FUEL OPTIMAL CRUISING STRATEGY FOR VEHICLE DESIGN

V Padmanaban [1]
Assistant Professor,
Dept. of Electronics and Instrumentation Engineering
Sri Sairam Engineering College.

Dr.Nalini.M [2]
Assistant Professor
Dept. of Computer Science and Engineering
Saveetha School of Engineering

1.ABSTRACT

This project presents a control strategy to reduce volume of fuel consumption to the lowest by using effective transmission alert system in a step gear mechanical transmission. Velocity is converted into a minute voltage. The voltage serves as an input to the Arduino/Raspberry pi Micro- controller. The microcontroller is programmed in such a way that the system sends a sound/LED/vibrating alert during transmission. The system aims to be portable and compatible with any vehicle.

Keywords:* *Mechanical transmission, voltage, Arduino/Raspberry pi Micro- controller, LED.

2.INTRODUCTION

Heavy automobiles consume comparatively more fuel than other vehicles. Public transport which comprises mostly of heavy automobiles does not have automatic transmission systems which lead to loss of mileage and a larger volume of fuel consumption. Thus considering fuel market, consumption rates primarily, a new look to the vehicle design would be sensible in the current scenario. Effective transmission will aim to cut off fuel consumption rate. A driver's performance is monitored using a cloud-based system. Further improvements involved in the collecting data set of performances and putting machine learning to compare and rate the driver based on their performances.

3.EXISTING SYSTEM

The commonly used cruising strategy is the CS operation, i.e., travelling at a fixed speed, which is frequently adopted by drivers and cruising control systems. However, due to the strong nonlinearity of engine efficiency, CS is not always the fuel optimal strategy. Pulse and Glide (PnG) type cruising strategy, also called accelerating and coasting strategy, has been proven to achieve better fuel economy by theoretical calculations and experiments, and the fuel saving rate can be up to 20%.

PnG strategy is a dynamic cruising strategy: drivers or control the systems need to accelerate the vehicle from a low speed to a higher speed first, called the pulse (or accelerating) phase, and then coast to the lower speed, called glide (or coasting) phase, with the average speed equal to the expected velocity. Compared to CS, PnG requires periodical fluctuations on vehicle speed and dynamic controls on engine and transmission. Although this strategy has great potential to reduce fuel consumption, not all accelerating operations can achieve the goal of fuel saving. The dynamic control inputs, i.e., engine output torque and gear position, should be carefully selected. Improper control inputs, e.g., following the qualitative tips like “accelerate softly”, may even cause more fuel consumption than CS.

4. PROPOSED SYSTEM

The PnG strategy is shown in Fig. 1.1. When the expected cruising speed is Ve, the vehicle first accelerates from a low speed V_{min} to a higher speed Vmax in the “accelerating phase” and then coast down to V_{min} in the “coasting phase,” with the average speed equal to the desired velocity Ve. This strategy can be used in situations with sparse traffic, e.g., driving on low density highways and suburban roads. Our goal is to determine the optimal gear position and engine torque profile to minimize the fuel consumption. This problem naturally fits into the optimal control framework with engine torque and gear position as the control variables. The performance index, state space equations, and constraints of this fuel-optimal control problem are described in the following. In the PnG strategy, the coasting rules significantly affect the fuel saving performance, which are often ignored. Though coasting with engine idling and transmission in neutral is often selected by default, here we study four types of coasting strategies:

(a) PnG-N-O, coasting with transmission in neutral and engine shut off;

(b) PnG-N-I, coasting with transmission in neutral and engine idling;

(c) PnG-G-D, coasting with transmission in gear, and the gears for accelerating and coasting can be different;

(d) PnG-G-S, coasting with transmission in gear and the gears for accelerating and coasting are same, which can avoid frequent gear shifting.

The models of these four PnG strategies are different but similar, so we only present the model of PnG-G-S in detail. Since the speed fluctuations of PnG is generally small, to avoid excessive complexity, we assume that only a particular gear is used to accelerate and to coast, i.e., gear shifting only happens at the beginning of accelerating or coasting.

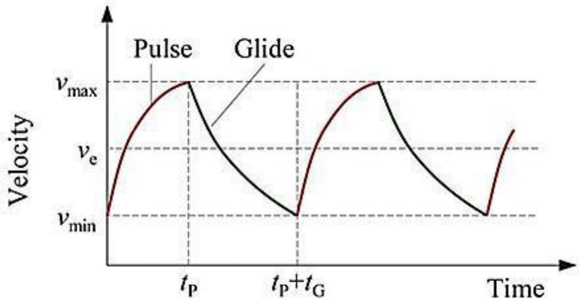


Fig.4.1. Concept of the PnG Strategy

5. USE OF SPEED AS FOR PERFORMANCE EVALUATION

Driver attitudes towards what constituted a safe travel speed were related to speeding behaviour, although not as strongly as driver age and accident history. This is further evidence of the need for an educational campaign aimed at changing this attitude.

6. BLOCK DIAGRAM:

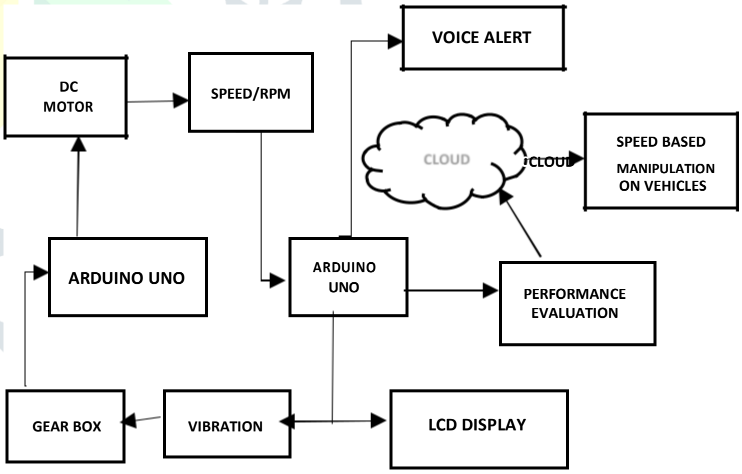


Fig.6.1. Block Diagram of Fuel optimization for vehicles

Previous experience suggests that achieving a change in attitude may require a long-term program of measures using a mufti-facet approach.

There is a need for a proper review of devices for limiting vehicle speeds in urban and rural areas, including their availability, suitability and likely costs and benefits. The question of how they could be introduced into the vehicle fleet also needs to be addressed. So the value of vehicle speed can be used as a data for evaluating quality of a driver and additional researches.

7. COMPONENTS

7.1 ARDUINO UNO

Arduino/Genuino Uno is a 8 bit microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins. It has a rated voltage of 5V.

7.2 L293D MOTOR DRIVER

L293D is a dual H-bridge motor driver integrated circuit (IC). Motor drivers act as current amplifiers since they take a low-current control signal and provide a higher-current signal. . Input logic 00 or 11 will stop the corresponding motor. Logic 01 and 10 will rotate it in clockwise and anticlockwise directions, respectively.

7.3 ARDUINO DECIMILIA

The Arduino Diecimila is a microcontroller board based on the ATmega168 It has 14 digital input/output pins. The Arduino Decimilia can be powered via the USB connection or with an external power supply. The board can operate on an external supply of 6 to 20 volts.

7.4 APR VOICE MODULE:

The APR9600 provided all the necessary features for recording and playing the audio with very fewer external components at a very low cost. The chip was manufactured by a Taiwan based company called APLUS Integrated Circuits Inc.

7.5 LIQUID CRYSTAL DISPLAY

LCD (Liquid Crystal Display) screen is an electronic display module. LCDs are economical; easily programmable; have no limitation of displaying special & even custom characters (unlike in seven segments), animations and so on.

7.6 VIBRATION MOTOR

An eccentric rotating mass vibration motor (ERM) uses a small unbalanced mass on a DC motor, when it rotates it creates a force that translates to vibrations.

7.7 HALL EFFECT SENSOR

A Hall Effect sensor is a transducer that varies its output voltage in response to a magnetic field. Hall Effect sensors are used for proximity switching, positioning, speed detection, and current sensing applications.

7.8 DC MOTOR

A DC motor is any of a class of rotary electrical machines that converts direct current electrical energy into mechanical energy.

7.9 JOY STICK

A joystick is an input device consisting of a stick that pivots on a base and reports its angle or direction to the device it is controlling.

8. WORKING PRINCIPLE

The Arduino microcontroller receives it Input from the Joystick. The Joystick consists of four normally open switches which conduct when the joystick is pushed to the required axis. The Arduino must be programmed suitably to manipulate speed using inputs from joystick.

The speed pulses to the maximum value of speed in the given gear position and a vibrating alert is given to the user to shift to the next gear. On failing to shift to the next gear, the speed glides to the lowest value of gear by cutting off supply to the motor.

The same procedure is repeated for all the gears in the vehicle and thus successfully achieving pulse and glide system in a vehicle. During operation of the vehicle in the last gear, the vibrating alert is replaced by a voice alert as further shift is not required. In this phase, the system provides qualitative advice to the driver such as “Accelerate smoothly”. The voice alert can also be used for other gear levels as well.

We can use the value of speed for further researches and the application provided further. Then, we compare the optimized driving styles to the original driving styles and evaluate the effectiveness of the optimization framework. Finally, we use this proposed model to develop a real-time feedback system, including visual instructions, to enable drivers to alter their driving styles in response to actual driving conditions to improve fuel efficiency.

9. ALGORITHM

9.1 START

All the Arduino boards, voice module and LCD display are powered by supply.

9.2 LCD DISPLAY IS ON

LCD display data pins DB1, DB2, DB3, DB4 are connected to Arduino digital pins 2,3,4,5 and a potentiometer is connected to pin 3 of LCD to adjust the contrast.

9.3 RPM IS DISPLAYED

RPM of the motor is displaying simultaneously in LCD and the rpm is calculated by using a Hall Effect sensor. The output of the Hall Effect sensor is connected to the analog pin of Arduino. From the Hall Effect sensor, the value of RPM is obtained using the formula

```
t = millis ();

cur_t= millis ();

 r=1000000*60/ (cur_t-t)
```

Cur_t and t are time variables.

When speed limit of first gear is reached, the vibration motor is on to indicate the gear shift or the gliding for first gear takes place.

9.4 SHIFT TO GEAR 2

When gear 2is pressed, the Digital pin 8of Arduino is high. When speed limit of 2nd gear is reached, the vibration motor is on to indicate the gear shift or the gliding for 2ndgear takes place.

9.5 SHIFT TO GEAR 3

When gear 3is pressed, the Digital pin 9of Arduino is high. When speed limit of 3rdgear is reached, the vibrator is on to indicate the gear shift or the gliding for 3rdgear takes place.

9.6 SHIFT TO GEAR 4

When gear 4is pressed, the Digital pin 10 of Arduino is high. Maximum speed is reached. When the maximum speed of 4th gear is reached, a voice alert will be given to decelerate and then the gliding takes place from top speed to the optimal speed.

9.7 STOP

The algorithm is written mainly for the working of the speed control procedure and alert system which forms the basics of future developments in the project. Further additions to the project can be made by adding more functions to the algorithm without changing the order of execution.

10. RECOMMENDOR SYSTEM

To create a user profile, the system mostly focuses on two types of information:

1. A model of the user's preference.
2. A history of the user's interaction with the recommender system.

Basically, these methods use an item profile (i.e., a set of discrete attributes and features) characterizing the item within the system. The system creates a content-based profile of users based on a weighted vector of item features. The weights denote the importance of each feature to the user and can be computed from individually rated content vectors using a variety of techniques. Simple approaches use the average values of the rated item vector while other sophisticated methods use machine learning techniques such as Bayesian Classifiers, cluster analysis, decision trees, and artificial neural networks in order to estimate the probability that the user is going to like the item.

Direct feedback from a user, usually in the form of a like or dislike button, can be used to assign higher or lower weights on the importance of certain attributes

In the following program to suggest the recommended drivers based on speed, We use two parameter vectors

$$X = \begin{bmatrix} - (x^{(1)})^T - \\ - (x^{(2)})^T - \\ \vdots \\ - (x^{(n_m)})^T - \end{bmatrix}, \quad \text{Theta} = \begin{bmatrix} - (\theta^{(1)})^T - \\ - (\theta^{(2)})^T - \\ \vdots \\ - (\theta^{(n_u)})^T - \end{bmatrix}.$$

The i-th row of X corresponds to the feature vector x(i) for the i-th driver, and the j-th row of θ corresponds to one parameter vector θ(j), for the j-th user.

The stepwise execution of the program is given as follows

☐ Loading Driver Dataset

☐ Collaborative Filtering learning algorithm

☐ Collaborative Filtering gradient

☐ Regularized cost function

☐ Learning driver recommendations.

11. COLLABORATIVE FILTERING LEARNING ALGORITHM

The collaborative filtering algorithm in the setting of driver recommendations considers a set of n-dimensional parameter vectors x(1)......... $x(n_m)$ and θ (1)..........θ (n_u), where the model predicts the rating for driver i by user j as $y^{(i,j)}$ = (θ $(j))^T)$ x(i). Given a dataset that consists of a set of ratings produced by some users on some drivers, we can learn the parameter vectors x(1) .... $x(n_m)$; θ (1)..........θ (n_u), that produce the best fit (minimizes the squared error).

The collaborative filtering cost function (without regularization) is given by

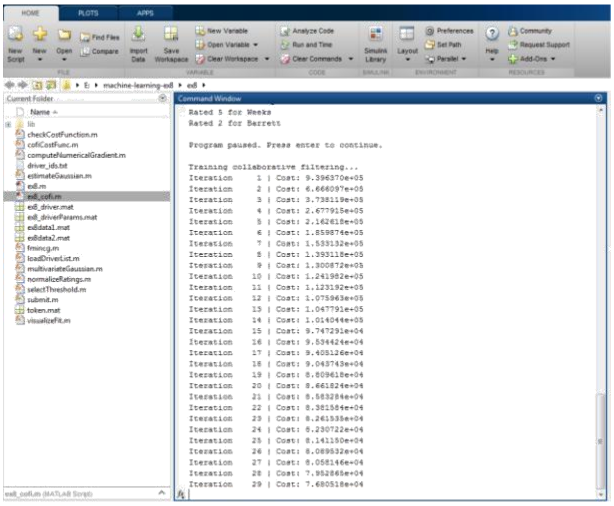
$$J(x^{(1)},\dots,x^{(n_m)},\theta^{(1)},\dots,\theta^{(n_u)}) = \frac{1}{2}\sum_{(i,j):r(i,j)=1}((\theta^{(j)})^T x^{(i)} - y^{(i,j)})^2.$$

12. COLLABORATIVE FILTERING GRADIENT

To implement the gradient (without regularization) specifically, we have to complete the code in cofiCostFunc.m to return the variables X grad and Theta grad. Note that X grad should be a matrix of the same size as X and similarly, Theta grad is a matrix of the same size as Theta. The gradient of the cost function is given by**:**

$$\frac{\partial J}{\partial x_k^{(i)}} = \sum_{j:r(i,j)=1}((\theta^{(j)})^T x^{(i)} - y^{(i,j)})\theta_k^{(j)}$$

$$\frac{\partial J}{\partial \theta_k^{(j)}} = \sum_{i:r(i,j)=1}((\theta^{(j)})^T x^{(i)} - y^{(i,j)})x_k^{(i)}.$$



```
Rated 5 for Weeks
Rated 2 for Barrett

Program paused. Press enter to continue.

Training collaborative filtering...
Iteration     1 | Cost: 9.396370e+05
Iteration     2 | Cost: 6.666097e+05
Iteration     3 | Cost: 3.738119e+05
Iteration     4 | Cost: 2.677915e+05
Iteration     5 | Cost: 2.162618e+05
Iteration     6 | Cost: 1.859874e+05
Iteration     7 | Cost: 1.533132e+05
Iteration     8 | Cost: 1.393118e+05
Iteration     9 | Cost: 1.300872e+05
Iteration    10 | Cost: 1.241982e+05
Iteration    11 | Cost: 1.123192e+05
Iteration    12 | Cost: 1.075963e+05
Iteration    13 | Cost: 1.047791e+05
Iteration    14 | Cost: 1.014044e+05
Iteration    15 | Cost: 9.747291e+04
Iteration    16 | Cost: 9.534424e+04
Iteration    17 | Cost: 9.405126e+04
Iteration    18 | Cost: 9.043743e+04
Iteration    19 | Cost: 8.809618e+04
Iteration    20 | Cost: 8.661824e+04
Iteration    21 | Cost: 8.583284e+04
Iteration    22 | Cost: 8.381584e+04
Iteration    23 | Cost: 8.261535e+04
Iteration    24 | Cost: 8.230722e+04
Iteration    25 | Cost: 8.141150e+04
Iteration    26 | Cost: 8.089532e+04
Iteration    27 | Cost: 8.058146e+04
Iteration    28 | Cost: 7.952865e+04
Iteration    29 | Cost: 7.680518e+04
```

Fig.12.1. Collaborative filtering gradient

The function returns the gradient for both sets of variables by unrolling them into a single vector. After completing the code to compute the gradients, the script exi. confirm will run a gradient check (checkCostFunction) to numerically check the implementation of your gradients. If the implementation is correct, the analytical and numerical gradients match up closely.

13. REGULARIZED COST FUNCTION

The cost function for collaborative filtering with regularization is given by

$$J(x^{(1)},\dots,x^{(n_m)},\theta^{(1)},\dots,\theta^{(n_u)})=\frac{1}{2}\sum_{(i,j):r(i,j)=1}\left((\theta^{(j)})^T x^{(i)}-y^{(i,j)}\right)^2+\left(\frac{\lambda}{2}\sum_{j=1}^{n_u}\sum_{k=1}^{n}(\theta_k^{(j)})^2\right)+\left(\frac{\lambda}{2}\sum_{i=1}^{n_m}\sum_{k=1}^{n}(x_k^{(i)})^2\right)$$

14. REGULARIZED GRADIENT

Now that we have implemented the regularized cost function, we should proceed to implement regularization for the gradient. We should add to the implementation in cofiCostFunc.m to return the regularized gradient by adding the contributions from the regularization terms.

Note that the gradient for the regularized cost function is given by:

$$\frac{\partial J}{\partial x_k^{(i)}}=\sum_{j:r(i,j)=1}\left((\theta^{(j)})^T x^{(i)}-y^{(i,j)}\right)\theta_k^{(j)}+\lambda x_k^{(i)}$$

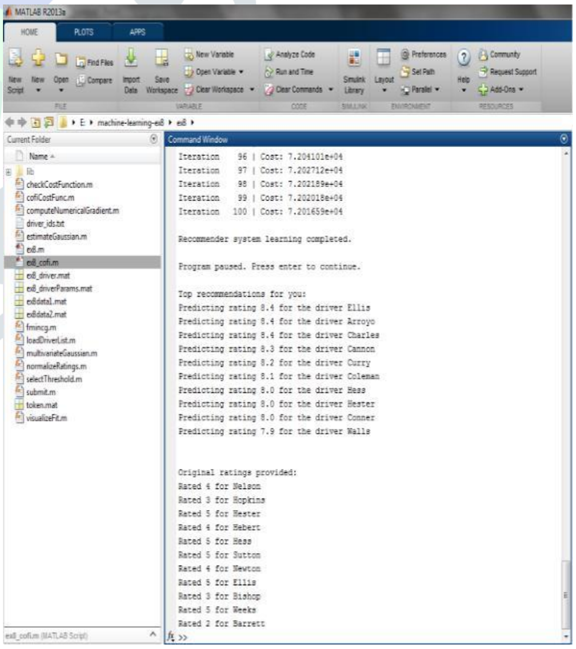
$$\frac{\partial J}{\partial \theta_k^{(j)}}=\sum_{i:r(i,j)=1}\left((\theta^{(j)})^T x^{(i)}-y^{(i,j)}\right)x_k^{(i)}+\lambda \theta_k^{(j)}.$$

15. LEARNING DRIVER RECOMMENDATIONS

After we have finished implementing the collaborative filtering cost function and gradient, we can now start training the algorithm to make driver recommendations for you. In the next part of the confirm script, you can enter your own driver preferences, so that later when the algorithm runs, you can get your own driver recommendations!

We have filled out some values according to our own preferences, but it can be changed according to people's own preferences. The list of all drivers and their speed in the dataset can be found listed in the file driver_idx.txt.

16. FINAL OUTPUT



```
Iteration    96 | Cost: 7.204101e+04
Iteration    97 | Cost: 7.202712e+04
Iteration    98 | Cost: 7.202189e+04
Iteration    99 | Cost: 7.202018e+04
Iteration   100 | Cost: 7.201659e+04

Recommender system learning completed.

Program paused. Press enter to continue.

Top recommendations for you:
Predicting rating 8.4 for the driver Ellis
Predicting rating 8.4 for the driver Arroyo
Predicting rating 8.4 for the driver Charles
Predicting rating 8.3 for the driver Cannon
Predicting rating 8.2 for the driver Curry
Predicting rating 8.1 for the driver Coleman
Predicting rating 8.0 for the driver Hess
Predicting rating 8.0 for the driver Hester
Predicting rating 8.0 for the driver Conner
Predicting rating 7.9 for the driver Walls


Original ratings provided:
Rated 4 for Nelson
Rated 3 for Hopkins
Rated 5 for Hester
Rated 4 for Hebert
Rated 5 for Hess
Rated 5 for Sutton
Rated 4 for Newton
Rated 5 for Ellis
Rated 3 for Bishop
Rated 5 for Weeks
Rated 2 for Barrett
```

Fig.16.1. Collaborative filtering gradient

17. CONCLUSION

In this project, a theoretical meta model for Pulse and Glide system and Data acquisition has been presented. It is evident that pulse and Glide system is more effective driving strategy for a vehicle than cruising as power supply is cut off during the glide region but the motor continues to decelerate with a constant gradient. It is an improved approach to change the way vehicles are driven s no alternate fuel has been found. An effective Driving Strategy will cut off fuel supply by 15% and adding an alert system will improve the efficiency by 13%. While adding this system to a vehicle, it is essential to have a microcontroller operated valve which will cut off and allow fuel into the engine only after the signal from the microcontroller. The model can be further advanced by uploading values of the driver system to a centralised cloud present in every traffic signal .This will allow vehicle tracking and speed travelled by the vehicle thus averting accidents. This Project can be simply implemented in a motorcycle using an Arduino controllers and Solenoid valves to the fuel line.

The benefits of this system include fuel economy, avoid the risk of accidents, provide desired drivers to frequent taxi travellers and provide quicker vehicle identification systems.

REFERENCES

[1] A. A.Malikopoulos, Real-Time, Self-Learning Identification and StochasticOptimal Control of Advanced Powertrain Systems. Cambridge, U.K.:ProQuest, Sep. 2011.
[2] V. Manzoni, A. Corti, P. De Luca, and S. M. Savaresi, “Driving style estimation via inertial measurements,” in Proc. 13th Int. IEEE ITSC, Madeira Island, Portugal, 2010, pp. 777–782.
[3] H. Liimatainen, “Utilization of fuel consumption data in an Eco drivingincentive system for heavy-duty vehicle drivers.” IEEE Trans. Intell. Transp. Syst.,Dec. 2011.
[4] A. Riener, “Subliminal persuasion and its potential for driver behaviour adaptation,” IEEE Trans. Intell. Transp. Syst., vol. 13, no. 1 Mar. 2012.
[5] M. A. S. Kamal, M. Mukai, J. Murata, and T. Kawabe, “Ecological vehicle control on roads with up-down slopes,” IEEE Trans. Intell. Transp. Syst., vol. 12, no. 3,Sep. 2011.
[6] K. Takeda, C. Miyajima, T. Suzuki, P. Angkititrakul, K. Kurumida, Y. Kuroyanagi, H. Ishikawa, R. Terashima, T. Wakita, M. Oikawa, and Y. Komada, “Self-coaching system based on recorded driving data: Learning from one’s experiences,” IEEE Trans. Intell. Transp. Syst., vol. 13, no. 4 Dec. 2012.
[7] Q. B. Dam, “The MPG survey: Questioning the biased perception of automobile fuel economy,” in Proc. IEEE Energy 2030 Conf., 2008,
[8] J. Van Mierlo, G. Maggetto, E. Van De Burgwal, and R. Gense, “Driving style and traffic measures—Influence on vehicle emissions and fuel consumption,” Proc. Inst. Mech. Eng., Part D, J. Autom. Eng., vol. 218,
[9] E. Ericsson, “Independent driving pattern factors and their influence on fuel-use and exhaust emission factors,”
[10] Andreas A. Malikopoulos, Member, IEEE, and Juan P. Aguilar,An Optimization Framework for Driver Feedback Systems
[11] Andreas Riener,Johann Reder,Collective Data Sharing to Improve on Driving Efficiency and Safety
[12]. Fuel-Optimal Cruising Strategy for Road Vehicles with Step-Gear mechanical transmission, Shaobing Xu, ShengboEben Li, Xiaowu Zhang, IEEE Transactions on Intelligent Transportation Systems, Volume: 16, Issue: 6, pp 3496 – 3507 Dec. 2015.