A.Ghaffari[*]
Professor

A.R. Khodayari[†]
Assistant Professor

S. Salehinia[‡]
M.S

M. Nouri-Khajavi[§]
Assistant Professor

M.R. Arbab Tafti[**]
Assistant Professor

Model Predictive Control System Design using ARMAX Identification Method for Car-following Behavior

The control of car following is essential due to its safety and its operational efficiency. For this purpose, this paper builds a model of car following behavior based on ARMAX structure from a real traffic dataset and design a Model Predictive Control (MPC) system. Based on the relative distance and relative acceleration of each instant, the MPC predicts the future behavior of the leader vehicle and according to this behavior, the acceleration of the follower vehicle is controlled. Validation of the presented controller is done by comparing the behavior of the controller with the human drivers. Results show that the MPC controller has a behavior much safer than that of real drivers and it can provide a pleasant trip for passengers.

Keywords: ARMAX identification, model predictive control (MPC), Intelligent Transportation Systems, car following behavior, modeling and control

1 Introduction

It has been estimated that mankind currently devotes over 10 million man-years each year to driving the automobile, which on demand provides a mobility unequaled by any other mode of transportation. And yet, even with the increased interest in traffic research, we understand relatively little of what is involved in the "driving task" [1]. Various theories attempt to describe the vehicular traffic flow process. One class of such theories, called car-following theories, is based on the follow-the-leader concept, in which rules of how a driver follows his/her immediate leading vehicle are established based on both experimental observations and theoretical (i.e., psychological) considerations [2]. Car following is quite common in many traffic fields such as railway, highway and etc. Car following is a crucial tactical-level model for a microscopic simulation system. Car following describes the longitudinal action of

[*] Professor, Mechanical Engineering Department, K. N. Toosi University of Technology, Tehran, ghaffari@kntu.ac.ir
[†] Corresponding Author, Assistant Professor, Department of Mechanical Engineering, Pardis Branch, Islamic Azad University, Tehran, arkhodayari@dena.kntu.ac.ir
[‡] M.S., Mechanical Engineering Department, Shahid Rajaee Teacher Training University, Tehran, saeed.salehinia@gmail.com
[§] Assistant Professor, Mechanical Engineering Department, Shahid Rajaee Teacher Training University, Tehran, mehrdadnouri@ieee.org
[**] Assistant Professor, Mechanical Engineering Department, Shahid Rajaee Teacher Training University, Tehran, mr.tafti@gmail.com

a driver when it follows another car and tries to maintain a safe distance to the leading car, as shown in figure (1).
One of the major achievements is the control laws for collision avoidance while the front car brakes suddenly in emergency in the course of their following operation [3-5]. However, due to the complexity of the car following problem, the current control of car following operation mainly dependents on the drivers' subjective judgment and their corresponding behavior. Its complexity could be summarized as follow [4]. The control of car following is a

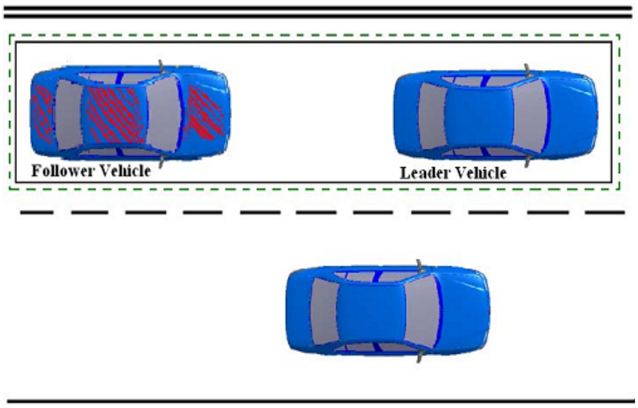


Figure 1 Driver-Vehicle Unites; follower vehicle (FV) and leader vehicle (LV) in car following behavior

problem with more constraints and multi-objective optimization; the control of car following is similar to the pursuit and escape problem in differential game, but somehow different from, which study the bilateral dynamic optimal control laws between the pursuer and the escapee; the control of car following belongs to the problems of keeping the safe distance in the reasonable value so that the time of pursuit be minimum.

Apart from the impacts of road and vehicle on driving, in car following behavior, the effects of the human driver have a major role in how the process is performed. Some examples of human drivers on driving behaviors are age, gender, emotional factors, and etc. It is almost impossible to define all these factors to place them in one model, and either if it is possible, the resulting model will be very complex.

The problem of designing a good control system is basically that of matching the dynamic characteristics of a process by those of the controller. In other words, if the dynamics of the process and the characteristics of the disturbances affecting it are known, then the controller that will provide the desired closed loop performance can be designed. Modern approaches to controller design therefore presuppose that a suitable description (model) of the process to be controlled is available [6]. But the desired model isn't always available and instead the incumbent data are accessible, so the next step is to estimate the model with identification methods.

A time series is a sequence of data points, measured typically at successive time instants spaced at uniform time intervals. We can construct time series from several sub-series and in a variety of configurations. In modeling process, the major task is to determine the most appropriate time-series structure and estimating the parameters of the polynomials.

Model predictive control is a form of control in which the current control action is obtained by solving, at each sampling instant, a finite horizon open-loop optimal control problem, using the current state of the plant as the initial state. The optimization yields an optimal control sequence and the first control in this sequence is applied to the plant. An important advantage of this type of control is its ability to cope with hard constraints on controls and states [7].

The MPC is probably the most applied advanced control technique in the industry due to several reasons:

- It handles multivariable control problems naturally.
- It can take account of actuator limitation.
- It can handle constraints on the inputs and the outputs of the process in a systematic way during the design and the implementation of the controller.
- It can handle changes in system parameters or system structure (including sensor or actuator failures) by regularly updating the parameters and the structure of the prediction model [8].

However, the use of MPC is not limited to the industry. The many advantages that MPC offers are also relevant for traffic control. In fact, MPC has already been extended to conventional roadside-based non-intelligent vehicle traffic management, traffic management and intelligent vehicles control [9-11]. It has also been used as driver assistant for ecological driving [12] and an approach to design adaptive cruise control [13]. In spite of mentioned cases, it seems that MPC has been widely used in vehicle platoon control, autonomous vehicles in Intelligent Vehicle Highway Systems (IVHS) and control systems such as Advanced Cruise Control (ACC). But it hasn't been used for driving behaviors including the human driver, e.g., for car following behavior [14].

Stability analysis is a very important task in the mathematical control theory and its practical application. It is a fundamental property of a dynamic system that summarizes the long-term behavior of that system. In engineering practice the control systems are designed so that stability is preserved in various classes of uncertainties – this property is known as the robust stability [15]. The stability problem of MPC based control can be solved by correct designing [8]. The MPC has been successfully applied to motorway traffic management too [11, 16].

Also, in this study, robustness analysis, i.e. analysis of the robustness properties of standard MPC designed for a nominal model is distinguished without taking into account uncertainty, and synthesis of MPC algorithms which are robust by construction. The robustness analysis of MPC control loops is more difficult than the synthesis, where the controller is designed in such a way that it is robustly stabilizing. This is not unlike the situation in the nominal case, where the stability analysis of a closed loop Multiple-Input-Multiple-Output (MIMO) system with multiple constraints is essentially impossible. On the other hand, the MPC technology leads naturally to a controller such that the closed loop system is guaranteed to be stable [17].

In this study, identification of a human driving behavior based on actual measured car following data is performed and a model predictive controller will be designed to control the car following behavior as the most common behavior in real traffic flow. Then the result of the controller will be compared with actual traffic data. The remaining parts of this paper are organized as follows: Section 2 describes the procedures of designing a car following behavior model based on ARMAX identification structure which will be employed as the plant of the control system. Section 3 presents the model predictive control system design and in section 4, the proposed controller will be evaluated. At last, the conclusion is given in section 5.

2 Model structure

2.1 Alternative for Modeling

Automotive first principle models range from precise computational (distributed) fluid dynamics models, in which the flows and possibly combustion is modeled precisely, to mostly lumped models for vehicle control in multi-body languages like ADAMS, or partly produced directly from 3D drawings product e.g. in CATIA. In the case of modeling human driving behavior, if it were to contain all parameters and behavior, models may become very complex. First principle models have always been the first choice of designers as they allow physical insight in the systems and to experiment with design changes.

This result from a common characteristic of all first principle models; because they contain the information known a priori, i.e., what the modeler is aware of at the start of the project. While this is unavoidable in some cases, e.g. when the system does not yet exist, it bears the enormous disadvantage that all deviations from expectation will not include. As a consequence, this approach will work with systems which are relatively easy to model, like the basic vehicle dynamics (2-DOF bicycle model) or the effect of speaking with cell phone on driving performance while driving [18, 19].

The industrial community has mostly a different approach, especially in the case of engine control. Control candidates with high parameter numbers are set up, and tuned experimentally with an enormous effort, yielding in practice an approximated inverse model and is used to design a reliable, yet utterly heuristic controller. The main reason for this is that first principle models tend to be too simple to describe the real systems to the necessary degree of precision necessary to meet the increasing performance requirements. Also in the case of human driving behavior, this kind of approach isn’t effective and cannot include all scenarios and behaviors.

This essential difference in modeling is mirrored in the practical development of the models and successful applications: while control of longitudinal dynamics as in [20], or the lateral approaches as in [21] can be built upon first principle models with few unknown parameters, the same is not true for car following behavior. Even though first principle models are always the most complete information source, frequently they cannot be used for control design or simulating human behavior, as they are too complex for on-line use, they must be approximated. So, a natural question could be: why bother for a good first principle model and not look directly for a good approximation?

3.1 Data-Based Model

The natural alternative to first principle models are data-based models, which typically fix a parameterized candidate model without reference to the physics of the real plant and requires large quantities of data in order to compute the optimal estimation of the parameters. The main drawbacks of purely data-based models are, of course, the need of data, but also the frequently unknown extrapolation properties: as in all nonlinear identification problems, the choice of the excitation is critical, because actually all moments of the exciting signal are critical. For some cases, the candidate model structure will in general not contain the true model; therefore system identification becomes essentially an approximation tool. Also “practical” approximators ,e.g. linear parameter varying methods as well as simple structures which can be easily included in the optimization like the one used by have been used in the quest for suitable models, i.e., for those who capture the main nonlinearities while remaining sufficiently simple. Moreover, the dimension of data-based models increases significantly (in comparison to a linear approach).

Against this background, there is a rationale for combining both approaches so to get simple and high performing models. There are at least two ways to do this, the first one, the so called gray-box model, uses physical understanding to describe explicitly aspects of the models which can be described analytically and combines them with data-based models. A second possibility consists in looking for global patterns in the observed data which can be expressed analytically (with few parameters) and thus infer not only the parameters of a given model but also its structure from the data. It is interesting to note that the real critical issue is not the method used for the identification, not even the class of functions, but the specific data [22].

4.1 Dataset

In order to design a car following behavior model, a dataset of car following behavior is needed. So, real car following data from US Federal Highway Administration's NGSIM dataset is used [23]. In June (2005), a dataset of trajectory data of vehicles travelling during the morning peak period on a segment of Interstate 101 highway in Emeryville (San Francisco), California has been made using eight cameras on top of the 154m tall 10 Universal City Plaza next to the Hollywood Freeway US-101. As shown in figure (2), on a road section of 640m, 6101 vehicle trajectories have been recorded in three consecutive 15 minute intervals.

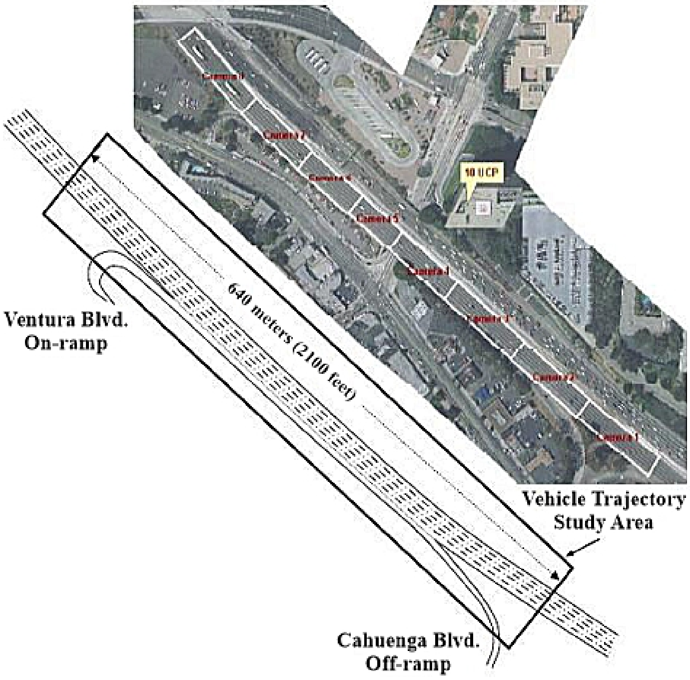


Figure 2 A Segment of Interstate 101 Highway in Emeryville, San Francisco, California [24]

This dataset has been published as the "US-101 Dataset". The dataset consists of detailed vehicle trajectory data on a merge section of eastbound US-101. The data is collected in 0.1 sec intervals. Any measured sample in this dataset has 18 features of each Driver-Vehicle Unites (DVU) in any sample time, such as longitudinal and lateral position, velocity, acceleration, time, number of road, vehicle class, front vehicle and etc. [24].

Based on a thorough analysis of this dataset, the data of the vehicles which had a car follow behavior were extracted. This datasets was comprised of the data of three vehicle classes; motorcycles, autos and trucks. For this study, the data of only autos was used.

The data extracted from the datasets, seem to be unfiltered and exhibit some noise artifacts, so these data must be filtered like [25, 26]. A moving average filter has been designed and applied to all data before any further data analysis. A comparison of the unfiltered and filtered data of the acceleration of the FV in one maneuver is shown in figure (3).

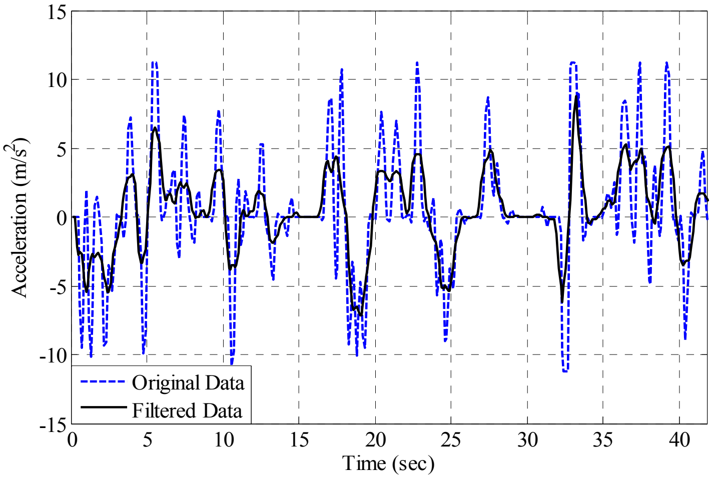


Figure 3 Comparison of unfiltered and filtered data

5.1 ARMAX model

The ARMAX is a linear polynomial structure to model time series data (For frequency-domain data, Output-Error model must be used). In this section, the structure of ARMAX will be looked at briefly.

Time series data have a natural temporal ordering. This makes time series analysis distinct from other common data analysis problems, in which there is no natural ordering of the observations. A general time series representation can be given as Eq. (1):

$$A(q^{-1})y(t) = \frac{B(q^{-1})}{F(q^{-1})}u(t) + \frac{C(q^{-1})}{D(q^{-1})}e(t) \tag{1}$$

Time series models are special cases of polynomial models for systems without measured inputs. Auto Regressive Moving Average Models with exogenous inputs (ARMAX) are typically applied to auto correlated time series data. ARMAX is a generalized model for discrete, time-varying systems. The ARMAX model structure is stated in Eq. (2):

$$\begin{aligned} y(t) + a_1 y(t-1) + \ldots + a_{n_a} y(t-n_a) = b_1 u(t-n_k) + \ldots + b_{n_b} u(t-n_k-n_b+1) \\ + c_1 e(t-1) + \ldots + c_{n_c} e(t-n_c) + e(t) \end{aligned} \tag{2}$$

where $y(t)$ is the output at time t, n_a is the number of poles, n_b is the number of zeroes plus 1, n_c is the number of C coefficients, n_k is the number of input samples that occur before the input affects the output, also called the dead time in the system, $y(t-1)\ldots y(t-n_a)$ is the previous outputs on which the current output depends, $u(t-n_k)\ldots u(t-n_k-n_b+1)$ is the previous and delayed inputs on which the current output depends and $e(t-1)\ldots e(t-n_c)$ is white noise disturbance value. A more compacted way to represent the ARMAX model in discrete form is shown in Eq. (3).

$$A(q^{-1})y_k = B(q^{-1})u_k + C(q^{-1})e_k \tag{3}$$

which $A(q^{-1}), B(q^{-1})$ and $C(q^{-1})$ are polynomials in the backward shift operator. The notation ARMAX refers to the model with A autoregressive terms, C moving average terms and B exogenous inputs terms as in Eq. (4):

$$A(q) = 1 + a_1 q^{-1} + \ldots + a_{n_a} q^{-n_a}$$
$$B(q) = b_1 + b_2 q^{-1} + \ldots + b_{n_b} q^{-n_b+1} \qquad (4)$$
$$C(q) = 1 + c_1 q^{-1} + \ldots + c_{n_c} q^{-n_c}$$

Equation (3) may be identified from input-output data (y_k, u_k) using system identification methods [27-33]. Equation (1) is sufficiently general to include FIR, ARX, ARMAX, and Box-Jenkins models. Like models identified using sub-space methods, Eq. (3) can be realized as a state space model in an innovation form [34]. There are several ways and codes in MATLAB to convert to state space or other forms. The Eq. (3) may be realized as a linear time invariant state space model in innovation form as in Eq. (5):

$$x_{k+1} = Ax_k + Bu_k$$
$$y_k = Cx_k \qquad (5)$$

In order to estimate the car following model with an ARMAX model, a dataset of car following behavior that presented in the preview subsection is used. To estimate the model, the system identification tool of MATLAB is used [35]. The accelerations a_{LV} and a_{FV} represent the accelerations of the LV and FV respectively and X_{FV} represents the traveled distance of FV. The two accelerations are used as the u_k and the position X_{FV} is used as y_k. There are several criteria to obtain the model that describes the car following behavior the best. The best model is the simplest model that accurately describes the dynamics. To compare models and choose the model with the best performance, the normalized root mean square error for overall best fit criterion has been considered.

Almost half of the extracted car following data was used for model identification stage. Several models were designed and the results obtained were compared with real dataset. After considering the influence of models coefficients on the performance of the behavior and an extensive review on the models behavior, the most appropriate model was selected. The identified model is a car following behavior model. It means that the model can simulate the behavior of an unknown human driver that is performing a car following behavior in the traffic flow. The obtained model does not have any parameter which needs to be tuned. The model overall best fit criterion is 99.59 and despite the driver parameters such as age, gender, driving condition and etc., the model simulates driver behavior very well. So it can be said that the model is a generalized model for car following behavior. This model was evaluated using the remaining half of the car following data. The comparisons of the output of the model with real data for one vehicle which had a car following behavior for the entire period of the maneuver are shown in figure (4).

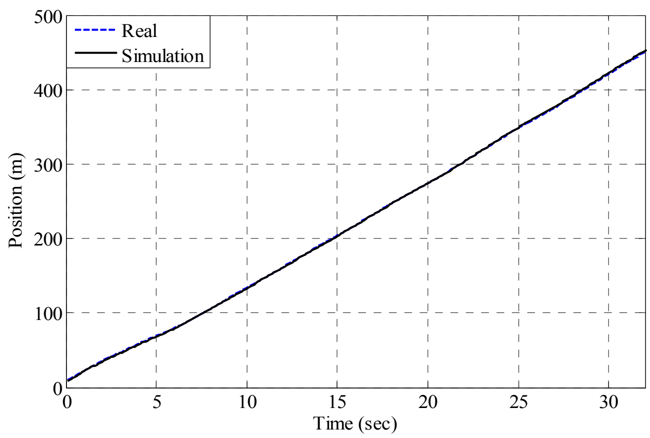


Figure 4 Comparison between the real data and the output of the FV model.

This result shows that the output of the proposed model and real data for a vehicle with a car following behavior are very similar. Here, the output of only one car following behavior is shown. In order to have a better understanding of the performance of this model, in figure (5) the error between the output of the model and real data is shown.

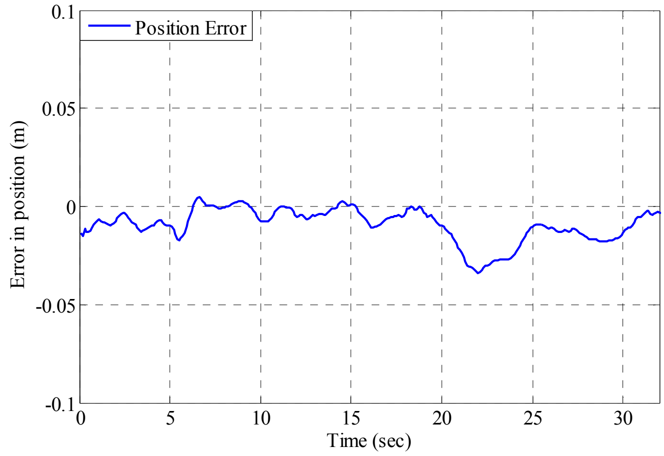


Figure 5 Error between the real data and the simulation results for ARMAX model

This result shows that the simulated output of the model has high accordance with the real data and the model simulated the behavior of the human driver very well.

To examine the performance of the obtained model with other data, various criteria were used to calculate errors. The criterion mean absolute percentage error (MAPE), according to Eq. (6), shows the mean absolute error that can be considered as a criterion to model risk to use it in real-world conditions. Root mean squares error (RMSE), according to Eq. (7), is a criterion to compare error dimension in various models. Standard deviation error (SDE), according to Eq. (8), indicates the persistent error even after calibration of the model. In these equations, x_i shows the real value of the variable being modeled (observed data), $\hat{x}$ denotes the real value of variable modeled by the model and N is the number of test observations [36]. Errors of the obtained car following model using several data considering these error criteria are summarized in Table 1 for the randomly selected vehicles.

$$MAPE = \frac{100}{N}\sum_{i=1}^{N}\frac{|x_i - \hat{x}_i|}{x_i} \tag{6}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \hat{x}_i)^2} \tag{7}$$

$$SDE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{|x_i - \hat{x}_i|}{x_i} - \frac{MAPE}{100}\right)^2} \tag{3}$$

Table 1 Result of error in the ARMAX car following model

Test Vehicle	MAPE	RMSE	SDE
Data1	0.3803	2.9393	0.0045
Data2	0.1969	0.8724	0.0011
Data3	0.4140	2.9644	0.0042
Data4	0.1435	0.5932	0.0014
Mean	0.2836	1.8423	0.0028

As it can be seen from Table (1), the car following model has low error values. The results show that the proposed model has a high compatibility with real car following behavior data.

3 Model predictive control system design

In this section, the design of the MPC system will be described. At first, a brief review on MPC is explained. Then, the MPC will be designed.

3.1 Model Predictive Control Basics

The MPC is the only advanced control methodology which has made a significant impact on industrial control engineering, the reason cited most often for the success of predictive control in the process industries is that the most profitable operation is often obtained when a process is running at a constraint, or even at more than one constraint. Often these constraints are associated with direct cost, frequently energy cost [8].

The general design objective of model predictive control is to compute a trajectory of a future manipulated variable u to optimize the future behavior of the plant output y. The optimization is performed within a limited time window by giving plant information at the start of the time window [37]. We will now briefly summarize the main ideas behind MPC. Figure (6) shows a schematic representation of MPC.

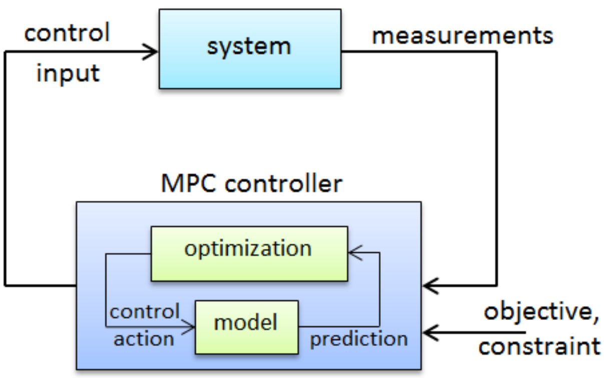


Figure 6 Schematic representation of MPC [9]

The MPC is based on (on-line) optimization and uses an explicit prediction model to obtain the optimal actions for the control measures. Let T_c be the control sampling interval, i.e., the (constant) time interval between two updates of the control signal settings. At each time step k (corresponding to the time instant $t = kT_c$), the controller measures or determines the current state $x(k)$ of the system and uses a model of the system to predict the behavior of the system over an interval $[k, k+N_p]$, where N_p is called the prediction horizon, as shown in figure (7). Next, the controller solves an open-loop optimal control problem to determine the optimal control inputs $u(k),\dots,u(k+N_p-1)$ that minimize a performance criterion $j(k)$ over the prediction period $[k, k+N_p]$, subject to the operational constraints. To reduce the computational complexity of the problem, one often introduces a constraint of the form $u(k+j) = u(k+j-1)$ for $j = N_c,\dots,N_p-1$, where $N_c(<N_p)$ is called the control horizon. When the optimal values are found by the controller, the control actions are applied in a

receding horizon fashion. This is done by applying only the first control sample $u(k)$ of the optimal control sequence to the system. Next, the prediction horizon is shifted one step forward and the prediction and optimization procedure over the shifted horizon are repeated using new system measurements. The receding horizon approach introduces a feedback mechanism, which allows reducing the effects of possible disturbances and mismatching errors between the actual output and the predicted output [9].

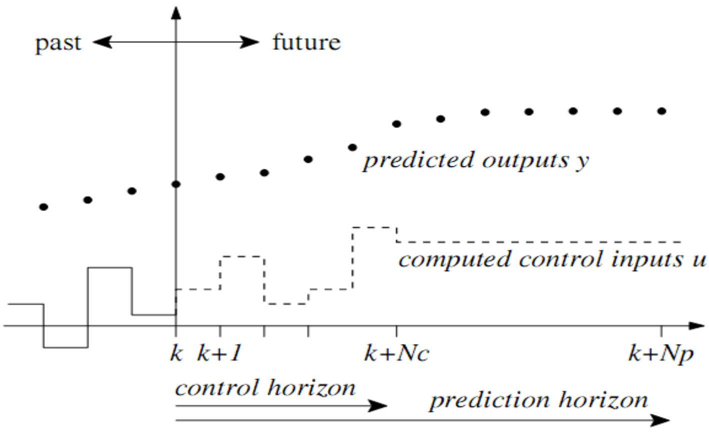


Figure 7 Representation of the MPC control scheme [9]

4.1 Model Predictive Control System Design

The MPC refers to a class of algorithms that compute a sequence of manipulated variable adjustment in order to optimize the future behavior of a plant. In this subsection, MPC is designed. One of the important parameters which have a significant role in tuning the MPC is weighting matrices which are defined in cost function. The cost function used in MPC is given by the scalar function according to Eq. (7).

$$J(k)=\sum_{i=1}^{N_P}(y(k+i\,|\,k)-r(k+i\,|\,k))^T \mathrm{Q}\,(y(k+i\,|\,k)-r(k+i\,|\,k))+\Delta u(k+i-1)^T \mathrm{R}\Delta u(k+i-1) \\ +u(k+i-1)^T \mathrm{P}\,u(k+i-1)+\rho_\varepsilon \varepsilon^2 \tag{7}$$

The cost function can be written in the matrix form. The main point of doing this is to remove the summation sign from the problem and the solution. This will simplify the discussion considerably and the problem of solving the control law is transformed to an on-line quadratic program (QP) problem, which can be solved using a mathematical programming solver [38]. Also solving the control law must satisfy the following constraint Eq. (8).

$$\begin{aligned} u_{\min}(i)-\varepsilon V^{u}_{\min}(i) &\le u(k+i\,|\,k)\le u_{\max}(i)+\varepsilon V^{u}_{\max}(i) \\ \Delta u_{\min}(i)-\varepsilon V^{\Delta u}_{\min}(i) &\le \Delta u(k+i\,|\,k)\le \Delta u_{\max}(i)+\varepsilon V^{\Delta u}_{\max}(i) \\ y_{\min}(i)-\varepsilon V^{y}_{\min}(i) &\le y(k+i\,|\,k)\le y_{\max}(i)+\varepsilon V^{y}_{\max}(i) \end{aligned} \tag{8}$$

The constraints on u, Δu, and y are relaxed by introducing the slack variable $\varepsilon \ge 0$. The weight ρ_ε on the slack variable ε penalizes the violation of the constraints. The larger ρ_ε with respect to input and output weights, the more the constraint violation is penalized. The equal concern for the relaxation vectors $V^*_{\min}, V^*_{\max}$ have nonnegative entries which represent the concern for relaxing the corresponding constraint; the larger V, the softer the constraint. V=0 means that the constraint is a hard one that cannot be violated.

Here, Q, R and P are symmetric and positive semi-definite weighting matrices specified by the designer. The more general choice is to specify Q, P and R as diagonal weighting matrices. Often, P is chosen as zero in order to obtain MPC with offset-free control, i.e., $y = r$ in steady state. The smaller the weighting matrices, the less important is the behavior of the corresponding variable to the overall cost function. The problem of choosing the weighting matrices are usually process dependent and must usually be chosen by trial and error. However, if Q, P and R are chosen as diagonal positive definite matrices and if the prediction horizon N_p is large (infinite) then it can be proved that the closed loop system with the optimal control is guaranteed to be stable even if the open loop system is unstable.

The goal of our MPC is to keep the future relative distance in a safe region by computing the appropriate acceleration of the FV. The Pipes law [39] defines this safe region as stated in Eq. (9).

$$S = L\left(1 + \frac{V_{FV}}{4.47}\right) \tag{9}$$

This distance is the safe distance that must be kept between FV and LV. As shown in figure 8, the value of the Pipe's law is applied to the controller as a dynamic reference. This value changes according to the instantaneous value of velocity in each instant and it is not a constant reference for the controller. Hence, the controller tries to control the acceleration of FV in order to keep the relative distance in the safe region. Also, the acceleration of LV is fed to the controller as a disturbance, therefore any changes in the behavior of the LV is reflected to the controller.

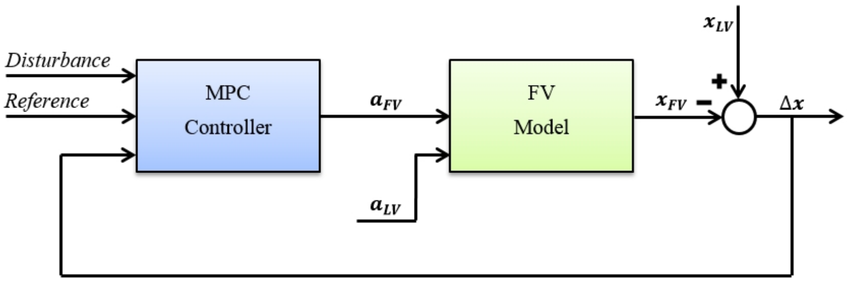


Figure 8 Schematic of the control system with MPC.

Here, since the system output (i.e. maintaining the safe distance) is more important that its input (i.e. produced acceleration from the controller), the weighting matrices are defined as mentioned in Eq. (10).

$$\begin{aligned} &\mathrm{Q} = \text{Diagonal Unit Matrix} \\ &\mathrm{R} = 0.1I \\ &\mathrm{P} = 0 \end{aligned} \tag{10}$$

By setting the weighting matrix R as mentioned in Eq. (10), the input increments $\{\Delta u(k\,|\,k),\ldots\Delta u(m-1+k\,|\,k)\}$ get more important than its input. This fact doesn't indicate that the acceleration generated by the controller does not have importance; but instead it is expressing that acceleration is controlled with another parameter, i.e. constraint, which has a better impact on its performance.

The constraint on the acceleration and its increments are defined as soft constraint as in Eq. (11).

$$\begin{aligned} 2-\mathit{0.5} &\le u_i(k) \le 2+\mathit{0.5} \\ 0.5-\mathit{0.1} &\le \dot{u}_i(k) \le 0.5+\mathit{0.1} \end{aligned} \tag{11}$$

The slack variable is displayed in *italic*. Since MPC requires the solution of an optimization problem at each time step, the feasibility of that problem must be ensured. If the on-line optimization problem is not feasible, then some constraints would have to be weakened. Finding which constraints to weaken is an extremely difficult process to do, since it is an nN_p-hard problem. A possible (and partial) remedy to the problem is to consider constraint softening variables ε on process output constraints (as shown in Eq. (11)) and include a penalty term such as ε^2 in the cost function. Feasibility, in addition to being a practical consideration, is also important for closed-loop stability of MPC [8, 40].

The performance of the designed MPC is evaluated in the next section.

4 Results and discussion

In this section, the performance of the designed controller will be investigated. To assess this performance, a dataset of real traffic is required. The dataset used here was introduced in section 2.

The performance of the designed controller will be investigated through simulation. In the available dataset, the data of hundreds of vehicles which had the car follow behavior are provided. In this paper, the simulation results of only one vehicle from the data set are presented. However, the simulation was done for many other test vehicles too, but it is impossible to provide all of them in one paper.

For the simulation of the randomly chosen test FV vehicle, the necessary data for this vehicle and its preceding vehicle (LV) was extracted from the dataset. The initial conditions (V_{FV} and X_{FV}) for the FV model was set similar to that of the real driver behavior, therefore the conditions at the start of the simulation for both the model and the driver are the same.

The behavior of the FV was simulated for a period of 41 sec. The sampling time (T_c) of the simulation was set to 0.1 sec. For the prediction horizon (N_p) and the control horizon (N_c) which were explained in section 3, 20 sec and 4 sec were selected respectively. The performance of the designed controller was evaluated through investigation of several variables. Then, the results of the simulation were compared with the data of the test vehicle that was used.

One of the variables of the system was the output of the controller as the acceleration of the FV. In order to limit the variations of acceleration, a numerical constraint was applied to its rate of changes (as mentioned in Eq. (11)). It was an essential constraint since the variations of acceleration could get so high. This constraint was applied to ensure the comfort of passengers and prevent the sudden movements of the vehicle. Figure (9) shows the comparison between the result of the simulation and that of the real vehicle.

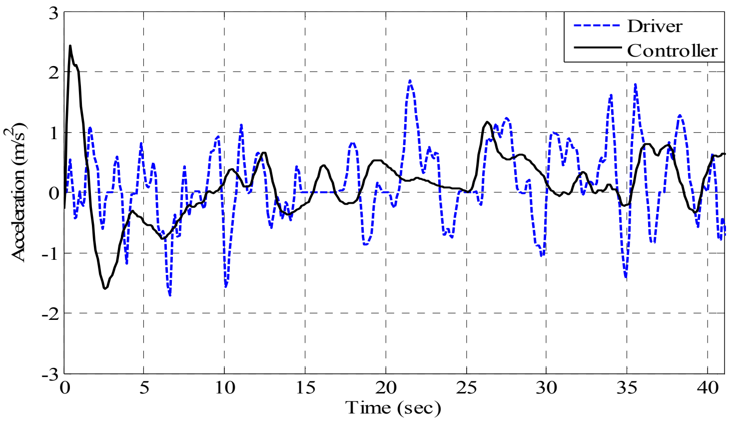
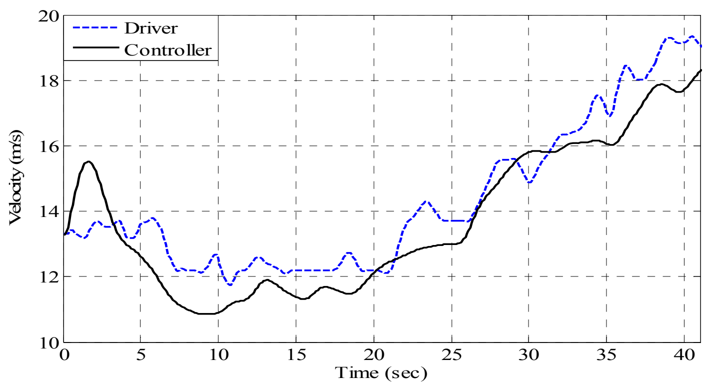


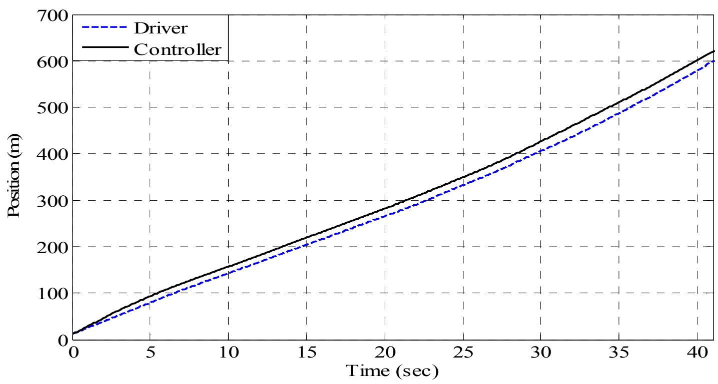
Figure 9 Comparison between the acceleration of the simulation and FV's real behavior.

The comparison shows that the controller's output follows that pattern of the real behavior to a satisfactory extent and indeed it has a smoother rate of changes. Fewer variations in the same period of time assure the comfort of vehicle and passengers.

Other variables that were obtained from the simulation were the velocity and position of FV. These results are shown in figure (10).



(a)



(b)

Figure 10 Comparison between the results of simulation and FV's real behavior, **(a)** Velocity, **(b)** Position

In figure (10a), the comparison shows that the velocity of the controller also has a smoother rate of changes and fewer variations in the same period of time. These factors result in lower consumption of energy and steadier travel of the vehicle which also guarantee the comfort of passengers. About figure (10b), it can be seen that the position resulted from simulation almost has a similar pattern with the position of the real vehicle. But the position of the vehicle is closer to LV in comparison with the position of real driver. However, though it is closer to LV, but it is keeping a safe distance with its preceding vehicle. This result can be concluded from figure (10) too. Being in a closer position, in comparison with real position, leads to smaller length of the platoon of vehicles in the traffic flow. In addition, figure (11) shows the relative distance between the FV and LV.

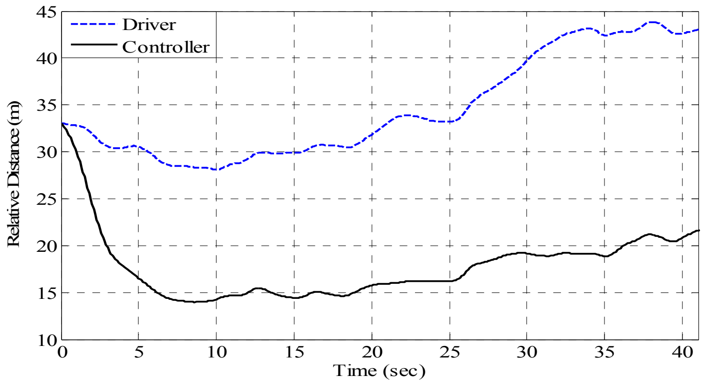


Figure 11 Comparison of the relative distance between the result of MPC and real driver

The results show that the real driver kept a very long unnecessary distance with its LV. But the controller kept a sufficient distance and avoided the unnecessary gap. It also confirms that the controller decreases the length of the platoon of vehicles in the traffic flow.

As stated in section 3, the relative distance must be kept in a safe region which is computed through Pipe's law. In other words, Pipe's law presents the safe distance value of each instant. This law acts like a constraint on relative distance to ensure the safety of the vehicles. In order to evaluate the performance of the MPC, the relative distance from the controller and the driver is compared with the value of Pipe's law for the entire behavior. This result is shown in figure (12) as the error between the safe and relative distance.

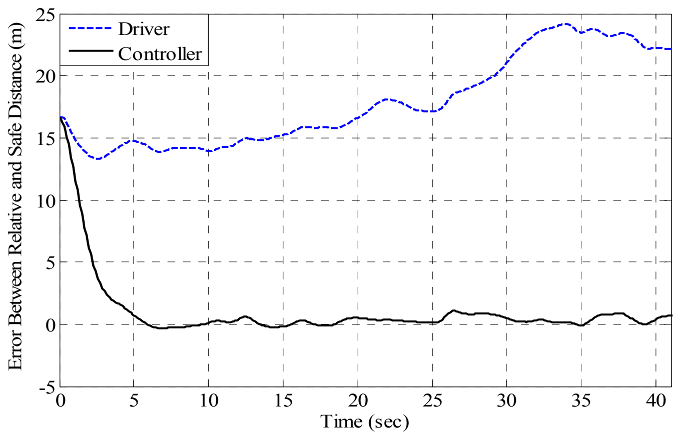


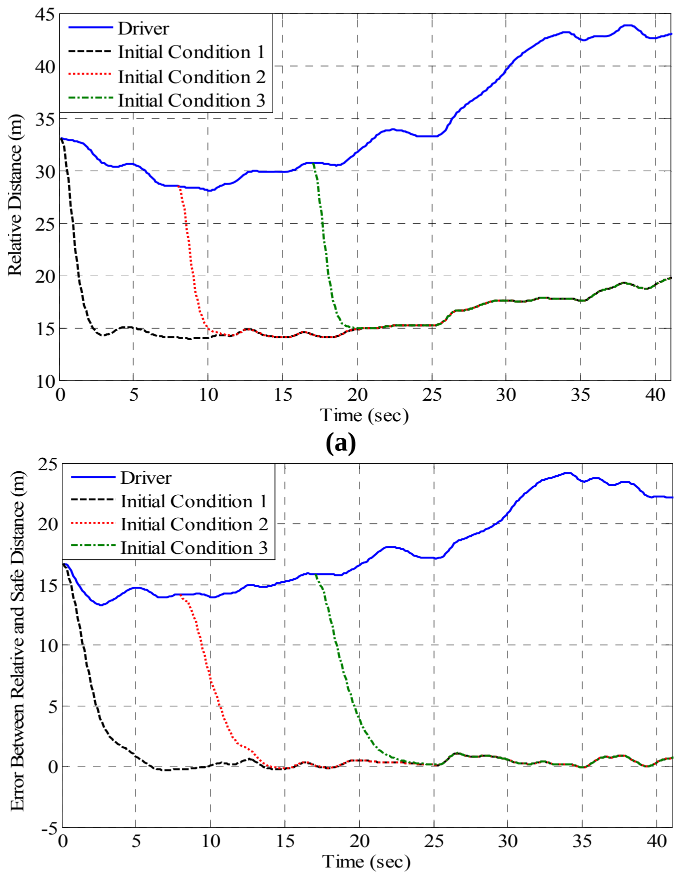
Figure 12 Comparison of the relative distance between the result of MPC and real driver

As it can be seen from the figure, the behavior of the real driver has a significant difference with Pipe's law. But the result of the controller has such a low error (around zero). Notice that at the start point, the error of the controller is similar with the error of real driver. It is due to the fact that the initial conditions are set similar and it takes a while for the controller to reduce the error.

To conclude, it can be said that the MPC controller has safer behavior than that of a human driver. It's due to the fact that MPC keeps the safe distance with its preceding vehicle, i.e., the FV keeps a safe distance with its preceding vehicle. Also the MPC produces the acceleration and velocity with have much fewer variations than the human driver. As a result, the FV travels with a steadier motion which leads to less fuel consumption. The human driver kept longer distance with its preceding vehicle than the distance which Pipe's law suggested according to its velocity. This unnecessary long distance will increase the traffic queue and leads to more driving risks for other vehicles in the traffic.

To evaluate the robustness of the controller, in different time intervals of the entire period of the behavior, the controller is applied to the model. It is expected that no matter when the controller was applied to the system, it must be able to keep the relative distance in the safe region, and keep the error around zero. So, different initial conditions for the controller will be selected.

For a vehicle chosen randomly from the dataset, the 1st initial condition was selected at the moment of the start of the maneuver. The two other initial conditions were randomly selected at seconds 8 and 17 of the entire behavior. The simulation result of the controller for this vehicle is shown in figure (13).



(a)

(b)

Figure 13 Comparison between the real driver behavior and result of MPC for different initial conditions, **(a)** Relative Distance, **(b)** Error

As shown in figure (13a), no matter when the controller was applied, it is able to keep the relative distance in the safe region, and as shown in figure (13b), it is also able to keep the error around zero. So, different initial conditions do not disturb the performance of the controller. In other words, the designed MPC controller is perfectly compatible to the changes of initial conditions.

In the next step, the stability of the MPC will be investigated. Two broad classes of stability definitions are associated with (a) stability with respect to initial conditions and (b) input-output stability, respectively. The two classes are complementary to each other and can also be combined. From the result of figure (13), it can be said that the controller has stability with respect to the initial conditions. The concept of stability is fundamental in the study of dynamical systems. Loosely speaking, stability is a dynamical system's property related to "good" long-run behavior of that system. While stability by itself may not necessarily guarantee satisfactory performance of a dynamical system, it is not conceivable that a dynamical system may perform well without being stable. There are several methods to perform stability test. First we examine our controller's stability through 'REVIEW' code in MATLAB. REVIEW checks for the potential design issues in the Model Predictive Controller design and generates a report. Review performs the following diagnostic tests:

-Is the optimization problem to be solved online well defined?

-Is the controller internally stable?

-Is the closed loop system stable when no constraints are active and there is no model mismatch?

-Is the controller able to eliminate steady-state tracking error when no constraints are active?

-Is there a likelihood that constraint definitions will result in an ill-conditioned or infeasible optimization problem?

The result of this diagnostic is showed in table (2).

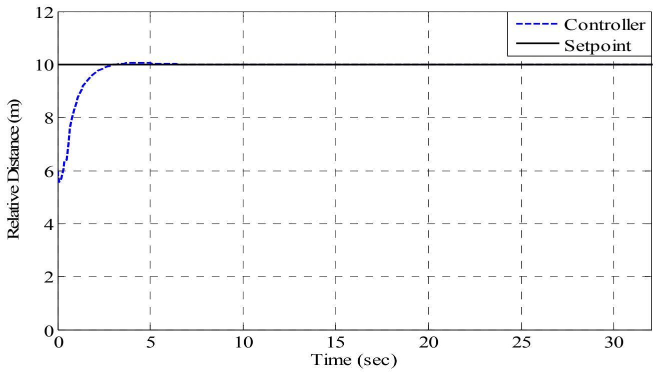
Table 2 Summary of Performed Tests

Test	Status
MPC Object Creation	Pass
QP Hessian Matrix Validity	Pass
Controller Internal Stability	Pass
Closed-Loop Nominal Stability	Pass
Closed-Loop Steady-State Gains	Pass
Hard MV Constraint	Pass
Other Hard Constraints	Pass
Soft Constraint	Pass

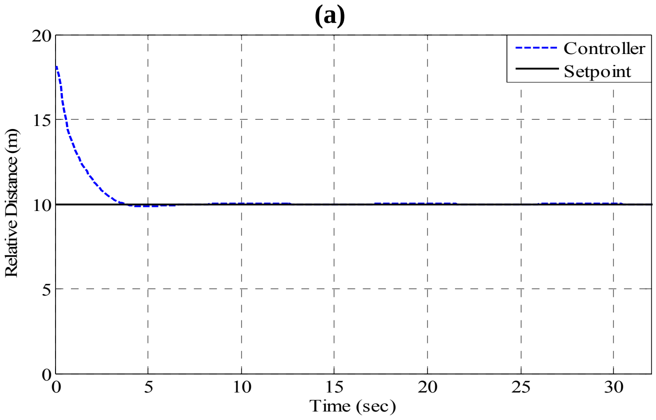
As the table shows, the controller passes all the diagnosis of tests. Therefore, it can be concluded that the MPC is designed correctly.

In MATLAB, the MPC block receives the current measured output signal (Δx), reference signal (*Ref*), and measured disturbance signal (a_{LV}). The block computes the optimal manipulated variables (a_{FV}) by solving a quadratic program (QP). The Quadratic Problem (QP) Hessian matrix validity shows that our choice of cost function parameters and horizon is acceptable that results in positive definite QP Hessian matrix. Passing internal stability of a controller shows that the eigenvalues of the MPC controller is designed properly. The closed-loop steady-state shows that the controller drives the output variables to their targets at steady state. The diagnoses about constraint indicate that there isn't a potential hard-constraint conflict with MV (a_{FV}) bounds or its rate of changes, and there is proper balance between hard and soft constraint. If the constraint is designed too hard, the controller might pay too much attention. Moreover, making a constraint harder cannot prevent a violation if the constraint is fundamentally infeasible. The Controller internal stability and closed-loop stability along with other test, all pass the diagnosis. Of course these tests cannot detect all the possible performance factors. So, additionally we will test our design using techniques such as simulations. Impulse-response or step-response is simple but practical tests that can provide result about stability [41]. Knowing the step response of a dynamical system gives information on the stability of such a system, and on its ability to reach one stationary state when starting from another [42].

Therefore, as the next step, a constant set-point is applied to the controller to investigate how well the controller is able to track the input with the existence of disturbance in the system. Instead of applying a normal step test, a constant input is applied to the control system to simulate a situation as if the LV is moving with a constant velocity and it is expected that FV keeps the relative distance with its LV in a constant value. In this situation, a relative distance of 10 meters is chosen and this input is applied to two different initial conditions. In the first case, the FV is too close to the LV and in the second case; the FV is too far from it.



(a)



(b)

Figure 14 Result of the response test, **(a)** FV close to LV, **(b)** FV far from LV

As it is show in figure (14a) and (b), in both cases, the Controller can easily track the input without any overshot. The result is exactly the one expected from the MPC. In the first case, the FV was too close to the LV and the controller lead the distance to the value of the safe distance that has been designated. In the second case, the FV kept an unnecessary distance from the LV and the controller leads it to the safe distance that has been designated.

To investigate the robustness of the MPC, the impulse-response or step-response are used which provide a practical description in many applications, as they can be easily determined from experimental tests, and allow a reasonably simple way to compute robust predictions. Impulse and step response descriptions are only equivalent when there is no uncertainty. If there is uncertainty, they behave rather differently [43].

In the next step, a robustness check was performed by testing our controller's sensitivity to prediction errors. To do such test, a second model was designed similar to the one explained in section II, but some small coefficient in its state were interfered in a way that the behavior of the second model have some perturbation and behave differently to the same input. It was named the perturbed model, and it act as if the vehicle is in an unbalanced car following condition. Then, a step input containing noise was applied to the controller to both the accurate model and the perturbed model. Both models used the same controller. First, the step is applied to the system and in the middle of signal, on a random time, an impulse is also inducted on the step. The result of such test is shown in figure (15).

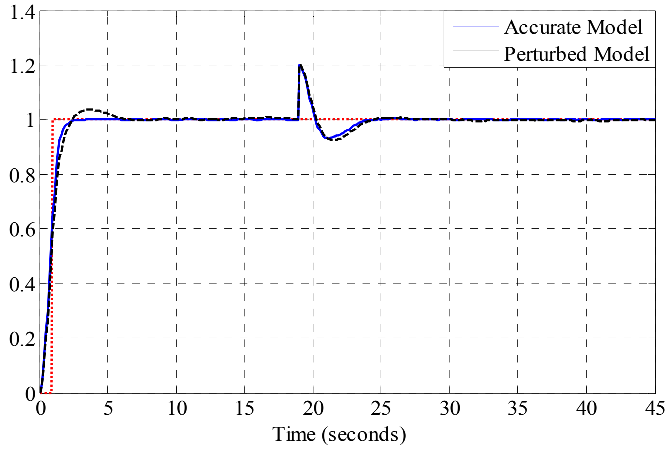


Figure 15 Robustness test

As it can be seen, the controller guides the accurate model and the perturbed model to the set-point defined for the system and it performs well despite the fact that the second model has a perturbation in the design. In the second part of the test, when the impulse influences on the set-point, the controller quickly return both the accurate and the perturbed model to the set-point. The accurate model reaches the set-point smoothly and without overshot, that was expected because the controller has been designed to do such a behavior. Injecting such an impulse to the system is like the situation that the FV suddenly accelerates or the LV suddenly brakes and the FV recedes from its safe distance target. In this situation the controller returns it to set-point as well. The perturbed model has an overshot because the perturbed model was designed in a way to show instability in the conditions of the FV. As a result, the controller's performance is robust against noise and changes in model behaviors. It can be said that another advantage of the controller is its durability to cope with instantaneous changes of the disturbance which in this case is the changes of the acceleration of the LV.

5 Conclusion

In this study, at first, an ARMAX model for car following behavior is designed based on real traffic data. This model is employed as the plant of the control system. The control of car following behavior is achieved through model predictive control approach. Based on the relative distance and relative acceleration of each instant, the MPC predicts the future behavior of the LV and according to this behavior, the acceleration of the FV is controlled. The MPC tries to control this acceleration in a way to keep the relative distance at a safe region. The important aspect of this type of control is MPC's ability to deal with constraints on controls. For this behavior, safety acts like a constraint on the relative distance between FV and LV. It's due to the fact that the value for safe distance is obtained at each time instant from Pipe's law and this value varies according to the velocity of FV at each time interval and the controller must try to predict it and then keep up with it during the entire period of the maneuver. Also, a numerical constraint is applied to the rate of changes of acceleration to ensure the smooth movement of the vehicle. To investigate the performance of the designed controller, the result of the system is compared with the behavior of real drivers with similar initial conditions. The simulation results show that the MPC controller has a behavior much safer than that of the real driver and it provides a pleasant trip for passengers. Fewer variations of acceleration and velocity in the same period of time assure the comfort of vehicle and passengers. Also, the controller results lead to lower consumption of energy and steadier travel. In addition, by keeping the FV in a proper distance with its LV, the controller decreases the length of the platoon of vehicles in the traffic flow. In addition for more evaluation, controller stability and robustness against noise were also studied. This proposed model can be applied in future studies related to car following behavior, since it can help to study different features of driver's behavior at maintaining safe distance. The presented control system can be used to improve the current control system's performances in driver assistant devices, safe distance keeping observers, collision prevention systems and etc.

Acknowledgment

The authors extend their thanks to US Federal Highway Administration and Next Generation Simulation (NGSIM) for providing the data set used in this paper.

References

[1] Rothery, R. W., "Car Following Models: In Traffic Flow Theory", Transportation Research Board, Vol. 165, pp. (4)1-(4)41, (1992).

[2] Zhang, H. M., and Kim, T., "A Car-following Theory for Multiphase Vehicular Traffic Flow", Transportation Research Part B: Methodological, Vol. 39, No. 5, pp. 385–399 (2005).

[3] Mar, J., and Hung-Ta, L., "The Car-following and Lane-changing Collision Prevention System Based on the Cascaded Fuzzy Inference System", IEEE Transactions on Vehicular Technology, Vol. 54, No. 3, pp. 910-924, (2005).

[4] Pan, D., and Zheng, Y., "Optimal Control and Discrete Time-delay Model of Car Following", IEEE 7th World Congress on Intelligent Control and Automation, Chongqing, China, pp. 5657-5661, (2008).

[5] Peng, Y. F., Hsu, C. F., Chih-Min, L., and Kao, C. J., "Robust CMAC Back Stepping Longitudinal Control of Vehicle Platoons", IEEE International Conference, Networking, Sensing and Control, Taipei, Taiwan, Vol. 1, pp. 571-576, (2004).

[6] Zakian, V., "*Control Systems Design: A New Framework*", Springer-Verlag, Berlin, (2005).

[7] Limon, D., Alvarado, I., Alamo, T., and Camacho, E.F., "MPC for Tracking Piecewise Constant References for Constrained Linear Systems", Automatica, Vol. 44, pp. 2382–2387, (2008).

[8] Maciejowski, J. M., "*Predictive Control with Constraints*", Prentice Hall, Harlow, England, (2002).

[9] Baskar, L. D., Schutter, B. D., and Hellendoorn, H., "Model-based Predictive Traffic Control for Intelligent Vehicles: Dynamic Velocity Limits and Dynamic Lane Allocation", IEEE Intelligent Vehicles Symposium, Eindhoven, Netherlands, pp. 174–179, (2008).

[10] Bellemans, T., De Schutter, B., and De Moor, B., "Model Predictive Control for Ramp Metering of Motorway Traffic: A Case Study", Control Engineering Practice, Vol. 14, No. 7, pp. 757–767, (2006).

[11] Hegyi, A., De Schutter, B., and Hellendoorn, J., "Optimal Coordination of Variable Speed Limits to Suppress Shock Waves", IEEE Transactions on Intelligent Transportation Systems, Vol. 6, No. 1, pp. 102–112, (2005).

[12] Kamal, M. A. S., Mukai, M., Murata, J. and Kawabe, T., "On Board Eco-driving System for Varying Road-traffic Environments using Model Predictive Control", IEEE International Conference on Control Applications (CCA), Yokohama, Japan, pp. 1636-1641, (2010).

[13] Naus, G. J. L., Ploeg, J., Van de Molengraft, M. J. G., Heemelsand, W.P.M.H., and Steinbuch, M., "A Model Predictive Control Approach to Design a Parameterized Adaptive Cruise Control", Automotive Model Predictive Control, Vol. 402, pp. 273-284, Springer-Verlag, Berlin Heidelberg, (2010).

[14] Salehinia, S., "Predictive Control System Design for Car Following Behavior in Traffic Flow", M.Sc. Thesis, Department of Mechanical Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran, (2012).

[15] Kvapil, D., "Stability Analysis of State-space Models in MATLAB", Journal of Applied Mathematics, Vol. 3, No. 2, pp. 189-198, (2010).

[16] Tettamanti, T., Varga, I., Kulcsár, B. and Bokor, J., "Model Predictive Control in Urban Traffic Network Management", IEEE 16th Mediterranean Conference on Control and Automation Congress Centre, Ajaccio, France, pp. 1538–1543, June, (2008).

[17] Bemporad, A., and Morari, M., "Robust Model Predictive Control: A Survey", Automatic Control Laboratory, Swiss Federal Institute of Technology (ETH), Springer, Switzerland, (2007).

[18] Brookhuis, K. A., de Vries, G., and de Waard, D., "The Effects of Mobile Telephoning on Driving Performance", Elsevier Journal of Accident Analysis and Prevention, Vol. 23, No. 4, pp. 309–316, (1991).

[19] Caird, J.K., Willness, C.R., Steel, P., and Scialfa, C., "A Meta-analysis of the Effects of Cell Phones on Driver Performance", Elsevier Journal of Accident Analysis and Prevention, Vol. 40, No. 4, pp. 1282-1293, (2008).

[20] Terwen, S., Back, M., and Krebs, V., "Predictive Powertrain Control for Heavy Duty Truck", In: Proceedings of the IFAC Symposium on Advances in Automotive Control, Salerno, Italy, Vol. 39, pp. 105-110, (2004).

[21] Falcone, P., Borelli, F., Asgari, J., He, T., and Hrovat, D., "A Model Predictive Control Approach for Combined Braking and Steering in Autonomous Vehicles", In: Proceedings of the IEEE Conference on Control and Automation, Athens, Greece, pp. 1-6, (2007).

[22] Re, L., Ortner, P., and Alberer, D., "Chances and Challenges in Automotive Predictive Control", Automotive Model Predictive Control Models, Methods and Applications, Vol. 402, pp. 1-22, (2010).

[23] US Department of Transportation, NGSIM – Next Generation Simulation, http://www.ngsim.fhwa.dot.gov (2009).

[24] The Federal Highway Administration Website: US 101 Highway (Hollywood Freeway), Dataset:http://www.fhwa.dot.gov/publications/research/operations/07030/index.cfm.

[25] Khodayari, A., Ghaffari, A., Kazemi, R., and Braunsting, R., "A Modified Car-following Model Based on a Neural Network Model of the Human Driver Effects", IEEE Transactions on Systems, Man, and Cybernetics, Part A-Systems and Humans, Vol. 42, No. 6, pp. 1440–1449, (2011).

[26] Thiemann, C., Treiber, M., and Kesting, A., "Estimating Acceleration and Lane-changing Dynamics Based on NGSIM Trajectory Data", Transportation Research Record: Journal of the Transportation Research Board, Vol. 2088, pp. 90-101, (2008).

[27] Box, G. P., Jenkins, G. M., and Reinsel, G. C., "*Time Series Analysis: Forecasting and Control*", 3rd ed. Prentice-Hall (1994).

[28] Hamilton, J. D., "*Time Series Analysis*", Princeton University Press, (1994).

[29] Katayama, T., "*Subspace Methods for System Identification*", Springer-Verlag, Berlin, (2005).

[30] Ljung, L., "*System Identification. Theory for the User*", 2nd ed. Prentice Hall, New York, (1999).

[31] Soderstrom, T. and Stoica, P., "*System Identification*", Prentice-Hall, New York, (1989).

[32] Overschee, P., Van and De Moor, B., "Subspace Identification for Linear Systems. Theory–implementation–applications", Kluwer Academic Publishers, (1996).

[33] Zhu, Y., "*Multivariable System Identification for Process Control*", Pergamon, London, (2001).

[34] Jørgensen, J. B., Huusom, J. K., and Rawlings, J. B., "Finite Horizon MPC for Systems in Innovation Form", 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC),Orlando, FL, USA, December 12-15, (2011).

[35] Ljung, L., "System Identification ToolboxTM 8 User's Guide", The MathWorks, Inc. (2012).

[36] Cohen, E. R., "An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements", Measurement Science and Technology, Vol. 9, No. 6, pp. 839-1026, June, (1998).

[37] Wang, L., "*Model Predictive Control System Design and Implementation using MATLAB*", Springer, Australia, (2009).

[38] Bemporad, A., Morari, M., Dua, and Pistikopoulos, V., "The Explicit Linear Quadratic Regulator for Constrained Systems", Automatica, Vol. 38, No. 1, pp. 3-20, (2002).

[39] Pipes, L. A., "An Operational Analysis of Traffic Dynamics", Journal of Applied Physics, Vol. 24, No. 3, pp. 274–281, (1953).

[40] Nikolaou, M., "Model Predictive Controllers: A Critical Synthesis of Theory and Industrial Needs", Elsevier Advances in Chemical Engineering, Vol. 1, No. 713, pp. 131–204, (2001).

[41] Palumbo, G., and Pennisi, S., "*Feedback Amplifiers-Theory and Design*", Springer, New York, (2003).

[42] Ogata, K., "*Modern Control Engineering*", Prentice Hall, New York, (2010).

[43] Garcia, C.E., Prett, D.M., and Morari, M., "Model Predictive Control: Theory and Practice-A Survey", Automatica, Vol. 25, No. 3, pp. 335–348, (1989).

Nomenclature

a	: Vehicle acceleration
$A(q^{-1}), B(q^{-1}), C(q^{-1})$	: polynomials in the backward shift operator
e	: Noise Disturbance
FV	: Follower Vehicle
J	: Performance criterion
k	: Time step
L	: Vehicle length
LV	: Leader Vehicle
$MAPE$	: Mean absolute percentage error
N	: Number of test observations
N_c	: Control horizon
N_p	: Prediction horizon
P, Q, R	: Weighting Matrix
r	: Reference
$RMSE$	: Root mean squares error
S	: Safe distance
SDE	: Standard deviation error
T_c	: Control sampling interval
u	: Input

V	: Vehicle velocity
x_i	: Real value of the variable being modeled (observed data)
$\hat{x}$	: Real value of variable modeled by the model
$\overline{x}$	: Real mean value of the variable
X	: Vehicle running distance
y	: Output

Greek Symbols

Δu	: Input increments
Δx	: Longitudinal distance Between FV and LV
ε	: Slack variable
ρ_{ε}	: Weighting Matrix

چکیده

باتوجه به افزایش روز افزون تردد وسایل نقلیه، معیار ایمنی و کنترل حجم ترافیکی از سوی مجامع علمی و تحقیقاتی بسیار مورد توجه قرار گرفته است. از اینرو برای کاهش تصادفات، خطاهای رانندگی و همچنین بهبود جریان ترافیکی فن‌آوری وسایل نقلیه با استفاده از روش‌های مدلسازی و سیستم‌های کنترلی رو به هوشمند شدن پیش می‌رود. هدف اصلی این پژوهش طراحی سیستم هوشمند کنترل طولی خودرو برای رفتار تعقیب خودرو می‌باشد. در این پژوهش برای کنترل کردن حرکت طولی خودرو، ابتدا به مدل‌سازی فرآیند تعقیب خودرو بر مبنای مدل ARMAX پرداخته می‌شود. برای ساخت این مدل از داده‌ها واقعی رفتار تعقیب خودرو استفاده خواهد شد. برای این امر، پارامترهای مورد نیاز از مجموعه داده استخراج شده است و مدلسازی بر مبنای شناسایی مدل صورت گرفته است. سپس عملکرد خروجی مدل به دست آمده با داده‌های واقعی مجموعه داده توسط معیارهای مختلف خطا سنجیده می‌شود. نتایج نشان می‌دهند که مدل به دست آمده با کمترین خطا به خوبی رفتار تعقیب خودرو را شبیه‌سازی می‌کند. در گام بعدی این پژوهش، با استفاده از تئوری کنترل پیش‌بین مدل اقدام به طراحی کنترل‌کننده پیش‌بین برای مدل ارائه شده می‌شود. این سیستم کنترلی علاوه بر حفظ فاصله ایمن سعی در فرآهم کردن لذت رانندگی را نیز دارد. مقایسه نتایج شبیه‌سازی سیستم کنترلی طراحی شده با مقادیر واقعی نشان می‌دهند که حفظ فاصله ایمن توسط سیستم کنترلی به خوبی انجام شده است. این سیستم کنترلی با تولید سیگنال بهینه شتاب، خودرو تعقیب‌گر را در منطقه ایمن نگه‌داشته و از تمامی فضای ایمن موجود استفاده کرده است. این امر منجر به نزدیک‌تر شدن خودرو تعقیب‌گر به خودرو راهنما شده است که نتیجه آن کاهش صف ترافیکی در پشت خودرو می‌باشد. همچنین شتاب و سرعت تولید شده توسط سیستم کنترلی نرخ تغیرات کمتری نسبت به شتاب و سرعت تولیدی توسط راننده انسانی دارد، که این امر منجر به کاهش مصرف سوخت و حرکت یکنواخت‌تر خودرو و در نتیجه لذت بردن سرنشینان از سفر می‌شود. علاوه بر این برای ارزیابی بیشتر، پایداری سیستم کنترلی و مقاوم بودن آن در برابر اغتشاشات نیز مورد مطالعه قرار گرفت. نتایج این ارزیابی نشان می‌دهند که سیستم کنترلی در مقابل تغییرات ناگهانی در سیگنال‌های ورودی مقاوم بوده و در حفظ فاصله ایمن در شرایط گوناگون بصورت پایدار رفتار می‌کند.