OPEN ACCESS Freely available online

Advances in Automobile Engineering

Research Article

Predictive Model Based Low-Speed Adaptive Cruise Control for Autonomous Vehicles

Orhan Alankus, Elif Toy Aziziaghdam, Kaan Çakin*

Department of Automotive Mechatronics and Intelligent Vehicles Istanbul Okan University, Istanbul, Turkey

ABSTRACT

European Union confirmed the "Vision Zero" objective in June 2019, as to achieve zero deaths and serious injuries by 2050. This can only be attained through connected and autonomous vehicles integrated into intelligent transport systems and a sustainable mobility system. This requires a cost-effective, fast,and efficient development process for advanced connected and autonomous vehicle functions. In this article, a methodology to develop low-speed Adaptive Cruise Control (ACC), which is one of the most important functions of an autonomous vehicle, is explained. Vehicle tracking at slow speeds is a problem especially for conventional vehicles with high levels of nonlinearities in the powertrain system. As a part of a university-industry collaboration project "SAE level 3 autonomous bus development", a flexible and realistic discrete plant model including longitudinal vehicle and powertrain model has been developed and discrete low-speed ACC is designed. The plant model aims to perform detailed and realistic software tests of autonomous features, which interfaces with the vehicle controllers. OKAN_UTAS autocorrected-multi-parameter longitudinal model is integrated. For engine modeling via shaft dynamometer the 3D map of the engine is reproduced. The transmission characteristics were prepared through the road tests. To increase the reliability of the developed functions, Software in the Loop (SIL) and Model in the Loop (MIL) simulations were conducted before the on-road vehicle tests. Finally, C code with the MISRA C standard of ACC is generated and embedded into a real-time platform. The plant model, ACC design, and Model in the Loop test results are presented.

Keywords: Autonomous vehicle; Adaptive cruise control; Vehicle longitudinal dynamic model; Model in the loop; Software in the loop

INTRODUCTION

In June 2019, the European Union confirmed the "Vision Zero" objectives and they set the goal to reduce fatal accidents between 2020 and 2030 by half and zero fatal accidents in 2050 [1]. With similar goals in the World, the importance of smart vehicle systems and autonomous vehicles has increased. Studies on autonomous vehicles have been intensified to make the use of urban and intercity traffic safer, more efficient, and comfortable.

Six levels of autonomous driving classified by the Society of Automotive Engineers (SAE) in 2017 [2]. When SAE Level 3 and Level 4 vehicle characteristics are examined, it is expected that vehicles can be driven by autonomous systems. Adaptive Cruise Control (ACC) systems of the vehicles with minimum SAE Level 3 characteristics should be capable of driving autonomously at low speeds on suitable roads in the traffic [2]. According to European Road Transport Research Advisory Council (ERTRAC), automated shuttles and automated buses will be used on dedicated roads in 2020-2024. Those vehicles will be observed and used on mixed traffic roads in 2024-2030 [3]. Than finally fully automated urban vehicles will be used after 2030 on all types of urban vehicle roads. According to the McKinsey report, in 2030, 15% of the vehicles to be sold are autonomous vehicles [4].

Adaptive Cruise Control (ACC) is a system that is one of the core technology for autonomous vehicles. The system called an extended version of Cruise Control (CC) [5,6]. CC can only keep the velocity which is limited by the driver. But ACC can control brake or throttle actions according to different situations of the traffic. Vehicles that have the ACC system should be equipped with camera and radar (sometimes LIDAR) to sense front objects. The main purpose of the ACC system is keeping the safe distance between ego (which is equipped with ACC) and lead vehicle according to the velocity that is set by the driver. ACC system design is standardized by International Organization for Standardization

Correspondence to: Kaan Cakin, Department of Automotive Mechatronics and Intelligent Vehicles Istanbul Okan University, Tuzla, Istanbul, Turkey, E-mail: kaan.cakin@hotmail.com

Received: May 16, 2020; **Accepted:** May 25, 2020; **Published:** June 12, 2020

Citation: Alankus O, Aziziaghdam ET, Çakin K (2020) Predictive Model Based Low-Speed Adaptive Cruise Control for Autonomous Vehicles. Adv Automob Eng 9:194. doi: 10.35248/2167-7670.2020.9.194

(ISO) [7]. Although ACC is called a comfort system, it also has safety functions. For instance, the ACC system is separated into two parts Collison Avoidance and Automated Highway Systems, and benefits of longitudinal automation are explained in [5] and two different falsification methods are developed using Rapidly-exploring random trees to avoid rear-end collisions [8].

For designing a proper ACC system, there are various studies in the literature. Different researchers use different methods to design the ACC system. Also, different methods for control theory have been applied to the ACC systems like optimized proportional-integral, gears and line approximation, gears and tangent approximation, basic tangent approximation, basic gain-scheduling approximation, on-line and off-line PWA MPC, Nonlinear MPC and they are compared [9]. Also, the adaptive PI method and off-line methods are found efficient in the presence of on-board computational intensity. These methods can also be used in real-life applications, even if a detailed database browser is needed.

A vehicle equipped with ACC can adjust its speed to the front vehicle speed via control gas and brake actions. The early version of the ACC systems working range was started with 40 km/hr and they could control the velocity until 200 km/hr. Also, the braking force is limited at 0.25 g and 1.0 g for maximum braking by authority [9]. In the studies of [10], sampling time, vehicle spacing policy, and an ACC system control algorithm are discussed. The most well-known spacing policy is described as a constant-time gap policy. But this method found not efficient for traffic flow stability. The polynomial root locus is used to plot a time-based sampling time stability system and pole locations are observed. Under State-state condition, a first-order modeled vehicle used for an ACC system design and three different controller methods (PID, sliding mode, constant-time-gap) is used [11]. Results were observed and these methods were not found appropriate for critical transitional maneuvers (sudden acceleration-deceleration). Then an ACC system designed for two-vehicle situations with Model Predictive Controller (MPC). ACC system is tested for three different vehicle masses and results were compared. In the studies of [6] a Nonlinear MPC controller is used for ACC which switches the system automatically to CC and ACC.In the studies of [12] data were collected and evaluated from seven different ACC equipped vehicles to explain the most appropriate car following system in traffic use.

Nowadays, the result of the intensive traffic congestion in big cities, ACC is evolved. The system that adjusts the velocity of the vehicle according to safety distance in low-speed situations called ACC Stop and Go. An ACC with Stop and Go ability is designed that tries to keep the vehicle in a safe distance and set velocity according to different road sxurfaces called Advanced Smart Cruise Control (ASCC) [13]. The paper aimedto calculate more confidential safety distance according to different friction coefficients of the road and keeping the vehicle in a safe distance and set velocity. Simulation is done for four different friction coefficients by the Carsim simulation program. With the goal of fuel-saving, a control switching method which switches the controller between PID, adaptive PID, and fuzzy PID is designed [14]. In Svaji and Sailaja'sstudies, two PID controllers (hybrid PID) are used for velocity and distance control via the ACC system that has stopped and go ability according to comfort factors [15].

For designing flexible systems that can be used by different types of vehicles in real road conditions, a realistic longitudinal dynamic model of the vehicle should be prepared. In literature, ACC systems are modeled by using a basic longitudinal vehicle dynamic model. But for reliable systems and for control parameters choosing a realistic vehicle model is needed. Several studies about vehicle longitudinal and lateral dynamic models are avaliable in the literature [5-7,16,17].

In this study, a flexible and realistic discrete plant model including a longitudinal vehicle model and powertrain model has been developed to perform detailed and realistic software tests of autonomous vehicles and intelligent systems, which interfere to the vehicle controllers. Model-based programming is used for the preparation of the model. The model is verified for the bus with Euro5, 4.5 L Diesel Engine, 4-speed automatic transmission with torque converter and electronic brake system. The data collected from the bus conducted via Vehicle CANBus and the collected data and delays on each subcomponent are analyzed. For engine modeling, the bus was connected to the shaft dynamometer and 3D map of the engine is reproduced. The 2D transmission map and torque converter characteristics were prepared through the road tests. Also, OKAN_UTAS autocorrected-multiparameter model [18] is used for self-updating and preparing a flexible longitudinal vehicle model for simulating different types of vehicles intelligent system developments.

REALISTIC AND FLEXIBLE VEHICLE MODEL

Generally, in literature when designing ACC, a basic vehicle dynamic model is used. Powertrain loses and delays are not considered. But if the closest safe distance (shorter than normal gaps) between two vehicles wanted to follow in the real road and reliable simulations of algorithms are desired, a detailed vehicle longitudinal dynamic model should be modeled. In this thesis, a realistic longitudinal vehicle dynamic model which is classified into two categories is designed.

- The dynamics of powertrain: Engine, transmission, torque converter
- The dynamics of vehicle body: Calculations of external forces on the vehicle that includes aerodynamic drag force, gravitational force in addition to rolling resistance and traction force

Powertrain model

A vehicle powertrain model generally contains engine model, transmission model, and torque converter model (if the vehicle has) [5]. Studies were done on the software base via Matlab/Simulink. 3D Engine map is reproduced via shaft dynamometer which is connected to the bus and transmission maps are composed of data gathering from vehicle's Controller Area Network (CAN). Finally, the algorithms are validated with real road tests.

As seen from Figure 1, there are two separate model boxes for transmission and engine. Transmission block produces current gear, selected gear, gear ratio when inputs are real vehicle speed and real throttle percentage. The selected gear is the number o/f gear for the most proper position of transmission for vehicle speed at that time. However, because of the process delays, the real gear position may be in a different step. To calculate this, current gear is observed. Current gear is the gear number that is defined by the transmission at the moment. To determine these outputs,

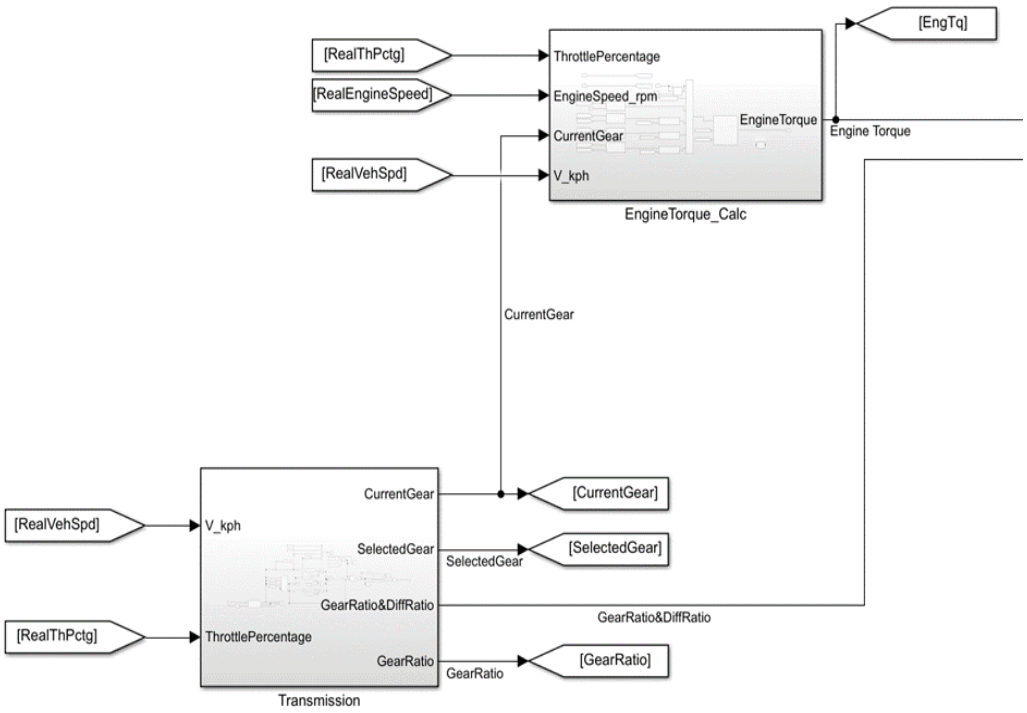

Figure 1: Powertrain model simulink desing.

real road test and data gathering are done. Data is collected via DSpaceMicroAutobox II from CAN port of dedicated bus when a professional test driver was driving the vehicle. Vehicle speed, acceleration, transmission ratio, engine speed, outputshaft speed, engine torque and gear information are collected for each throttle position of 0%, 10%, 20%, 30%, 40%, 50%, 70%, 100%. The map is created on two tables as up-shift and down-shift than integrated to Simulink algorithms. Finally, current gear is generated according to instant vehicle speed and throttle pedal percentage.

According to engine torque calculations, four different shift map is designed for four different gear (the test vehicle has four-stage gear). First of all, the bus is connected to shaft dynometer. The speed of the output shaft is fixed by the shaft dyno, the gear and throttle percentage were sent through CANBus. For determined gear, engine speed, and throttle percentage values, engine torque information is collected from the vehicle (CAN). This data creates input information of the map. According to current gear, throttle percentage, and engine speed, these maps give the result as engine torque. Different from the others, first gear shift map inputs are velocity and throttle percentage. Although first gear has a specific ratio, it has torque converter which gives different ratios for different velocities according to torque converter applications. Therefore, the map is designed for different velocities and throttle percentages at first gear which is represented in Figure 2.

Calculations of powertrain variables are recognized via calculation patterns that is included [5] and also [6] studies are considered.

One of the elements of the powertrain is torque converter consists of three main parts, the impeller, the turbine, and the reactor. The impeller is connected to the crankshaft that transmits the power of the engine to the turbine by the hydraulic oil inside the torque converter. The relationship between engine torque and impeller torque is defined in equation (1).

$$I_{ei}N_e=T_e(u_t,N_e)-T_i \quad (1)$$

N_e is the engine speed (rpm)

I_{et} is the engine and impeller moment of inertia

T_i is the impeller torque (Nm) and it is equal to converter input torque

T_e is the engine torque (Nm) as a function of engine speed (N_e(rpm) and percentage throttle position (u_t)

For proper matching, the engine and converter should have the same capacity factor. And also assume that the velocity of impeller equals to the engine speed. The equation can be written like (2) [19].

$$\mathrm{Ti}=\left(\frac{N_e}{K_{tc}}\right)^2 \quad (2)$$

If speed ratio, torque ratio and capacity of the torque converter are known, outputs of the torque converter can be calculated, $T_t=C_{tr}\times T_i$ and $N_t=C_{sr}\times N_i$. The speed ratio is, $C_{sr}=N_t/N_i$ torque ratio is $C_{tr}=T_t/T_i$, Tt is the turbine torque, N_t is the turbine angular velocity.

While the torque converter is engaged and transmission gear ratio is known (R), outputs of the gearbox are calculated from $T_{out}=R\times T_{in}$ and $N_{out}=N_{in}/R$ equations. Transmission input torque is turbine torque T_t and the torque is transmitted to the wheel. It gives wheel torques T_w. The wheels' torque can be calculated from $T_w=(1/R)$ T_t at steady-state condition. Therefore, the relation between the transmission speed (N_{tr}) and wheel speed (N_w)will be calculated with $N_{tr}=(1/R)N_w$.

The engine dynamic is constitutively represented according to losses with equation (3).

$$I_{ei}N_e=T_i-T_f-T_\alpha-T_p \quad (3)$$

T_i is the engine combustion torque that is the torque produced by the engine without any loses. $T_f$isthe torque frictional losses, T_a isthe accessory torque and they cause a total loss on the engine. T_p

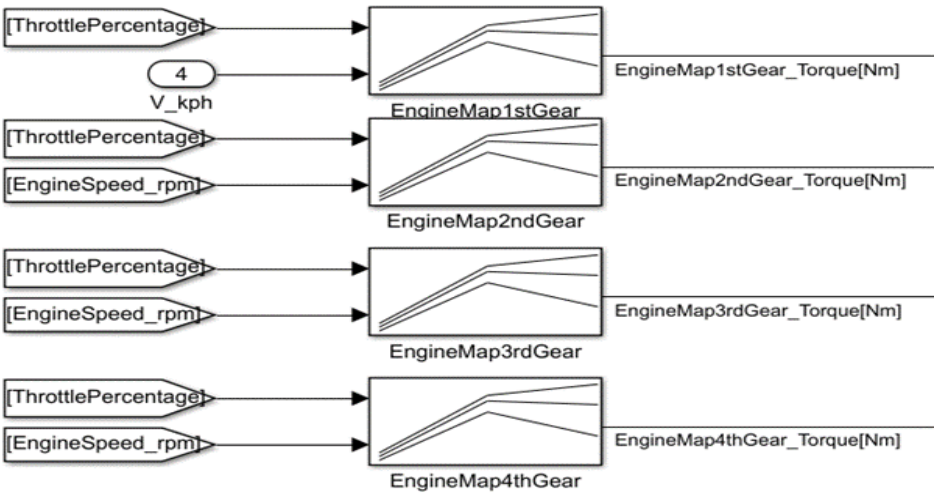

Figure 2: Powertrain model simulink desing.

is the pump torque coming from the torque converter and it can be calculated with $T_p=C_{cf}(C_{sr})W_p^2$. C_{cf} is the capacity factor and w_p is the angular velocity of the pump (rpm).Therefore, engine torque can be defined as, $T_e=T_i T_f T_\alpha$.

A basic equation for all inertias related to rotating parts are described as using equivalent inertia.

$$I_{eq}=I_e I_d^2 I_t^2 I_d I_d^2+I_t \quad (4)$$

I_e, I_d, I_t are respectively the moment of inertias of the engine, differential, and transmission. I_t and I_d are transmission and differential ratios. Than net torque at the engine can be found by,

$$T_{enet}=T_e I_d I_t \eta_t \eta_d \quad (5)$$

Although there are various equations for engine modeling, due to combustion non-linearity these equations could not provide satisfactory accuracy. Hence in this study a data-driven model for the engine is prepared. η_t and η_d are the transmission and differential efficiencies [19]. All these formulas were added to the model-based design that is designed via Matlab/Simulink to form a realistic longitudinal model of the vehicle.To present the model of the engine there are engine maps for different motors that show the relationship between engine torque (N/m), the engine rotation speed (rpm), and the percentage of throttle pedal (%). In this project, the vehicle was connected to the shaft dynamometer and 3D map of the engine is reproduced.

Brake model design

The simple model that feed the braking torque to the system as a function of braking action is used according to brake algorithm design. The braking torque is obtained by finding out the amount of pressure produced behind the brake disk P_{bi} while applying the brake pedal. u_{bi} is the position of the pedal which changes between 0 and 100 [19].

$$P_{bi}=1.5K_{ci}u_{bi}\tau P_{bi} \quad (6)$$

τ is the lumped lag obtained by combining two lags coming from the servo valve and the hydraulic system. K_c is pressure gain. i define the front and rear tire values. It changes f and r according to the front tire and rear tire calculations. To add the model to Simulink, the function is converted to transfer function with Laplace transform.

$$P_{bi}(S)=\frac{1.5Kciubi}{(1+\tau s)} \quad (7)$$

According to load transfer, the brake torque on the front and rear tire will be different. The difference will be determined by different for K_{bi} each axle and also it depends on the wheel velocity.

$$T_{bi}=P_{bi}K_{bi}min(1,N_{wi}/0.001) \quad (8)$$

Vehicle body dynamics model

The general force equation is calculated from Newton's second law from Figure 3.

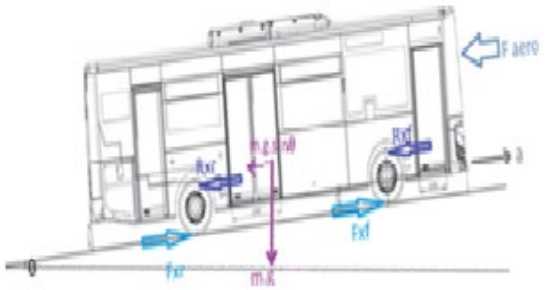

Figure 3: Powertrainmodel simulinkdesing.

$$mmx=F_{xf}+F_{xr}-R_{xf}R_{xr}-F_{aero}-mg\ sin\theta \quad (9)$$

$\ddot{x}$ is acceleration and m is a mass of the vehicle. θ is the angle coming from the slope of the road. R_{xf} and R_{xr} are rolling resistances on front and rear tires. F_{aero} represents aerodynamic resistance force and they are calculated by (10).

$$F_{aero}=0.5\rho C_d A_\alpha (v_x+v_{wind})^2,\ R_{xf}+R_{xr}=f(F_{zf}+F_{zr}) \quad (10)$$

ρ is the air density, C_d is the aerodynamic drag coefficient and A_α is the frontal area of the vehicle. v_x and v_{wind} are the velocities of vehicle and wind respectively. Wind velocity may have negative or positive sign according to direction of the wind. F_{zf}, F_{zr} are the normal forces of front and rear wheels and they can be calculated the equation that is obtained by taking moment.

$$F_{zf}=\frac{F_{aero}h_{aero}-m\ddot{x}h-mghsin\theta+mgl_r cos\theta}{l_f+l_r},\ F_{zr}=\frac{F_{aero}h_{aero}+m\ddot{x}h+mghsin\theta+mgl_f cos\theta}{l_f+l_r} \quad (11)$$

VEHICLE MODEL PARAMETER PREDICTION METHOD

For controlling dynamic systems, it may not possible to measure the required data instantly, for each period. Different vehicle and road conditions are not directly measurable.For this reason, it is provided to converge the estimated values to the desired data with the Extended Kalman filter. As the input of the filter, the angular velocity with noise is used to set up the equation. Also, as the simplest method to discriminate prediction equations an advanced Euler method approach is used [18].

$$W_k = W_{k-1} + \frac{(F_{x(k-1)} R_W \pm T_b)}{J} T_S \quad (12)$$

$$V_{ego\_k} = V_{ego\_k-1} + \frac{F_{x(k-1)}}{m} T_S \quad (13)$$

$$\sigma_k = V_{ego\_k-1} + \frac{F_{x(k-1)}}{m} T_S \quad (14)$$

$$\gamma_k = a_{ego}(1 - e^{-b\sigma k-1} - c\sigma_k \quad (15)$$

w is the angular velocity of ego vehicle's tire. Radius of the wheel symbolized with R_w. T_b is the brake torque. T_s is the sample time. σ is slip rate. γ is road friction coefficient. Equations mentioned above (12-15) in discrete time as a result of the estimated input and status vector are represented in equation (16).

$$g = w and\ h = \begin{bmatrix} W_k \\ V_{ego\_k} \\ \sigma_k \\ m_k \\ \gamma_k \end{bmatrix} \quad (16)$$

Kalman Filter is a form of predictor-corrector commonly used in control systems engineering to predict unmeasured states of a process. For extended Kalman Filter two-step predictor-corrector algorithm is used. As part of the algorithm, two Jacobians equations are used and represented below.

$$F_k = \begin{bmatrix} 1 & 0 & 0 & \frac{T_s R_g m_k}{J} & \frac{T_s R_g u_k}{J} \\ 0 & 1 & 0 & -T_s g & 0 \\ \frac{-R_w}{V_{ego\_k}} & \frac{R_w w}{(V_{ego\_k})^2} & 0 & 0 & 0 \\ 0 & 0 & a_{ego}(be^{-b\sigma k} - c) & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, H_h = [1 \quad 0 \quad 0 \quad 0 \quad 0] \quad (17)$$

ADAPTIVE CRUISE CONTROL DESIGN

The purpose of the ACC system is tracking a set velocity and a safe distance from the lead vehicle by adjusting longitudinal acceleration.

The ACC controller tries to keep the ego vehicle at a safe distance. For that purpose, it gives acceleration and brake commands to vehicle according to adjusting the proper relative distance, which is represented diagrammatically in Figure 4 . The main logic is represented below.

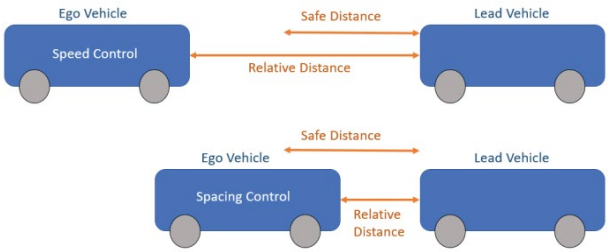

Figure 4: Control strategies of ACC system.

If Relative Distance ≥ Safe Distance, speed control mode is active. So the controller tries to keep velocity at a set velocity.

If Relative Distance ≤ Safe Distance, the distance control mode is active. So the controller tries to keep the distance at safe distance.

Safe distance can be described basically,

$$d_{safe} = d_{defoult} + t_{gap} \times v_{ego} \quad (18)$$

d_{safe} is safe distance. $d_{defoult}$ is default distance at standstill between ego and lead vehicle. t_{gap} is the time gap between vehicles and v_{ego} is velocity of ego vehicle. According to provide comfortable and safe driving, the ACC system is separated in two parts [20]. First part is vehicle independent part which controls desired distance between vehicles and velocity of ego vehicle with Model Predictive Controller (MPC). Radar information (relative velocity and distance) is an input for vehicle independent part. Second one is vehicle dependent part that is used for giving proper gas-brake commands according to keeping desired distance and velocity, (Figure 5).

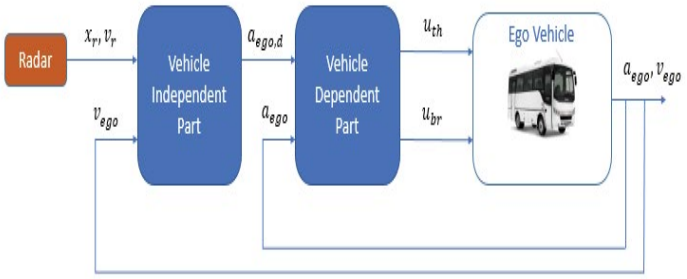

Figure 5: Twocontrol parts of ACC system.

The information about relative distance and velocity is obtained by radar and transferred to vehicle independent part. This part calculates and determines desired acceleration (deceleration for braking situation) of ego vehicle. This information is the input data with ego vehicle acceleration to vehicle dependent part. The vehicle dependent part tries to keep desired acceleration [7] by using throttle and brake actions. u_{th} and u_{br} are throttle and brake signals. Finally, ego vehicle produces desired acceleration and sends current acceleration and velocity information to ACC system in continuous loop.

Vehicle independent part

ACC system is described as a comfort system in literature. But beside this ability it has also safety and reducing fuel consumption goals [20]. According to meet this aim, the system designed with predictive future. Vehicle independent part determines the desired acceleration (deceleration for braking) to follow lead vehicle in safe, comfortable and efficient time gap. In this study, vehicle independent part designed with Model Predictive Control. Because

of the computational load the control model needs to be simple but functional. The previous maneuver must be performed by the ACC system until the next MPC calculation is created, which will create the next acceleration/deceleration order [21]. The ACC system decides speed control or spacing control according to the data coming from the sensors.

The linear model of the controlling structure for MPC is represented below.

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + Ez(k) \\ y(k+1) &= Cx(k) + \mathrm{F} \end{aligned} \tag{19}$$

$$A=\begin{bmatrix} 1 & 0 & T_s & -0.5T_s^2 & 0 \\ 0 & 1 & 0 & T_s & 0 \\ 0 & 0 & 1 & -T_s & 0 \\ 0 & 0 & 0 & 1 - T_s/t_d & 0 \\ 0 & 0 & 0 & -1/t_d & 0 \end{bmatrix}, B=\begin{bmatrix} 0 & 0 & T_s/t_d & 1/t_d \end{bmatrix}^T, C=\begin{bmatrix} 1 & -t_{gap} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$E=[0.5T_s^2\ 0\ T_s\ 0\ 0]T$, $F=[-d_{defoult}\ 0\ 0\ 0]$

$X(k)=d_r\ v_{ego}\ v_r\ a_{ego}(a_{ego}(k)\text{-}a_{ego}(k\text{-}1))/T_s$

$y(k)=[\Delta d\ v_r\ a_{ego}(a_{ego}(k)\text{-}\ a_{ego}(k\text{-}1))/T_s]$,

t_{gap} and t_s are the time differences between two vehicles and the sample time respectively. t_d is the time constant for dependent part controller. d_r is the relative distance and v_r is the relative speed between vehicles. The distance error and velocity error between ego and lead vehicle are determined as $\Delta d=d_r\text{-}d_{safe}$ and $\Delta v=v_r\text{-}v_d$ Accelerations of ego and lead vehicles are represented as aego and alead. The jerk of the ego vehicle can be calculated by $(a_{ego}(k)\text{-}a_{ego}(k\text{-}1))/T_s$.

Predictive model of MPC

One of the most useful features of MPC is its ability to predict future behavior. The prediction states are calculated through prediction horizon by future control inputs.

$$\begin{aligned} x_p(k+\rho) &= \tilde{A}x(k) + \tilde{B}U(k+\mu) + \tilde{E}Z(k+\rho) + K_x(x(k) - x_p(k)) \\ y_p(k+\rho) &= \tilde{C}x(k) + \tilde{D}U(k+\mu) + \tilde{G}W(k+\rho) - \tilde{F} + K_y(y(k) - y_p(k)) \end{aligned} \tag{20}$$

ρ and μ are the prediction horizon and control horizon. Output vectors are calculated at time stepk. $X_p(k+\rho)$ and $y_p(k+\rho)$ are the prediction state and output vectors. Control input vector is $U(k+\mu)$ and disturbance vector is $W(k+\rho)$. The disturbance vector are constant as long as predictive horizon is calculated for the first value, $z(k)$. The matrices in the equation (19) are expanded state matrices of the prediction model. They are obtained from state matrices by substitution of each. The factors for predict are corrected according to the prediction output and state errors in the previous time step.

Constrains and control strategy

Control strategy is constructed on solving the optimization problems into the cost function and obtain smooth control action. For this reason, cost criterion J is defined for minimizing each step of cost function.

$$\min J = \sum_{i=1}^{\rho} (y_p(k+i) - y_{ref}(k+i))^T Q (y_p(k+i) - y_{ref}(k+i)) + \sum_{i=1}^{\mu-1} u(k+i)^T W u(k+i)^T \tag{21}$$

Q and W are the weighting matrices. y_{ref} is the reference commands coming from the actions of lead vehicle.

For defining the system constrains, objective constrains and state constraints are obtained separately. The objective constrains are represented as, $d_r \geq d_{defoult}$. The system constraints are;

$$V_{ego,max} \geq v_{ego} \geq v_{ego,min}, \alpha_{ego,max} \geq \alpha_{ego} \geq \alpha_{ego,min}, j_{ego,max} \geq j_{ego} \geq j_{ego,min}.$$

The ACC system must ensure to keep the safe distance between vehicles by smooth and comfortable maneuvers. Therefore, the acceleration and jerk must be adjusted continuously and logically.

Generally, Quadratic Programming(QP) is used to define cost functions in each state and objective constrain. This enables real-time solving of each step.

Finally, min $J(u(\mu)\times(\rho))$ and $PU(\mu)\leq v$ can be obtained. P is the matrix that contains all system matrices and it is limited with ύ vector.

Vehicle dependent part

The desired vehicle acceleration calculated in Vehicle Independent Part is continuously compared with the actual acceleration of the vehicle. PI controller was used to bringing these two values closer together [21].

This part works as a Cruise Control system. It tries to continue at the set speed until an object or other vehicle appears in front of the vehicle. If an object or vehicle appears in front of it, the ACC system is active and efforts are made to ensure a safe distance. But unlike cruise control, braking is also available in the dependent model, byWhen a positive force request is received from the controller, the throttle pedal is intervened to ensure this. Motor torque is obtained by calculating the desired force with the formulas in chapter II. This resulting torque enters the reverse engine map and matches the appropriate engine speed and desired throttle percentage is obtained. using switch method. If the desired force is negative, the negative acceleration has been calculated and the brake controller of the bus has been transmitted, thereby decelerating occurs.

SIMULATION AND VALIDATION

In this study, the model was designed at discrete time in Matlab/Simulink and generated in C code compatible with MISRA C and integrated into the DSpaceMicroAutobox II real-time platform. The model simulations were performed in two steps. Firstly, the model outputs calculated for the tests of the realistic and flexible vehicle model were compared with the data collected from the CAN system of the real vehicle and the model was verified. The vehicle parameters which are further improved are integrated in the model. Later, ACC model in the loop tests were performed with the realistic vehicle model. As a result, ACC tests were performed at low and variable speeds, which is one of the most difficult conditions for ACC systems. All graphs are presented below Figure 6.

The accuracy of the model for different parameters are represented below in Table 1.

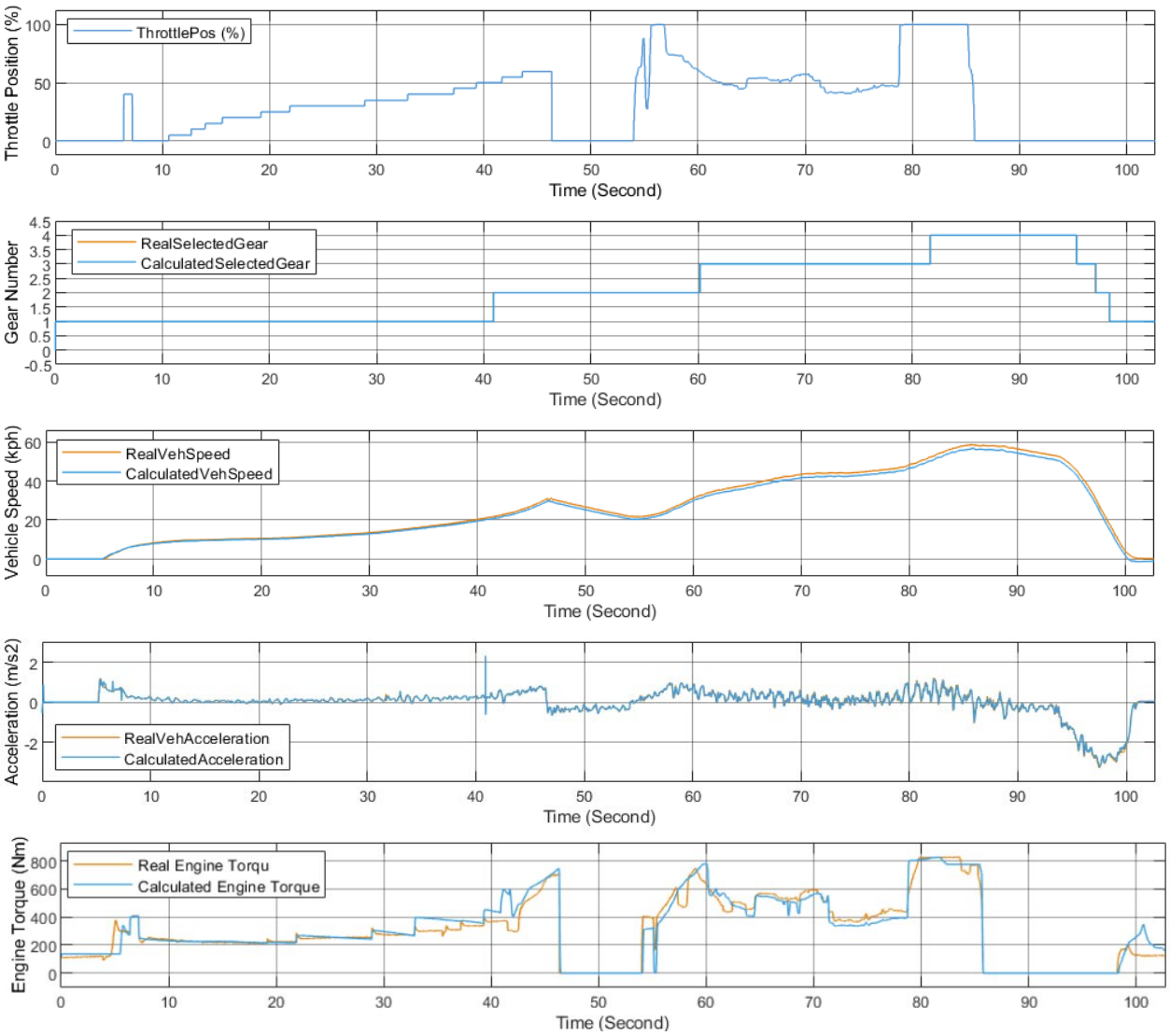

Figure 6: Graphs of throttle position (%), gear number and comparison charts of real-calculated vehicle speed (kph), real-calculated vehicle acceleration (m/s^2), real-calculated engine torque (Nm).

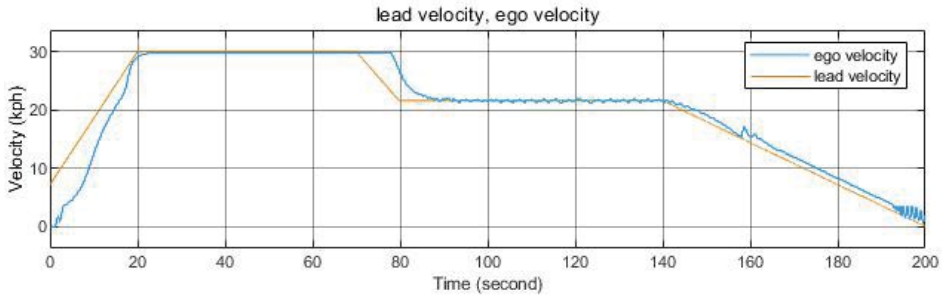

Figure 7: Comparison graph for lead vehicle velocity and ego vehicle velocity according to following with ACC.

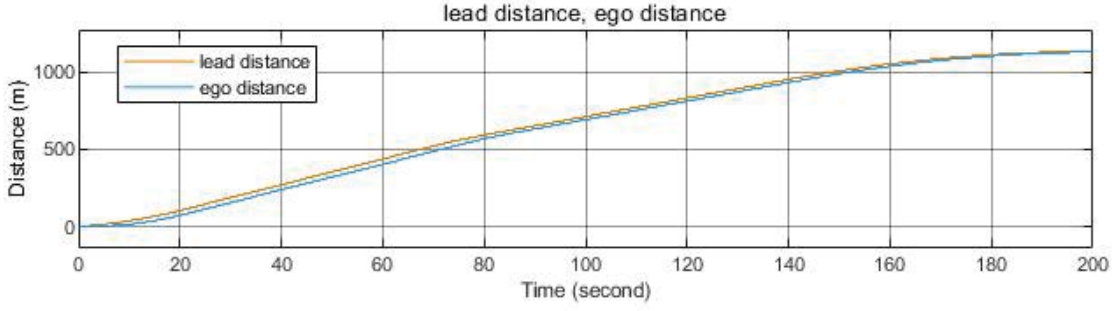

Figure 8: Comparison graph for distance between lead and ego vehicle according to following with ACC.

Table 1: The accuracy of the model (percentage).

	V. Acceleration	V. Speed	Shaft Speed	Engine Torque	Selected Gear
Accuracy %	84.2	95.7	96.6	83.3	99

The designed vehicle model is compared with real vehicle data and improvements are made. Realistic and flexible model is obtained. With this model, ACC system with low speed operation ability is tested and result graphs are represented below (Figures 7 and 8).

The speed of the ego vehicle is set to 30 kph. ACC adjusts the speed of the vehicle to 30 kph until it reaches a safe distance or a vehicle comes in front, then sets the distance difference between the two vehicles to 2 seconds.

CONCLUSION

In this study, a flexible and realistic discrete plant model including a longitudinal vehicle model and powertrain model has been developed to perform detailed and realistic software tests of autonomous vehicles and intelligent systems, which interfere to the vehicle controllers. Model-based programming is used for the preparation of the model. The model is verified for bus with Euro5 4.5L Diesel Engine, 4-speed automatic transmission with torque converter and electronic brake system. For engine modeling, the bus was connected to the shaft dynamometer and 3D map of the engine is reproduced.The 2D transmission maps and torque converter characteristics were prepared through the road tests. On the designed vehicle model, MIL and SIL tests of the MPC controlled ACC were performed with the vehicle plant model. The model was improved with parameter validations. In future studies, the ACC system tests will be carried out on the real vehicle. Also,a designed vehicle model can be used in platooning simulations of connected vehicles. Therefore, studies have been carried out for the design of SAE level 3 autonomous vehicles, and the infrastructure has been designed to test and simulation for intelligent vehicle systems.

ACKNOWLEDGEMENTS

The study about this paper is a part of the university-industry collaboration project between Istanbul Okan University-OTOKAR A.Ş. We would like to thank OTOKAR for providing equipment and know-how to us during our works, as well as equipment installation, infrastructure works, and providing all support.

We would also like to thank TÜBİTAK for providing financial support to this project (5170067) under the 1505 University-Industry Cooperation Support Program.

REFERENCES

1. Commission staff working document. EU Road Safety Policy Framework 2021-2030-Next steps towards "Vision Zero", 2019.
2. SAE International Releases Updated Visual Chart for Its “Levels of Driving Automation” Standard for Self-Driving Vehicles, 2018 .
3. ERTRAC working group automated driving roadmap report, 2017.
4. McKinsey and Company. Automotive revolution perspective towards 2030. Advanced Industries Report, 2017.
5. Rajamani Rajesh. Vehicle dynamics and control, mechanical engineering series. Edition 1, Spinger US, 2006.
6. Shakouri P, Ordys A. Nonlinear model predictive control approach in design of adaptive cruise control with automated switching to cruise control. Control Eng Pract. 2014;26:160-177.
7. ISO 15622: Intelligent transport systems, “Adaptive Cruise Control Systems”, performance requirements and test procedures 2018.
8. Koschi M, Pek C, Maierhofer S, Althoff M. Computationally efficient safety falsification of adaptive cruise control systems. IEEE Intelligent Transportation Systems Conference. 2019;2879-2886.
9. Corona D, Schutter BD. Adaptive cruise control for a Smart car: A comparison benchmark for MPC-PWA control methods. IEEE T Contr Syst T. 2008; 16:365-372.
10. Kakade R. Adaptive cruise control system: Design and implementation. 5th International Conference on Information Science and Technology (ICIST), 2008.
11. Ali Z. Transitional controller design for adaptive cruise control systems. University of Nottingham, 2008.
12. Gunter G, Stern R, Work DB. Modeling adaptive cruise control vehicles from experimental data: a model comparison. IEEE Intelligent Transportation Systems Conference. 2019;3049-3054.
13. Hong D, Park C, Yoo Y, Hwang S. Advanced smart cruise control with safety distance considered road friction coefficient. Int J Comput Theory Eng. 2016;8:198-202.
14. Haroon Z, Khan B, Farid U, Ali SM, Mehmood CA. Switching control paradigms for adaptive cruise control system with stop-and-go scenario. Arab J Sci Eng. 2019;44:2103-2113.
15. Sivaji VV, Sailaja. Adaptive cruise control systems for vehicle modeling using stop and go manoeuvres. Int J Eng Res Applications. 2013;3:2453-2456.
16. Hirz M. Basics of longutudinal vehicle dynamics. Graz University of Technology. Automotive Engineering, 2016.
17. Dias JEA, Pereira GAS. Longitudinal model identification and velocity control of an autonomous car. IEEE Trans Intell Transp Syst. 2015;16:776-786.
18. Can Özyurtlu, OrhanAlankuş. elektrikliaraçlarindinamikvepara metrikmodellenmesi, kütlevesürtünmekat-sayisitahminedilmesi. [Dynamic and parametric modeling of electric vehicles, estimation of mass and friction coefficient]. Otomatik Kontrol Ulusal Toplantısı. YıldızTeknikÜniversitesi, İstanbul, TOK. 2017:21-23.
19. Shakouri P, Ordys A, Askari M, Laila DS. Longitudinal vehicle dynamics using Simulink/Matlab. UKACCConference Proceeding. 2010:1-6.
20. Naus GJL, Ploeg J,Van deMolengraft MJG,Heemels WPMH, Steinbuch M. A model predictive control approach to design a parameterized adaptive cruise control. Automotive Model Predictive Control. 2010:1-11.
21. Kural E, Güvenç BA. Model predictive adaptive cruise control. IEEE International Conference on Systems. 2010:1455-1461.