vehicle speed. The results are shown in Fig. 11, where each point represents the KPL of a $2km - 3km$ length road segment. To eliminate the impact of traffic, the traces that have speed decrease due to brake are not included. In this experiment, EcoDrive shows more than 10% fuel efficiency than cruise control and more than 30% fuel efficiency than human drivers on average.

6.1.3 Travel Time and Fuel Efficiency

In this experiment, we evaluate the tradeoff between travel time and fuel consumption.

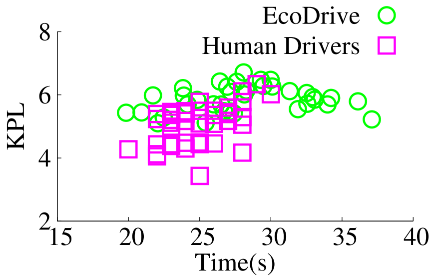

Figure 12: Tradeoff between travel time and fuel consumption on a 250m road segment.

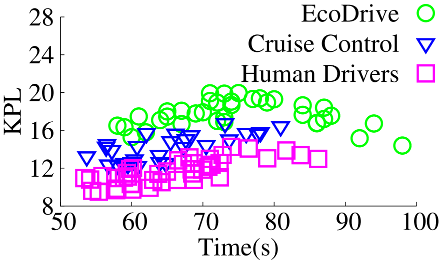

Figure 13: Tradeoff between travel time and fuel consumption on highway.

Urban. The tradefoff between travel time and fuel consumption of EcoDrive on $250m$ road segment is shown in Fig. 12. For EcoDrive, the optimal KPL is around 7 and the travel time is around 28 seconds. On the other hand, the average KPL for human drivers is around 5.5 and the average travel time is around 25 seconds. EcoDrive can achieve a 20% fuel efficiency improvement on average by sacrificing travel time by around 10%. EcoDrive can also improve fuel efficiency under same travel time in most cases.

Highway. Fig. 13 shows the tradeoff between travel time and fuel efficiency on highway. Each highway segment length is around $2km - 3km$. EcoDrive can improve fuel efficiency by more than 30% on average without sacrificing travel time. The gain comes from smooth acceleration and road adaptation. More fuel efficiency can be achieved by cruising at the peak KPL speed, which is around $90km/h$ in this evaluation.

6.2 Modeling Accuracy

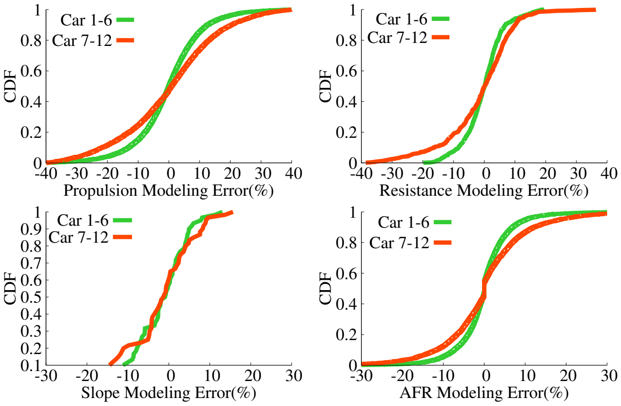

Figure 14: The cumulative distribution function of vehicle dynamics modeling errors.

In this experiment, we evaluate the modeling accuracy of vehicle dynamics models. Different vehicle forces, propulsion, resistance and grade resistance, are evaluated separately. The AFR profile calculation accuracy is evaluated as well. We divide the data into two sets, one set is the training set used to build the model and another set is the testing set for evaluating the model. For each model, we repeat this process and plot the modeling accuracy of various vehicle dynamics models in Fig. 14. From top-left to bottom-right, they are propulsion model evaluation, drivetrain loss and wind resistance model evaluation, grade resistance model evaluation, and AFR profile evaluation.

We use **green** curves to represent the cars have more than one thousand miles driving traces (Car 1-6) and use **red** curves to represent the rest (Car 7-12). The **green** curves show that the fitting errors of more than 95% cases are within ±10%. The **red** curves show less fitting accuracy due to less miles.

6.3 Driving Data Statistics

6.3.1 Urban Road Segment Length

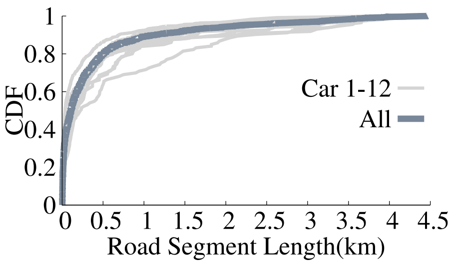

Figure 15: Road segment length in urban area.

We summarize the length of road segments in Fig. 15. A road segment is defined as the distance between two stand/stop locations. A stand/stop location is defined as following: 1) The speed of the car is zero; 2) The car is making a turn. The speed of the car can be easily sensed from the OBD port. We identify turns from GPS data. The driving direction of car can be calculated from two adjacent GPS points. If the accumulated driving direction change are close to 90 degree, then we mark this is a stand/stop location. This rough road segment length statistics give us a guideline for experiments. As shown in the figure, the most of road segments are within 1000m. This indicates that the driving pattern in urban area is primarily accelerate-brake.

6.3.2 Urban Acceleration

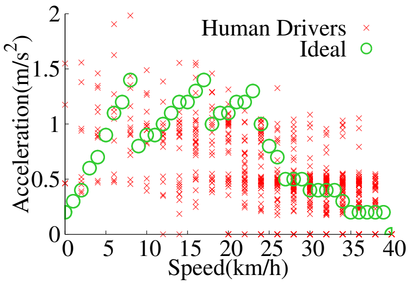

Figure 16: The urban acceleration patterns of different drivers.

Fig. 16 explains the acceleration patterns of car 1 and the ideal acceleration pattern calculated by EcoDrive with equal road length. As shown in the figure, drivers are not aware of the most efficient accelerations of different speeds and not able to drive at a certain speed. They drive either lower than desired accelerations, which will increase fuel consumption due to longer travel time in low KPL speeds, or higher than desired accelerations, which will increase fuel consumption due to extra fuel consumption to achieve the target speed.

6.3.3 Highway Driving Speed

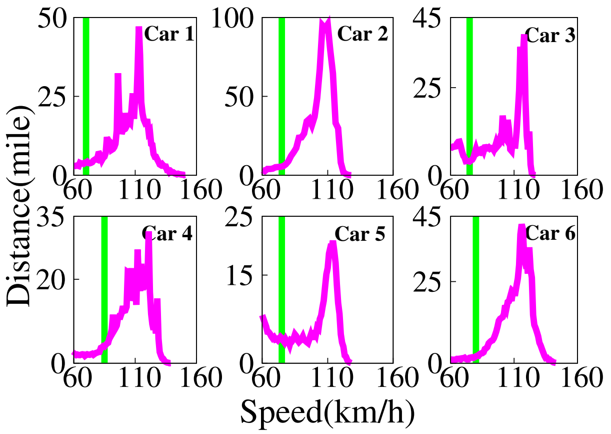

Figure 17: The highway driving speed distributions of car 1-6.

The speed limits of rural highway is 65mph (104.6km/h) to 70mph (112.6km/h), and the speed limit of urban highway is 55mph (88.5km/h). We summarize the highway driving speeds of car 1-6 and illustrate the distribution and best KPL speed in Fig. 17. We observe that most drivers drive above the speed limits with noticeable mileages and much higher than the best KPL speed. This is partially because the drivers are not aware of the much higher fuel consumption in high speeds. Also, slowing down when following a slow car will increase fuel consumption due to power loss when braking. Frequent accelerations and decelerations increase fuel consumption as well.

6.4 Trace-driven Simulation

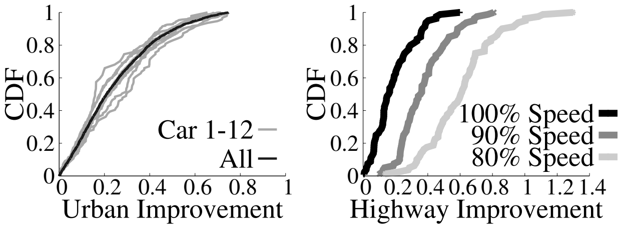

Figure 18: Fuel efficiency improvements by trace-driven simulation.

In this experiment, we evaluate EcoDrive by trace-driven simulations.

Urban. We divide urban trips into variable-length road segments where the vehicular speed is 0 at the start and end of each segment. Each segment is divided into four parts, accelerating part, cruising part, braking part and idle part. Accelerating part ended when the speed does not increase in

5 seconds. Braking part started when the driver release gas pedal at the end of each segment. Cruising part falls between accelerating part and braking part. Idle part follows braking part after the vehicular speed reach 0. The speed limit is calculated based on the average speed of cruising part. We calculate the fuel cost of EcoDrive by replaying acceleration and cruising parts. We use the same braking and idle cost extracted from original traces. We sum up the two costs to calculate the final fuel consumption of EcoDrive. As shown in Fig. 18, EcoDrive can improve a median of 20% fuel efficiency in urban environment. The travel time is reduced by 10%-30% in 20% cases and is increased by less than 25% in 60% cases. The plots of travel time are omitted due to the space limit.

Highway. We divide highway trips into road segments based on deceleration. If the deceleration is larger than a threshold, we separate current segment into one highway segment. The average speed of highway segment is used as target speed for EcoDrive. We replay each highway segment by EcoDrive with three different target speeds, the same target speed, 90% of target speed and 80% of target speed. As shown in Fig. 18, EcoDrive can improve fuel efficiency by more than 20% on average without sacrificing travel time and can improve an average of 40% fuel efficiency by sacrificing 10% travel time on highway.

7. RELATED WORK

7.1 Cruise Control

Cruise control [9] is a system that automatically controls the vehicle to drive in a certain speed. [18] introduces Adaptive Cruise Control (ACC) that adjusts vehicle speed based on distance to cars ahead by using radar. [25] studies the effects of time-gap settings of adaptive cruise control on driving performance in a bus driving simulator. [26] studies a distributed car control system that every car is controlled by adaptive cruise control. [19] develops an intelligent cruise control system for automatic vehicles and verify the effectiveness of the system under emergency situations by simulation. EcoDrive is more fuel efficient than adaptive cruise control in two ways. First, EcoDrive is able to improve fuel efficiency by adjusting speed according to road conditions and vehicle types. Second, EcoDrive tracks the fuel efficiency under different cruising speeds and provides a tradeoff between fuel efficiency and travel time.

7.2 Fuel Efficiency Enhancement

Increasing vehicular fuel efficiency has been a topic of much recent research [22, 17, 27, 28]. GreenGPS [22] requires a fuel map server to record the statistical route fuel usage and calculate the most fuel efficient route for drivers. SignalGuru [17] requires vehicles sharing traffic light information so that drivers can adjust the speed of the vehicle to avoid stops in front of traffic lights. [27] uses an on-board eco-driving device that provides instantaneous fuel economy feedback affects driving behaviors, and consequently fuel economy. [28] evaluates the impacts of a smartphone application on driving behaviors to enhance fuel efficiency. Different from existing approaches that are focusing on advising human drivers to adjust driving behaviors, EcoDrive is designed to assist human drivers. Both [29] and [30] show that some reduction of fuel consumption can be achieved by vehicle-to-vehicle communication.

7.3 Vehicle Dynamics Sensing

Smartphones equipped with various sensors are widely used to sense vehicle dynamics to understand driving behaviors and road conditions. [31] detects aggressive driving behaviors based on accelerometer sensor readings from smartphone. [32] determines driver phone use by comparing the centrifugal force of smartphone with that of a fixed reference sensor during turns. [33] monitors road surface conditions by using accelerometers. [34] detects various road and traffic conditions, e.g., potholes, bumps, braking, and honking, by using various sensors in smartphones. [35] senses speeds from the OBD port to identify aggressive driving behaviors and offer discounts for drivers with good driving habits. However, none of these work models vehicle dynamics as functions of instant fuel consumption.

7.4 Vehicle Force Modeling

[12] models rolling resistance and air drag based on some measurements on the vehicle, e.g., effective areas to model air drag. [36] models vehicle transmission system based on gear sizes. [37] models tire friction based on tire parameters, e.g., radius. [38] presents an electronic system attached on wheel to measure and monitor of the torque transmitted to the wheel of a moving vehicle. [10] models engine torque and horsepower by RPM when the engine is not connected with transmission. Our model uses the parameters available on car CAN bus to find the relations between fuel consumption and vehicle speed changes. Different from existing work, our model does not rely on the detail physics properties of the car, e.g., tire radius and transmission gear size etc.

7.5 Autonomous Driving

There have been major advances in designing and building autonomous vehicle operating systems to realize safer and more convenient vehicles [1, 4, 5, 21]. These work focuses on issues related to object detection and efficient navigation and so forth. CarSpeak [3] is designed as a communication system to share sensory data between autonomous vehicles for obstacle detection. Different from autonomous cars studies, we focus on the design and implementation of fuel-aware acceleration control system that works on existing regular vehicles. We expect that EcoDrive can be integrated with autonomous driving cars, but EcoDrive can also be used independently. Also, EcoDrive can work on existing vehilces with little hardware changes.

8. DISCUSSION

In this section, we discuss some design considerations of EcoDrive.

Hybrid Vehicle (HV) and Electric Vehicle (EV). EcoDrive is designed for gasoline-powered vehicles, but the modeling process and gas pedal emulation can also be applied to HVs and EVs. Currently, the majority of vehicles running on street are still gasoline-powered vehicles. It is important to develop systems like EcoDrive to work on regular vehicles to improve fuel efficiency and limit carbon pollution. The core of EcoDrive is that it can work on regular vehicles with easy and recoverable installation.

Instant Fuel Economy Display. In urban environments, acceleration is not fuel efficient, but accelerating to a higher speed in a careful way can improve fuel efficiency. If the driver refuses to accelerate due to low instant fuel efficiency, the car will drive in low fuel efficient speed and eventually increase fuel consumption. Similarly, releasing gas pedal can increase instant fuel efficiency dramatically, but frequent acceleration and deceleration will consume more fuel than cruising under certain speed. Therefore, instant fuel economy display is misleading and may reduce overall fuel efficiency.

Impact of Traffic and User Experience. EcoDrive controls the vehicle in a way that is similar to cruise control, so the driver should keep a safe distance to front vehicle. Since EcoDrive accelerates the vehicle in various ways according to different road segment lengths, it is more challenging for human drivers to keep safe distance than cruise control in dense traffic scenarios, because drivers need to frequently brake to avoid front-end collision. Therefore, EcoDrive is more suitable for drivers to use in low traffic volume scenarios. However, it is shown from the data we collected that drivers tend to accelerate much more aggressive than the optimal acceleration patterns. In other words, EcoDrive achieves higher fuel efficiency by accelerating slower than most drivers usually do. Also, EcoDrive uses fixed acceleration pattern for fixed road length and speed limit, drivers and passengers can easily get used to the driving style of EcoDrive. Therefore, we expect EcoDrive can be useful in most urban driving scenarios (except traffic hours) and drivers can still feel that the vehicles are under control due to predictable accelerations. We also expect that EcoDrive can tolerate more complex traffic conditions when integrating with front object detection and route planning systems equipped on driverless cars.

Limitations. First, EcoDrive requires the road length as input in short length road segment (e.g., shorter than 200m). Road length can be obtained by navigation softwares, but how to retrieve such information is beyond the scope of this paper. Second, it relies on drivers to brake or switch to disable EcoDrive mode. The operation complexity is acceptable for drivers as proven by cruise control. Third, EcoDrive requires mileage trainings to build an accurate model, i.e., 1000 miles driving data can build a very accurate model as shown in evaluation. Some optimization can be made to reduce training time, e.g., sharing the AFR profile among same vehicle models. Fourth, we did not evaluate EcoDrive by end-to-end scenarios in urban environment, e.g., what are the fuel savings from home to work. There are different traffic volumes and different traffic light schedules among different trips. Therefore, it is challenging to compare EcoDrive with human drivers in such scenarios. Given the fuel improvement in arbitrary length road segments, we believe EcoDrive can improve fuel efficiency in end-to-end scenarios as well.

Each trip can be divided into three parts: acceleration, cruising and braking. Acceleration and cruising consume most of the fuel during a trip. For this work, EcoDrive focuses on the acceleration part and cruising part. There is no direct way to fairly compare two brakes that made by different drivers, or different brakes made by the same driver. Therefore, we did not include the fuel consumption of braking in our evaluation.

There are other factors that affect fuel consumption, e.g., tire pressure, temperature and wind direction etc. A more accurate model can be built by using more vehicle parameters. However, there is no direct way to access some in-vehicle parameters like tire pressure. We advocate that the vehicle manufacturers provide more information through OBD port that can be used to enable more applications. Some other real world parameters, e.g., traffic light schedule, stop sign and wind direction etc., can also be used to improve fuel efficiency. We believe such extra information can be used to improve EcoDrive in broader aspects, the detailed integration methods are out of scope of this paper though.

9. CONCLUSION

This paper introduces EcoDrive, a fuel consumption sensing and control system that assists human driver to drive fuel efficiently. EcoDrive calculates driving strategy based on the properties of individual vehicles and road conditions. To this end, it models vehicle dynamics as functions of instant fuel consumption and calculates acceleration strategies accordingly. We build a prototype of EcoDrive on a mobile embedded platform. The prototype is installed on a regular vehicle and tested for more than 100 miles in both urban and highway environments. In comparison with human drivers, EcoDrive improves fuel efficiency by 10%-40% in urban environments. It has an average of 10% higher fuel efficiency than vehicle built-in cruise control and more than 30% fuel efficiency than human drivers on highway.

10. ACKNOWLEDGMENTS

We would like to acknowledge the anonymous reviewers and our shepherd, whose comments helped bring the paper into its final form. This research project is supported in part by the US National Science Foundation through awards CNS-1405667, CNS-1345293, CHE-1230751, CNS-1343363 and CNS-1525586.

11. REFERENCES

[1] Google driverless car. *http://en.wikipedia.org/wiki/Google_driverless_car.*

[2] Qnx operating system for diverless car by blackberry. *http://www.qnx.com/.*

[3] Swarun Kumar, Lixin Shi, Nabeel Ahmed, Stephanie Gil, Dina Katabi, and Daniela Rus. Carspeak: a content-centric network for autonomous driving. In *Proceedings of the ACM SIGCOMM 2012.*

[4] Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher Baker, Robert Bittner, MN Clark, John Dolan, Dave Duggins, Tugrul Galatali, Chris Geyer, et al. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 2008.

[5] Todd Litman. Autonomous vehicle implementation predictions. 2013.

[6] Morgan Stanley Research. Autonomous cars: Self-driving the new auto industry paradigm. 2013.

[7] Obd. *http://en.wikipedia.org/wiki/On-board_diagnostics.*

[8] Obd pids. *http://en.wikipedia.org/wiki/OBD-II_PIDs.*

[9] Cruise control, wikipedia. *http://en.wikipedia.org/wiki/Cruise_control.*

[10] Chi-Man Vong, Pak-Kin Wong, and Yi-Ping Li. Prediction of automotive engine power and torque using least squares support vector machines and bayesian inference. *Engineering Applications of Artificial Intelligence*, 19, 2006.

[11] Robert A Giannelli, EK Nam, Kent Helmer, Theodore Younglove, George Scora, and M Barth. Heavy-duty diesel vehicle fuel consumption modeling based on road load and power train parameters. Technical report, SAE Technical Paper, 2005.

[12] Robin Andersson. Online estimation of rolling resistance and air drag for heavy duty vehicles. 2012.

[13] National elevation databases. *http://nationalmap.gov/elevation.html.*

[14] Fuel gauge. *http://auto.howstuffworks.com/fuel-gauge1.htm.*

[15] Adriano Alessandrini, Francesco Filippi, and Fernando Ortenzi. Consumption calculation of vehicles using obd data. 2012.

[16] Min Goo Lee, Yong Kuk Park, Kyung Kwon Jung, and Jun Jae Yoo. Estimation of fuel consumption using in-vehicle parameters. *International Journal of U-& E-Service, Science & Technology*, 4(4), 2011.

[17] Emmanouil Koukoumidis, Li-Shiuan Peh, and Margaret Rose Martonosi. Signalguru: leveraging mobile phones for collaborative traffic signal schedule advisory. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 127–140. ACM, 2011.

[18] Johan Bengtsson. *Adaptive cruise control and driver modeling.* Department of Automatic Control, Lund Institute of Technology, 2001.

[19] Petros A Ioannou and Cheng-Chih Chien. Autonomous intelligent cruise control. *IEEE Transactions on Vehicular Technology*, 1993.

[20] Wikispeedia, a road speed limit database. *http://en.wikipedia.org/wiki/Wikispeedia.*

[21] Junsung Kim, Ragunathan Raj Rajkumar, and Markus Jochim. Towards dependable autonomous driving vehicles: a system-level approach. *ACM SIGBED Review*, 2013.

[22] Raghu K Ganti, Nam Pham, Hossein Ahmadi, Saurabh Nangia, and Tarek F Abdelzaher. Greengps: a participatory sensing fuel-efficient maps application. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 151–164. ACM, 2010.

[23] Kerem Koprubasi. *Modeling and control of a hybrid-electric vehicle for drivability and fuel economy improvements.* PhD thesis, The Ohio State University, 2008.

[24] Apu platform. *http://www.pcengines.ch/apu.htm.*

[25] Tsang-Wei Lin, Sheue-Ling Hwang, and Paul A Green. Effects of time-gap settings of adaptive cruise control (acc) on driving performance and subjective acceptance in a bus driving simulator. *Safety science*, 47(5):620–625, 2009.

[26] Sarah M Loos, André Platzer, and Ligia Nistor. Adaptive cruise control: Hybrid, distributed, and now formally verified. In *FM 2011: Formal Methods*, pages 42–56. Springer, 2011.

[27] Kanok Boriboonsomsin, Alexander Vu, and Matthew Barth. Eco-driving: pilot evaluation of driving behavior changes among us drivers. *University of California Transportation Center*, 2010.

[28] Johannes Tulusan, Thorsten Staake, and Elgar Fleisch. Providing eco-driving feedback to corporate car drivers: what impact does a smartphone application have on their fuel efficiency? In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 212–215. ACM, 2012.

[29] Dominik Lang, Thomas Stanger, and Luigi del Re. Opportunities on fuel economy utilizing v2v based drive systems. Technical report, SAE Technical Paper, 2013.

[30] Dominik Lang, Roman Schmied, and Luigi Del Re. Prediction of preceding driver behavior for fuel efficient cooperative adaptive cruise control. *SAE International Journal of Engines*, 7(2014-01-0298):14–20, 2014.

[31] Derick A Johnson and Mohan M Trivedi. Driving style recognition using a smartphone as a sensor platform. In *Intelligent Transportation Systems (ITSC), International IEEE Conference on.* IEEE, 2011.

[32] Yan Wang, Jie Yang, Hongbo Liu, Yingying Chen, Marco Gruteser, and Richard P Martin. Sensing vehicle dynamics for determining driver phone use. *ACM Mobisys*, 2013.

[33] Jakob Eriksson, Lewis Girod, Bret Hull, Ryan Newton, Samuel Madden, and Hari Balakrishnan. The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring. In *MobiSys*, 2008.

[34] Prashanth Mohan, Venkata N. Padmanabhan, and Ramachandran Ramjee. Nericell: Rich monitoring of road and traffic conditions using mobile smartphones. SenSys '08. ACM, 2008.

[35] Progressive. Linking driving behavior to automobile accidents and insurance rates. *Report*, 2012.

[36] R Zanasi, A Visconti, G Sandoni, and R Morselli. Dynamic modeling and control of a car transmission system. In *2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. IEEE, 2001.

[37] C Canudas de Wit and P Tsiotras. Dynamic tire friction models for vehicle traction control. In *Proceedings of the 38th IEEE Conference on Decision and Control*. IEEE, 1999.

[38] Valner Brusamarello, Alexandre Balbinot, Luiz Carlos Gertz, and André CervieÌĄri. Dynamic torque measurement for automotive application. In *Instrumentation and Measurement Technology Conference (I2MTC)*. IEEE, 2010.