EBCCSP2018 Conference is financially sponsored by the Institute for Innovation & Technology

Technical Sponsors

Centre National De La Recherche Scientifique

Université de Perpignan Via Domitia

University of Perpignan
Session 3
Event-Based Vision and Industrial Control Systems

Day 2 – June 28, 10:00-11:00

Session 4
Robustness Testbed for Industrial Control System Components

Day 2 – June 28, 11:00-11:30

Coffee Break

Day 2 – June 28, 11:30-12:30

Lunch

Day 2 – June 28, 12:30-14:00

Coffee Break

Day 2 – June 28, 15:00-15:30

Session 5
Event-Based Control & Signal Processing in Biomedical Applications
Tuning of a Reset Controller via Genetic Algorithms for an ACC System in Following Mode

Miguel Cerdeira, Pablo Falcon, Emma Delgado and Antonio Barreiro
School of Industrial Engineering
University of Vigo
Vigo, Spain
Email: {mcerdeira, pfalcon, emmad, abarreiro}@uvigo.es

Miguel Diaz-Cacho
School of Computer Engineering
University of Vigo
Vigo, Spain
Email: mcacho@uvigo.es

Abstract—The aim of this work is to explore the potential of genetic algorithms to tune a reset controller for an Adaptive Cruise Control (ACC) system in following mode in order to outperform a linear controller for changes in the safety distance with respect to the front vehicle. The controller has been tuned for a restrictive set of comfort and high-performance design requirements.

Index Terms—Reset Control, Genetic Algorithms, Adaptive Cruise Control

I. INTRODUCTION

This article is based on two previous works [1], [2]. It consists of a reset controller for an Adaptive Cruise Control system. As opposed to the aforementioned works, in this one, the tuning of the parameters of the controller is done by means of genetic algorithms.

An ACC system is just an upgrade to the standard Cruise Control (CC) system, which actuates on the accelerator to keep a constant velocity set by the driver [3]. An Adaptive Cruise Control system has a sensor (sonar, radar or lidar) to measure the existing distance to the vehicle ahead. The system is also endowed with breaking capabilities, so that, if there is a car located at a certain distance, the vehicle will switch from speed to distance control [4]. Therefore, an ACC system adapts to different traffic conditions avoiding that the driver have to reactivate the system every time the vehicle is obliged to break, as it was the case for a Cruise Control system. Due to these operational characteristics, ACC systems are considered to be indispensable for the future generations of intelligent vehicles [5].

Nowadays, numerous vehicles are equipped with this technology. The first manufacturer which introduced the ACC system was Mitsubishi which was based on laser technology and, as opposed to nowadays systems, it did not actuate on the brakes but on accelerator and gearbox [6]. Today, manufacturers such as Audi, BMW, Jaguar or Mercedes-Benz equip some of their models with this system.

A reset controller is just a standard compensator endowed with a reset mechanism which resets to zero or to a certain percentage one or several of the controller states, whenever a particular condition holds. The first reference existing in the literature was published in the work of J. Clegg [7]. In this work, Clegg demonstrated the advantages of reset control compared to a classic control. However, despite its advantages, the study of reset control was abandoned until the 1970s, when Horowitz published two papers about it [8], [9] where he demonstrated how the use of reset control can help to overcome the fundamental limitations which affect linear systems [10], [11]. In the 1990s, the number of research groups interested in this control technique incremented considerably. Monograph [12] describes extensively how reset control has been used in different applications.

Since adjusting the parameters of the nonlinear controller is a difficult task and there is no analytic way to find a controller satisfying all the design criteria, genetic algorithms (GA) are used to facilitate the design. A genetic algorithm is a metaheuristic method for optimization, originally developed by Holland [13], which is inspired by the process of natural selection. A GA can explore a far greater range of potential solutions to a problem than other methods. They can be employed to obtain a reset controller which satisfy all the design requirements of the problem.

This article is organized as follows. First, Section II introduces the problem statement. After that, Section III describes how the use of reset control can overcome the fundamental limitations which affect linear systems and the design of the controller. In Section IV, genetic algorithms are introduced and defined to obtain the parameters of the reset controller. Then, the simulations are shown in Section V. Lastly, the conclusions are presented in Section VI.

II. PROBLEM STATEMENT

For an ACC system working in tracking mode (see Fig. 1), when the driver changes the safety distance between vehicles, an increment or decrease of the desired distance produces an error between reference and the distance measured by the sensor, and as a result, the system acts on the acceleration or the brake to compensate that distance error. The system is supposed to be working in steady state when there is a change in the desired distance, that is to say, the following vehicle has already reached the steady state and both cars, follower and leader, are traveling at the same velocity. Since we are mainly interested in the control perspective, we considered a simple scenario with ideal sensors for a longitudinal movement.
A. Vehicle model

Typically, longitudinal control is composed by two loops, an internal loop, which compensates the nonlinear vehicle dynamics, and another external loop which regulates the distance between vehicles [14]. In a real vehicle, the controller must deal with the action on acceleration and brake to maintain a good performance. This article focuses on the outer control loop, assuming that the internal vehicle dynamics are already compensated.

The dynamic model was chosen to be as simple as possible, \( P(s) = 1/s^2 \), as it was already done to simplify the problem in [1].

B. Control loop

The control loop can be seen in Fig. 2, where \( C \) is the controller, \( 1/(s+1) \) represents the actuation dynamics and \( P \) is the plant.

The system compares the measured distance to the reference and then, computes the error. Acceleration will be calculated by the controller with respect to the error. However, this acceleration will not be the real acceleration of the vehicle due to the aforementioned dynamics (\( \tau = 0.4 \) estimated conservative value [15]). It is represented by the block \( 1/(s+1) \) whose output is the real acceleration of the vehicle. Block \( P \) has as its output the position of the vehicle. It must be noted that, the sensor has to be simulated. Therefore, the control loop for the simulation can be seen in Fig. 2.

\[
\begin{align*}
\dot{d}_{ref} & \rightarrow e \rightarrow C \rightarrow \ddot{x}_{est} \rightarrow 1/(s+1) \rightarrow \ddot{x} \rightarrow P \rightarrow x \rightarrow x_{lead} \rightarrow \dot{x}.
\end{align*}
\]

Fig. 2. Simulation control loop.

A constant spacing strategy was considered. The distance set by the driver is kept constant, \( d = d_{ref} \). \( d_{ref} \) is the reference of the safety distance and \( d \) is the distance measured by the sensor.

III. RESET CONTROL

A. Reset control and fundamental limitations

As mentioned above, the objective of this work is to investigate the use of genetic algorithms for tuning a reset controller for an ACC system. For that reason, this section is devoted to the description of reset control, its advantages with respect to linear control and its implications for the study at issue.

A reset controller, with input \( e(t) \) and output \( u(t) \), is defined by the state-space equation \( u(t) = Cx(t) \) and (1) where \( A_r \) is a diagonal matrix whose elements are ones for those states which are not reset and zeros for those which are reset and \( x(t) \) the state vector (not to be confused with \( x \) in Fig. 1 and Fig. 2).

\[
\begin{align*}
\dot{x}(t) &= Ax(t) + Be(t) & \text{if } e(t) \neq 0 \\
x(t^+ &= A_r x(t) & \text{if } e(t) = 0 \quad (1)
\end{align*}
\]

Most of the time, a reset controller behaves like a linear compensator but, whenever the reset condition holds, one or more states are reset [12]. The linear compensator subjected to the reset action is referred to as base linear controller. Linear systems are subject to the well-known fundamental limitations. Particularly, when the system presents a simple integrator in open loop, it has been demonstrated that for every linear controller \( \int_0^\infty e(t)dt = 1/K_v \) is verified [16]. In this case, as it was said in Section II, the vehicle is modeled with two integrators, and, consequently, this limitation holds being the velocity gain \( K_v \to \infty \). In summary, every linear controller will have the following restriction for any second order plant: \( \int_0^\infty e(t)dt = 0 \).

Nonetheless, non-linear controllers are not subject to this limitation. Fig. 3 shows the step response of the two systems, the linear and the reset system.

From the point of view of rise time and overshoot, in Fig. 3 it can be clearly appreciated that the reset controller outperforms the linear one (the design of the controller will be explained in Section III-B). A priori, it could be thought that there exist a linear controller (different from the base linear controller) capable of achieving better results than the reset compensator. However, this is not possible since the response must fulfill a set of limitations such as those specified in Fig. 3 and labeled A, B, C as it was demonstrated for an integrator plant in [16].

A is related to the smoothness of the initial part of \( y(t) \), for \( 0 \leq t \leq t_1 \). In our case study, it will be limited by the comfort specifications, i.e. acceleration \( |\ddot{y}(t)| \) and jerk \( |\dot{\ddot{y}}(t)| \). B, which corresponds to the time interval \( t_1 \leq t \leq t_2 \), is a limitation related to the system robustness that prevents OS = \( max_e(y(t) - 1) \) from exceeding a certain value. Given that \( y(t) \) represents the inter-vehicle distance, excessive overshoot could endanger the safety of the system. C corresponds to the
settling time which imposes a high bandwidth (low $t_2$) and accuracy in the transient response.

Every linear solution satisfies $\int_0^\infty e = \int_0^{t_1} e + \int_{t_1}^{t_2} e + \int_{t_2}^\infty e = 0$. These integrals correspond to the error areas, i.e., the areas comprised between $y(t)$ and the step input $y = 1$. Assuming that the linear solution meets limitations $A$ and $C$, $A$ limit yields a large positive integral $I_A$, whereas the limit given by $C$ can produce either a positive or negative integral but with a fewer absolute value than integral $I_C$. Given that $I_C \ll I_A$, from $0 = \int_0^\infty e$, the integral of the error has to be more negative than $-I_A + I_C$. In other words, any linear controller meeting $A$ and $C$, will necessarily produce an average overshoot $AO$ given by (2) [1].

$$AOS = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} (-e(\tau)) \, d\tau \geq \frac{1}{t_2 - t_1} (I_A - I_C) = AOS_{\text{min}}$$

For the ACC system, large values of $AO$ during $[t_1, t_2]$ would produce a pronounced difference between the real intervehicular distance and the input reference, which would result in serious risks for safety.

In summary, nonlinear controllers are not subject to fundamental limitations which, in turn, affect linear ones. Particularly, the reset system shown in Fig. 3 can produce a response meeting limitations $A, B, C$, which is not attainable by means of a linear regulator.

This controller structure was chosen since it is the simplest realization to which a reset mechanism could be applied.

**B. Controller design**

Firstly, the physical and comfort limits, acceleration and jerk, have been taken into consideration. These limits have been taken from ISO 22179 norm [17] and they are related to barrier $A$, depicted in Fig. 3. It is also necessary that settling time is not too high (barrier $C$), in other words, the overshoot does not prolong for too long.

The controller selected is a lead compensator represented by the transfer function in (3). If the control loop of Fig. 2 is transformed into a deviation control loop, where the distance between vehicles is controlled, the measured distance can be treated as a motion deviation from the nominal trajectory [1].

A typical control loop is obtained by changing the variable of control, as it is shown in (4). The plant is $-\frac{1}{\tau s + 1}$ and therefore, the phase will be close to 0 degrees. As there exists a delay in the actuation defined by a time constant of $\tau$, its presence smooths the values of acceleration and jerk. If this block did not exist, other pole would have to be added to avoid infinity values of jerk.

$$C(s) = \frac{a_1}{s + a_2} + a_0$$ (3)

$$\begin{align*}
\dot{d} &= x_{\text{lead}} - x \\
\dot{x} &= \dot{x}_{\text{lead}} - \ddot{x} \\
\ddot{x} &= -\dddot{x}
\end{align*}$$

As far as the reset condition is concerned, the zero crossing of the controller input is normally chosen. In this work, the reset condition is not investigated and a zero-crossing reset policy is considered. Regarding the action of the reset controller, usually, a full state reset is used although it can be set to another arbitrary value selected by the designer. In this work, GA will provide the reset percentage, related to the magnitude of the reset action. The equation of the reset compensator can be seen in (5).

$$u(s) = a_0 \, e(s) + \frac{a_1}{s + a_2} e(s)$$

As it was explained in [2], the usual design procedure from [18] for a reset controller cannot be used for a double integrator plant. This design method consists in, first, selecting a linear controller $C(s)$, named base linear controller, with a fast and underdamped response. Fig. 6 depicts an example. The closed-loop can be approximated by a second order system with frequency $\omega$ and a low value of damping $\xi$.

The second part of the design procedure involves applying a full reset whenever the error signal crosses zero. It can be demonstrated that at those instants, the reset action does not have a significant effect.

If the plant $-\frac{1}{\tau s + 1}$ has a control signal $u(t)$ and an output signal $d(t)$ and the parameter $\tau$ has a low value, the signal $u(t)$ oscillates with almost the same phase than $d(t)$, as it can be seen in Fig. 6 and Fig. 7. $e(t)$ and $u(t)$ are near to zero when the reset event is triggered and, therefore, the state value is zero too. Consequently, the usual procedure of design from [18] should not be used for this case because is not effective for the plant $P$.

For that reason, it has been decided to reset to a non-zero value. The new value of the state after a reset action results from multiplying the current value of the state and a
parameter called the reset percentage, known as \( p_r \), which is also calculated by means of GA.

In summary, the proposed controller is:

\[
\begin{align*}
\dot{\zeta} &= -a_2 \zeta + e & \text{if } e(t) \neq 0 \\
\zeta^+ &= p_r \zeta & \text{if } e(t) = 0 \\
u &= a_1 \zeta + a_0 e
\end{align*}
\]

where \( \zeta \) is the controller state. The controller contains four parameters: \( a_0, a_1, a_2 \) and \( p_r \).

IV. GENETIC ALGORITHM

As it was mentioned before, the use of genetic algorithms (GA) is proposed to find a good set of parameters for the reset controller. GA is a method for solving constrained and unconstrained optimization problems based on a natural selection process that imitates biological evolution. With a well-posed problem, the genetic algorithm allows to find a good solution fulfilling all the requirements by means of an iterative search.

The use of GA allows to build an automatic search of the different parameters of the controller. In each step, the algorithm modifies randomly a population of individual solutions, selecting individuals from the current population and creating the next generation of solutions. Over successive generations, the population "evolves" into an optimal solution.

In this work, GA have been employed to obtain the parameters \( a_0, a_1, a_2 \) and \( p_r \) of the controller (3) which best fit the design specifications.

To evaluate the validity of the solution for each of the candidates, a cost or fitness function must be adequately defined. The fitness function must be defined by the designer based on his experience and it requires a set of rules which determine the reliability of the solution. In general, a typical genetic algorithm may comprise the following elements [19]:

- A way of assessing how good or bad the individual solutions within the population are.
- A method for mixing fragments of the better solutions in order to form, on average, better solutions.
- A mutator operator is employed for the genetic algorithm not to result in a permanent loss of diversity within the solutions.

Once established the above, the implementation is done by using Matlab Optimization Toolbox and consists of the following steps:

1) A population of size 50 is randomly initialized within the lower and upper bounds of \( a_0, a_1, a_2 \) and \( p_r \).
2) Each member of the current population is scored by computing its fitness value from (7). These values are called the raw fitness scores.
3) 5\% of the individuals with the lowest fitness are chosen as elite and directly pass to the next generation. These are known as elite children.
4) 80\% of the remaining 95\% of the descendant generation is obtained by combining the genes of a pair of parents. These are known as crossover children.
5) The rest of the specimens to complete the new generation are created by introducing random changes, or mutations, to a single parent. These are known as mutation children.
6) The algorithm replaces the current population with the children to form the next generation.
7) The algorithm stops when one of the stopping criteria is met: time limit, fitness limit, stall generations or function tolerance.

The proposed fitness function for this problem is:

\[
\begin{align*}
CF_{GA} = & w_1 \frac{\text{accel}_{\text{max}}}{\text{accel}_{\text{norm}}} + w_2 \frac{\text{jerk}_{\text{max}}}{\text{jerk}_{\text{norm}}} + \\
& + w_3 \frac{t_r}{5} + w_4 \frac{OS}{5} + w_5 \frac{t_s}{10} \quad (7)
\end{align*}
\]

where \( \text{accel}_{\text{max}} \) is the maximum magnitude of the averages of longitudinal acceleration in absolute value calculated for every time window of 2 seconds, \( \text{jerk}_{\text{max}} \) is the maximum magnitude of the averages of longitudinal jerk in absolute value for every time window of 1 seconds, \( t_r \) is the rise time, \( OS \) is overshoot and \( t_s \) is settling time. To compute the first term of the cost function, the acceleration contribution, it is necessary to know the sign of the acceleration magnitude because, for positives values of it, the limit \( (\text{accel}_{\text{norm}}) \) is 2 \( m/s^2 \) and otherwise, 3.5 \( m/s^2 \). The maximum magnitude of jerk is limited in the norm by a value of 2.5 \( m/s^3 \). The other requirements for the controller are a rise time of 5 seconds, a overshoot of 5\% and a settling time of 10 seconds. The \( w_i \) parameters are the weight of each term of the function and it is used to establish the level of importance of the solution. In this case, the weights \( w_i \) of the cost function were set at same value for all parameters, accomplishing the requirements to the same extent.

The stopping criteria is a composition of three limits: a generation limit (100*number of variables), stall generations (50).
and function tolerance ($10^{-6}$). The generations limit specifies the maximum number of iterations for the genetic algorithm to perform and the stall generations allows to stop the execution of the algorithm if the average relative change in the fitness function value is less than or equal to the function tolerance. This tolerance reflects the minimal change in value to take into account.

With all the previous assumptions, the obtained controller can be seen in equation (8), with $p_r = 8.9695$. Equation (9), where $\zeta$ is the state, represents the reset controller. This kind of reset regulator is called FORE (First order reset element).

$$C(s) = \frac{-1.0625}{(s + 2.0679)} + 0.5956$$ (8)

$$\begin{cases} \dot{\zeta}(t) = -2.0679 \zeta(t) + e(t) & \text{if } e(t) \neq 0 \\ \zeta(t^+) = 8.9695 \zeta(t^-) & \text{if } e(t) = 0 \end{cases}$$ (9)

V. SIMULATIONS

In order to assess the efficiency of reset control, some simulations were performed where the response of the system was tested for a change in the input reference for both the linear and the reset controller.

As previously mentioned, a constant spacing policy was considered for this work. Initially, both vehicles, leader and follower, are assumed to be traveling at 25 m/s. Both cars are separated by 60 meters. From these initial conditions, the set-point is changed from 60 meters to 75 meters.

Fig. 8 depicts the responses of both the linear and the reset systems. It can be clearly appreciated how the reset system has a better and faster response than the linear system since both overshoot and settling time are reduced considerably. As explained in III, this is due to the fact that the integral of the error for the linear system is equal to 0, which results in a greater overshoot.

Fig. 9, Fig. 10 and Fig. 11 show control, instantaneous acceleration and instantaneous jerk, respectively. According to ISO 22179 [17], which contains the values of acceleration and jerk to guarantee comfort, average jerk calculated for intervals of 1 s cannot surpass $2.5 \text{m/s}^3$ when the vehicle’s velocity is greater than 20 m/s; whereas average acceleration for an interval of 2 s cannot be lower than $-3.5 \text{m/s}^2$. Taking into account this, figures 12 and 13 confirm that the comfort limitations are met.

Despite the results obtained, one would think that there might be any linear controller adjusting to all the design
The controller found by this method meets all the comfort requirements. In order to demonstrate that this is not attainable, the values of $AOS$ and the integrals of the errors have been computed.

Concerning $AOS$, limitations related to rise time and settling time have been imposed. These must be met in order not to surpass jerk and acceleration limits (see Fig. 14). Barrier A is mainly given by the jerk limitation and barrier B by an exponential function whose decay rate is given by the slowest pole of the system.

To compare our reset controller with any linear controller meeting limitations A and C, $AOS_{min}$ has been calculated for the linear controllers and for our reset controller. $AOS_{min}$ (minimum overshoot for a linear controller) is calculated by means of equation (2) in Section III-A. Barrier C begins at the settling time (5%). Table I contains the integrals $I_A$ and $I_C$ and $AOS_{min}$ as well as $AOS_{reset}$. It can be confirmed that $AOS_{min}$ is greater than $AOS_{reset}$.

The table II contains $IE_{reset}$ and $IE_{linear}$. As expected, $IE_{linear}$ is equivalent to zero when time tends to infinity.

VI. CONCLUSIONS

The objective of this work was to investigate the use of genetic algorithms to tune a reset controller for an adaptive cruise control system. In this case, the parameter to be controlled is the distance between the vehicle and a preceding vehicle. To perform this task, an ideal scenario and a simple dynamic model were considered to focus the study on the reset control strategy to be applied. Under these conditions, and by using an iterative search, all the parameters of the reset controller, including the reset percentage, were obtained. The controller found by this method meets all the comfort specifications detailed in ISO 22179 as well as all the physical limitations.

The results obtained prove the effectiveness of reset control and its advantages compared to linear control. ACC systems endowed with a reset controller may yield good responses in terms of rise and settling time without producing an excessive overshoot.

ACKNOWLEDGMENT

This work has been supported by the European Regional Development Fund (ERDF) and MINECO, Spain, under the project DPI2016-79278-C2-R.

REFERENCES


