Development and testing of a fully Adaptive Cruise Control system

Gennaro Nicola Bifulco a,*, Luigi Pariota a, Fulvio Simonelli b, Roberta Di Pace c

a Department of Transportation Engineering, University of Naples Federico II, via Claudio 21, 80125 Napoli, Italy
b Department of Engineering, University of Sannio, Corso Garibaldi 107, 82100 Benevento, Italy
c Department of Civil Engineering, University of Salerno, via Ponte Don Melillo 21, 84084 Fisciano (SA), Italy

A R T I C L E   I N F O
Article history:
Received 26 July 2010
Received in revised form 5 July 2011
Accepted 14 July 2011

Keywords:
Adaptive Cruise Control (ACC)
Car-following
Learning-machine
Advanced Driving Assistance Systems (ADAS)
Intelligent Transportation Systems (ITS)
Dynamic systems

A B S T R A C T
Adaptive Cruise Control systems have been developed and introduced into the consumer market for over a decade. Among these systems, fully-adaptive ones are required to adapt their behaviour not only to traffic conditions but also to drivers’ preferences and attitudes, as well as to the way such preferences change for the same driver in different driving sessions. This would ideally lead towards a system where an on-board electronic control unit can be asked by the driver to calibrate its own parameters while he/she manually drives for a few minutes (learning mode). After calibration, the control unit switches to the running mode where the learned driving style is applied. The learning mode can be activated by any driver of the car, for any driving session and each time he/she wishes to change the current driving behaviour of the cruise control system.

The modelling framework which we propose to implement comprises four layers (sampler, profiler, tutor, performer). The sampler is responsible for human likeness and can be calibrated while in learning mode. Then, while in running mode, it works together with the other modelling layers to implement the logic. This paper presents the overall framework, with particular emphasis on the sampler and the profiler that are explained in full mathematical detail. Specification and calibration of the proposed framework are supported by the observed data, collected by means of an instrumented vehicle. The data are also used to assess the proposed framework, confirming human-like and fully-adaptive characteristics.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

Adaptive Cruise Control (ACC) systems have been actively developed and introduced into the consumer market by vehicle manufacturers in the past decade. They extend earlier systems (CCC – Conventional Cruise Control) to cases when driving at a fixed constant speed is not possible because of traffic conditions.

This paper deals with human-like ACCs, where control not only ensures safety but also produces trajectories consistent with observed drivers’ behaviours. In other words, a human-like ACC should choose the most natural driving behaviour among all those consistent with safety. Human likeness contributes to identify a fully-adaptive system, able to adapt not only to actual traffic conditions but also to driver’s actual attitudes and preferences. This has been often recognised in the literature as a key feature, given that, in order to earn acceptance from drivers, ACCs should be perceived as a sort of co-pilot. In Kesting et al. (2008) for instance, even if the analysis is mainly oriented to the effects of ACC on traffic stability and performance (a topic not covered by our work), it is recognised that the system should be able to apply different driving behaviours in different
traffic conditions (automatically detected) exactly as a human driver would have done. This point is confirmed in Viti et al. (2008), where frequent deactivations of ACCs are observed during a field operational test in the Netherlands. This occurs more frequently in dense traffic conditions and could be due to a scarce human likeness. User acceptance is not only important for vehicle manufacturers (market penetration) but also crucial from a social perspective (actual adoption of the system).

The importance of fully adaptive systems has long been recognized in the scientific literature and efforts have been made to identify drivers’ preferences through suitable parameters (see for example Reichart et al., 1997; Franchet al., 1998). However, most of the applications propose only traffic adaptation, without dynamic adaptation also to drivers’ attitudes and preferences. The parameters of the models (and characteristics of the drivers) are often proposed according to values determined statistically off-line (Marsden et al., 2001; Yi and Moon, 2004; Zheng and McDonald, 2005; Moon and Yi, 2008). In most practical implementations, the approach by vehicle manufacturers is to ask for manual selection of a driving style among a set of predefined values. However, the application of a (predefined) set of parameters cannot take into account the complexity and the great heterogeneity of drivers’ preferences, skills and attitudes, nor the fact that these can vary for a given driver, depending on trip context, purpose and duration or on other trip-specific conditions, often very difficult to be fully understood. Heterogeneity and adaptation are confirmed in several works, both directly oriented to ACCs (Ervin, 2005) and traffic simulation (Wu et al., 2003; Ranjitkar et al., 2004; Punzo and Simonelli, 2005; Brockfeld et al., 2005). A performance-based benchmarking of car-following models in representing observed driving behaviours was carried out by Ranjitkar et al. (2005); it was found that the same models performed differently from driver to driver and that the variability of the parameters of the models was higher than inter-model variability. The same result was found by Ossen and Hoogendoorn (2010). However, we are not interested here in car-following in order to reproduce driving behaviours for traffic simulation purposes. Rather, we propose an approach based on a linear car-following model, whose parameters can be calibrated on-demand in order to reproduce observed driving behaviours (human likeness) to be applied in ACC contexts.

Calibration can be carried out through a learning-machine approach, in real time, on the driver’s request and by means of a short manual-driving session. The on-demand calibration can be carried out for any single driver, in any single driving context and for any single driving session, or even within the driving session, if desired and requested by the driver, thus allowing for the implementation of a fully-adaptive ACC.

The proposed ACC is structured as a multilayer approach where different modelling layers are responsible for different tasks. The first layer is the sampler, in charge of the human likeness of the control logic. It actually coincides with the ACC-oriented car-following model, and the real-time (and on-demand) estimation of its parameters allows implementation of the learning mode phase. Once learning is terminated, the logic is applied within a broader modelling structure and the running mode is activated. If the driver is no longer satisfied, he/she can switch again to the learning mode, regaining control of the vehicle for a new training session. Clearly the proposed framework is much more adaptive than those based on general, albeit sophisticated, preset rules (Alonso et al., 2005; Tange et al., 2005).

One of the key aspects of the approach is the ACC-oriented car-following model implemented in the sampler. Most existing car-following models were originally conceived for microscopic simulation oriented to traffic analyses, planning or control purposes. However, in recent years their application to cruise control, driving simulators and psychological studies of driving behaviour has also been proposed. In ACC contexts car-following models could be useful to develop and implement control logics (e.g. Chakroborty and Kikuchi, 1999) to be embedded into on-board electronic control units (ECUs).

Simonelli et al. (2009) have already shown that regressive models, based on simple linear or higher-order regressions or on artificial neural networks (ANNs), can perform very satisfactorily, at least when implemented as the basis for ACC applications. In particular, the advantages of a linear approach have already been demonstrated in reproducing observed speeds in car-following conditions. This paper moves on from these previous considerations to present a different model oriented towards reproducing spacing (gap distance with the vehicle ahead) instead of speed. The work of selecting the most effective specification (linear, artificial neural network, etc.) for the spacing-oriented approach, already done for the speed-oriented one, is not repeated here and the linear model is directly applied; our results in Section 5 (together with the easiness-of-use of the linear approach) justify the adopted formulation.

A review of car-following models lies beyond the scope of this paper (see Tordeux et al., 2010, for a recent review). For our purposes, it is sufficient to observe that traditional approaches are often characterized by a relatively high complexity and by relatively poor computational performance in case of real-time and fast-time applications, not compatible with the performance required by on-board ECUs. This is even more crucial for our fully-adaptive control logic that needs to be also calibrated on demand and not only applied in real time. Moreover, the parameters of traditional car-following models rarely present, in real cases, the behavioural interpretation they claim to possess. Finally, as shown in several works (e.g. Ranjitkar et al., 2004; Punzo and Simonelli, 2005), complex models perform in reproducing experimental data not much better (if not worse) than simpler ones (Newell, 2002). This is particularly true in validation tests, probably because of over-fitting phenomena due to the large number of parameters exposed by complex models.

Starting from all previous considerations, this work presents some crucial developments towards a human-like, fully-adaptive ACC system based on a linear car-following model suitable for real-time and on-demand estimation and application. The system is intended to be adopted in stand-alone mode and does not rely on vehicle-to-vehicle (v2v) communication. Of course, v2v communication is compatible with our framework; it can be viewed as an alternative way for collecting spacing and relative speed with respect to the leading vehicle. This could replace the use of radar/lidar or rather allow for the integration of more data sources. The effects on traffic flows of the market penetration of our system are not treated here and are left for future possible works.
The proposed modelling framework is composed by four layers. The framework is presented in Section 2, where the data collected to support development and testing are briefly described. The first layer (the sampler) is responsible for the main ACC control logic and is described in Section 3. The profiler is presented in Section 4; it is a key component which is asked to translate the driving behaviour produced by the sampler into admissible (continuous) kinematic profiles, suitable to be applied to the controlled vehicle. The third and fourth modelling layers, respectively the tutor and the performer, are not covered in depth in this paper. Roughly speaking, the tutor ensures that the reference driving trajectory obtained by using the profiler is ultimately safe, so that it can be applied by the performer. The performer is in charge of actuation by controlling the vehicle actuators (e.g. the throttle, the brakes, etc.). Development and testing of the performer goes far beyond the scope of this paper; we assume that an effective and efficient performer already exists. For example, CarSim software (CarSim 8: Mechanical Simulation Corporation, website) can be used for this purpose. Results from the application of the modelling framework are presented in Section 5 and compared with real-world collected data. Our conclusions summarize and discuss the main findings and introduce some research perspectives.

2. The model architecture and the supporting data

In this section the four-layer framework is briefly presented (Fig. 1). The approach was explicitly conceived for ACC applications. We argue that it can be employed for other kinds of applications but here we do not discuss this opportunity.

The inception idea is that the driver’s behaviour in car-following conditions can be identified by means of the time-series of spacing (intervehicle separation distance, in metres), sampled at a given frequency (e.g. at 1 Hz). Computation of this sequence is the role of the sampler. Within the sampler the vehicle dynamics is neglected, as is consistency among different kinematic variables (position, spacing, speed, acceleration, etc.). The second modelling layer (the profiler) recovers the full (and consistent) representation of vehicle trajectory by adopting a time-continuous state-space approach. Of course, the profiler has to ensure that the resulting trajectory fits the points identified by the sampler; this is obtained by assuming these points as the requests supplied to the profiler. The profiler also checks whether these requests are actually admissible. Checking is based on general kinematic considerations and rough hypotheses about vehicle performance. For instance, if the sampler requests too high a variation of position to be satisfied with an admissible acceleration, the profiler limits the reached position consistently with a predefined maximum acceleration.

The trajectory produced by the profiler is continuous, consistent and likely to be admissible; it represents a reference trajectory. However, the actual trajectory can be different from the reference one given that it results from the actuation (throttle, brake, engine, etc.) performed by the vehicle mechanics and electronics, as well as from interaction with the road (slope, grip, etc.). The simulation of how the vehicle and its actuators are able to actually match the profiler-supplied reference trajectory pertains to the performer. Actually, development of a tool like the performer is a typical service for the automotive sector. Some commercial tools already exist to this aim (e.g. CarSim). Hence the performer is here totally neglected and the assumption is made that such a tool is available and able to actuate the reference trajectory; on these assumptions are also based all results presented in section 5.
Between the profiler and the performer, the tutor is inserted; it ensures that safety conditions are satisfied. It computes the maximum allowed speed (or spacing, or position increment) compatible with the safety, revealed by applying a safety model to real-time (and high-frequency – much higher than the sampler) sensor measurements. The performer then tries to apply the lowest position increment suggested by the profiler or the tutor.

The modelling layers described here have to be specified, calibrated and validated. In particular, a great effort in terms of calibration has to be devoted to the sampler which is responsible for reproducing a human-like cruise control logic. Thus, a considerable amount of data is required. These can be collected by means of different techniques. For instance, two or more vehicles can be equipped with Global Positioning System (GPS) devices (Gurusinghe et al., 2002; Punzo and Simonelli, 2005) and drawbacks. That said, these experiments require non-negligible equipment costs and most of the experimental effort an alternative, one vehicle can be traced, being equipped with sensors able to record its relative kinematics with respect to the vehicle ahead and/or behind (Wu et al., 2003). Both the GPS-based and the sensor-based approaches have advantages and drawbacks. That said, these experiments require non-negligible equipment costs and most of the experimental effort is towards unbiasing the observed car-following trajectories (see for instance Ma and Andreasson, 2005; Punzo and Simonelli, 2005). Some early versions of the ACC logic proposed herein were presented in Simonelli et al. (2009) and Bifulco et al. (2008) and were respectively based on pure GPS data (properly corrected for un-biasing) and model-generated data, based on parameters accurately calibrated by using GPS trajectories. Unfortunately, for pure GPS data heavy-filtering and consistency-reaching procedures are required so that, given the extreme accuracy needed for ACC applications, few minutes (1–3) of useful trajectory were available for each of the eight trajectories collected. Moreover, a sizeable part of each trajectory was used for calibration and just a small part was available for validation. In the case of model-generated trajectories, the parameters of a standard Gipps (1981) were accurately calibrated against real-world data. The obtained model was then used to generate (given a long GPS-revealed trajectory of the sole leader) a car-following trajectory. In this second case the trajectory obtained was long enough but perhaps too smooth and synthetic.

However, the extreme accuracy of the proposed regression in reproducing Gipps-produced synthetic data (in more than 90% of cases the discrepancy with the Gipps trajectory was less than 10%) indirectly proves the suitability of the linear approach or, at least, that its performance is comparable with that of widely accepted non-linear car-following models.

The research advances which we present are based on new data, collected by using an instrumented vehicle with radar-based sensors. This collection technique has the great advantage of being consistent with the technological context (radar or lidar sensors) in which ACCs are widely employed on the field. The instrumented vehicle is a Fiat-Multipla equipped with acquisition and video-recording devices. It allows data to be collected not only about the vehicle itself but also about vehicles ahead and/or behind. The technological system is based on a central unit consisting in a notebook PC that manages real-time data acquisition. The PC mounts a PCMCIA-CAN (with 2 ports) and a PCMCIA-DAQ card (allowing for 8 analog outputs, 8 digital outputs, 2 counter/timers). The vehicle is able to supply several data, including vehicle speed, the position of the accelerator pedal, the brake and clutch position, the rotation angle of the steering and many other signals intercepted from the vehicle controller area network (CAN). Moreover, it is possible to intercept data from the two on-board radars (TRW Auto-cruise AC10) and, in particular, the relative speed and relative spacing with respect to up to four vehicles ahead or behind. All data can be collected with high frequency; the one employed in our studies was 10 Hz. This value is consistent with similar experiments described in the literature (e.g. Wu et al., 2003), but higher values can be reached if needed. The video-capture system consists of three basic elements: cameras (rear and front); the titling card; the digital video recorder. The images are captured by the cameras and sent to the titling card that makes the overlay of some information (absolute time, relative speed and relative spacing with respect to the vehicle ahead or behind); finally, the recorder registers the titled videos. All real-time acquisitions are synchronized and managed by using a single software program developed in LabWindows.

Using the equipped vehicle, extensive car-following data were collected in a period from 2008 to 2010 along two different routes near the city of Naples (southern Italy). One route had few intersections and a higher average speed, the other more flow disturbance and a lower average speed. Twenty trajectories were collected, relative to different drivers. Each driver was allowed a short period of acclimatisation to the vehicle and was then asked to follow a confederate vehicle.

As can be seen from the few statistics about the assortment of drivers and the characteristics of driving sessions (Table 1), most of the drivers were male and the incidence of young (less than 30-years old) drivers was high. Weather conditions during driving sessions were mostly good. The length of the observed trajectories was on average of 10 km for route 1 and 6 km for route 2. Of course, significantly shorter than average or longer trajectories were observed in both cases; however, trajectories shorter than 2 min were excluded from the experiment. The average speed (excluding stops at intersections) was 64 km/h on average over the thirteen driving sessions on route 1 and 42 km/h on average over the seven

<table>
<thead>
<tr>
<th>Route</th>
<th>Drivers (%)</th>
<th>Length (km)</th>
<th>Average speed (km/h)</th>
<th>Max speed (km/h)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Female</td>
<td>Young</td>
<td>Number</td>
<td>Good weather (%)</td>
</tr>
<tr>
<td>1</td>
<td>38</td>
<td>77</td>
<td>13</td>
<td>69</td>
</tr>
<tr>
<td>2</td>
<td>43</td>
<td>71</td>
<td>7</td>
<td>86</td>
</tr>
</tbody>
</table>

Table 1
Drivers and driving sessions.
sessions on route 2. Similarly, the max observed speed was (as an average of the maximum values observed over all driving sessions) 105 km/h for route 1 and 82 km/h for route 2; the observed absolute maximum was 130 km/h for route 1 and 120 km/h for route 2.

Data were employed to estimate the parameters of the sampler (first seconds of each trajectory), as well as to compare the results of the modelling framework. It is worth noting that the nature of the fully adaptive approach excludes the calibration of a set of parameters common to all the trajectories, nor the calibration of the dispersion of these parameters. Rather, the parameters are intended to be calibrated for each single driver and for each single driving session (in real time and on demand), according to the learning-mode phase of the ACC system. From this point of view, each of the 20 calibration sessions made available by the data we collected should be treated separately. We show and discuss in Section 5 the results related to one of these trajectories. However, to provide evidence of the general suitability and robustness of the method, we also compare the current trajectory with all the others and, in particular, the envelop will be shown for the best and worst cases. This is not intended to draw general rules but only to investigate the sensitivity and robustness of the approach with respect to different sets of collected data.

3. The sampler: a car-following model for the human-like ACC problem

The sampler is responsible for estimating a time-series for spacing. From this, a time-series of driven distances can be derived. It reproduces as closely as possible the sampling of the car-following trajectory that a driver would have applied in manual driving conditions. The adopted sampling frequency is in the order of magnitude of 1 Hz; this was chosen on the basis of some major considerations:

(a) the typical reaction time of car-following models proposed in the literature is around 1 s; from that a sort of human-like refreshing frequency of 1 Hz can be argued; in other words, we assume that drivers habitually distinguish with a granularity no more detailed than 1 s (of course, if things go smoothly);

(b) even if the human likeness of the trajectory is sampled at a 1 Hz frequency, the ACC system checks the safety at a much higher frequency (say, 10 Hz); this is the task assigned to the tutor; as a result, the system is able to react to stimuli, if dangerous, much more promptly than the driver (say, human likeness is excluded in the case of danger);

(c) the time step between sampled points is a trade-off between opposite interests as expressed by the following points (i) and (ii);
   i. the car-following model implemented by the sampler is based on some approximate assumptions on how the leading vehicle moves; a shorter time step between two successive sampled points bounds the errors introduced by this approximation;
   ii. the car-following trajectory between two sampled points evolves according to the profiler; having fixed the sampled points, some optimisation can be done in the transition, according to some external objectives (e.g. reduction in consumption and/or pollutants); this is aided by a longer time step.

The sampler works with respect to two main traffic regimes: free-flow and car-following. The free-flow regime is where the ACC actually acts as a CCC and a pre-defined speed is applied. This could vary along the route, possibly being associated to location-aware (dynamic) speed regulation policies and on-board speed navigators. That said, the free-flow speed is not a modelling task in the context of this paper and is treated as a known, fixed (constant within each time step) and exogenously given value. Of course, the desired position increment in the case of free-flow speed is easily computable. Thus, a model for the car-following regime is the main focus of the sampler.

It is worth noting again that no safety considerations are made in order to avoid collisions. Actually, given the capital importance of safety considerations, these are applied to the vehicle at a higher frequency and are superimposed upon any other consideration. For such a reason, safety and emergency considerations are not included as sampler tasks. Rather, they are postponed between the profiler and the performer and constitute the main task of the tutor.

The ACC-oriented car-following approach proposed here is based on a simple linear model, able to relate the instantaneous speeds of the leader and the follower and their spacing with the target spacing the follower desires for next time-step. The output of the sampler aims to be a reproduction of driver behaviour, but the analytical structure of the model is not intended to mimic behavioural processes. From this point of view, it might be said that any behavioural interpretation has been distanced from our approach. Drivers’ behaviours are reproduced by a purely descriptive (regressive) model. Hence the first modelling layer is here called sampler (and not, for instance, modeller).

The choice of reproducing spacing is in some respects original. In ACC applications it is more frequent to control the acceleration of the vehicle and only indirectly speeds and distances. However, assume that the speed is almost perfectly reproduced (due to an even more accurate reproduction of acceleration), but for 1 or 2 s the actuated speed is only slightly wrong. After that, the speed is again reproduced with no errors. The few seconds of the wrong speed have induced erroneous spacing; this error does not further increase when the speed is recovered, but it will never be corrected if the goal is to reproduce speed (or acceleration). This does not happen if the spacing is directly controlled, errors in terms of spacing can be recovered. Moreover, if the time series is correct in terms of spacing (and so is vehicle advancement, given the leader’s trajectory), it is even more correct in terms of spacing derivatives (speed and acceleration); with reference to speed, this phenomenon is clearly represented in Figs. 3 and 4 in Section 5.
Application of a linear model in order to reproduce the observed spacings is justified by previous experience discussed in Bifulco et al. (2008), to which the reader should refer for details. The advantage of using more-than-linear formulation for the regressive model was estimated at 9% in terms of better RMSE of the simulated vs. observed spacing. Similarly, the advantage obtained by an even more flexible regression model based on artificial neural networks was measured at 15%. These advantages are not enough, in the authors’ opinion, to forgo the great efficiency of the linear approach, especially in our case where real-time and on-demand estimation of the parameters is required.

As a consequence, the main equation of the sampler is:

\[ \Delta z(k, k + 1) = \beta_0 + \beta_1 \Delta x(k) + \beta_2 \Delta v(k) + \beta_3 v_1(k) \]  
(1)

where \( \Delta z(k, k + 1) \) is the target spacing (intervehicle separation) estimated at time step \( k \) for time step \( k + 1 \); \( \Delta x(k) \) is the spacing at time step \( k \); \( \Delta v(k) \) is the relative speed (difference of speeds) at time \( k \) between the leader and the follower; \( v_1(k) \) is the speed of the leader at time step \( k \).

By definition:

\[
\begin{align*}
    s_l(k) &= \Delta x(k) + s_f(k) \\
    v_l(k) &= \Delta v(k) + v_f(k)
\end{align*}
\]

(2)

where \( s_l(k) \) is the position of the leader at time step \( k \); \( s_f(k) \) is the position of the follower at time step \( k \); \( v_l(k) \) is the speed of the follower at time step \( k \).

To reach the target spacing the vehicle computes at time step \( k \) a target position increment for time step \( k + 1 \):

\[
\begin{align*}
    \Delta s_f(k, k + 1) &= \sigma_f(k + 1) - s_f(k) \\
    &= \sigma_f(k + 1) - \Delta z(k, k + 1) - s_f(k) \\
    &= \sigma_f(k + 1) - \Delta z(k, k + 1) - s_f(k) - \Delta z(k, k + 1) + \Delta x(k) \\
    &= \Delta \sigma_f(k, k + 1) - (\beta_0 + \beta_1 \Delta x(k) + \beta_2 \Delta v(k) + \beta_3 v_1(k))
\end{align*}
\]

(3)

where \( \sigma_f(k + 1) \) is an estimate of the position the follower will reach at time \( k + 1 \). \( \sigma_f(k + 1) \) is an estimate of the position the leader will reach at time \( k + 1 \) and, consistently, \( \Delta \sigma_f(k, k + 1) \) is an estimate of the space driven by the leader from time \( k \) to time \( k + 1 \).

In particular, the sampler estimates at time step \( k \) the distance driven by the leader in the next future by considering a uniformly accelerated motion equation:

\[
\Delta a_L(k, k + 1) = 0.5a_L(k)\Delta T^2 + v_L(k)\Delta T
\]

where \( a_L(k) \) is a direct measure or an estimate of the acceleration of the leader at time step \( k \). Since the controlled vehicle will actually apply a driven distance increment computed according to Eq. (3) and in this equation the driven distance increment by the leader has been estimated, the actual driven distance increment at time step \( k + 1 \) will differ from the target one by a quantity \( Err \). This is the error in estimating the leader progression, deliberately introduced in the model because of the uniformly accelerated motion.

Formally:

\[
Err(k + 1) = \Delta \zeta(k, k + 1) - \Delta x(k + 1) = \sigma_f(k, k + 1) - s_f(k + 1) = s_l(k) + \Delta \sigma_f(k, k + 1) - s_f(k + 1)
\]

(4)

From a modelling point of view, the error is not propagated because at each time step the spacing on which Eq. (1) is based is refreshed according to the real (measured) value \( \Delta x(k) \) and the position of the leader is refreshed by using Eq. (2).

Eq. (3) is the output of the sampler, as well as the input for the profiler. It can be rearranged to make the dynamic process explicit in terms of position progression of the follower:

\[
\sigma_f(k + 1) = \beta_1 s_f(k) + s_f(k)(1 - \beta_1) + v_f(k)(\Delta T - \beta_2) + 1/2a_L(k)\Delta T^2 - \beta_2 \Delta v(k) - \beta_0
\]

(5)

We can also rearrange our formulation in a way similar to the well-known stimulus-response paradigm (Chandler et al., 1958; Gazis et al., 1961), where we define the response as the target variation of the spacing. In fact, Eq. (6) below can be easily verified to be equivalent to Eq. (1).

\[
\Delta \zeta(k, k + 1) - \Delta x(k) = \beta_0 + (\beta_2 + \beta_3) \Delta v(k) + \beta_3 v_f(k) + (\beta_1 - 1) \Delta x(k)
\]

(6)

Eq. (6) states the stimulus-response approach in a non-classic way. The response is broadly defined in the literature as the acceleration applied by the follower; our definition refers to the target spacing which also depends on the distance driven by the leading vehicle in the meantime. Eq. (6) can be specified for the case where the stimulus from the relative speed is null:

\[
\Delta \zeta_{0}(k, k + 1) - \Delta x(k) = \beta_0 + \beta_2 v_f(k) + (\beta_1 - 1) \Delta x(k)
\]

(6.a)

By also imposing that the response is null, we obtain pairs (speed, spacing) that represent equilibrium points:

\[
\Delta \zeta' = \frac{\beta_0}{1 - \beta_1} + \frac{\beta_3}{1 - \beta_1} v'
\]

(7)
In other words, when the stimulus related to the relative speed is null, a car-following dynamics still can be observed (the response is not null), as in Eq. (6a). This happens if the actual spacing is not suitable for the speed at which the vehicles cruise. However, if the system satisfies Eq. (7), then it is at equilibrium and the spacing no longer changes. This phenomenon fits the expectations and confirms that the variables employed in the sampler have been correctly identified. It is worth noting that Eq. (7) implicitly states the admissibility of parameters \( \beta \) of the model. It has to be ensured that for any feasible (non-negative) speed \( \nu' \) a feasible (non-negative) value of \( \Delta x' \) can be obtained. Moreover, it is expected that the regime spacing increases according to the regime speed. In practice, the (linear) curve \( \Delta x' = \Delta x(\nu') \) defined by Eq. (7) has to be a non-decreasing and non-negative function. Sufficient conditions for that are:

\[
\frac{\beta_0}{1 - \beta_1} \geq 0, \quad \frac{\beta_3}{1 - \beta_1} \geq 0
\]  

(8)

Eq. (8) above could be assumed as a constraint during the calibration of the sampler; however, they were heuristically employed within an unconstrained calibration algorithm.

The sampler should be calibrated in real time, while the vehicle is driven by the real driver. It represents the learning mode of the ACC system. Without dealing with more complicated considerations, a simple explicit formulation can be applied that minimises the distance between the model outputs and the observed outputs by solving an ordinary least square (OLS) problem:

\[
\hat{\beta} = (\Gamma^T\Gamma)^{-1}\Gamma^T\gamma
\]  

(9)

where \( \hat{\beta} \) is the vector containing the desired estimates of the model parameters; \( \Gamma \) is the matrix enlisting in each column the values observed at each time step \( k \) for the independent variables \( \{\Delta x, \Delta v, h_i\} \), plus a first column of unitary values, accounting for the estimation of the intercept of our model \( \{\beta_0\} \); \( \gamma \) is the vector of all the observations of the dependent variables (observed target spacing).

Values in \( \Gamma \) and \( \gamma \) are the observed data discussed in Section 2.

Several algorithms can be employed to solve the OLS problem. One of the most efficient for real-time applications is the recursive least squares (RLS) algorithm (Haykin, 2001); it is widely applied in many areas, such as real-time signal processing. In the general formulation it minimizes a weighted least squares cost function related to the input signal under the hypothesis that a new sample of signals is received at each iteration. Given the incremental nature of the algorithm, computation takes a very reasonable time even if a considerable amount of observed data is processed. Compared with most of its competitors, the RLS exhibits extremely fast convergence. However, in-depth discussion of the algorithm or of its convergence and stability issues go well beyond the scope of this paper; this may well be the subject of future works. The algorithm was used here in a heuristic way and the stop criterion for terminating the estimation of the parameters was based on two conditions (occurring jointly): Eq. (8) holds and the objective function of the algorithm improves negligibly. Of course, it is not guaranteed that the algorithm has reached stable solutions but some empirical evidence can be claimed by considering all the successfully performed calibrations (see also the discussion on the calibration results in Section 5). Moreover, effectiveness of the algorithm can be empirically accepted if good fitting of the calibrated model against the observed data is evidenced a posteriori, as happens in all our cases.

4. The profiler

The sampler estimates at each time step \( k \) the distance that the controlled vehicle should drive in order to reach at time step \( k + 1 \) a human-like target spacing. The responsibility of the profiler is to produce a time-continuous trajectory consistent with the requests of the sampler. This is done by representing the trajectory of the controlled vehicle (and not the controlled vehicle itself) as a state-space dynamic system evolving from time step \( k \) to time step \( k + 1 \). This evolution is controlled (forced) in order to impose the distance requested to be covered according to the estimates of the sampler. Note that the profiler is not in our case a car-following model, unlike other time-continuous differential-equation-based approaches (e.g. Tordeux et al., 2010). Within the profiler some of the variables introduced in the previous section are redefined: \( t_0 \) is the time instant corresponding to the sampler discrete time \( k \); \( \Delta T \) is the duration of the time step defined in the discrete-time approach adopted for the sampler; as a consequence, the profiler is in charge of producing a continue trajectory in the time interval \( [t_0, t_0 + \Delta T] \) and the instant \( t_0 + \Delta T \) coincides with time \( k + 1 \) of the sampler; \( u = \Delta s_s(k, k + 1) \) is the driven distance the profiler should impose from \( t_0 \) to \( t_0 + \Delta T \), as supplied by the sampler; it is the instantaneous step-solicitation; \( \Delta s_s \) is the distance actually covered by the vehicle in the time interval \( [t_0, t_0 + \Delta T] \); it could prove different from the requested one \( (u) \); \( \Delta v \) is the variation in speed actually attained in the time interval \( [t_0, t_0 + \Delta T] \); \( x_1(t) \) is the instantaneous acceleration at a generic time instant \( t \in [t_0, t_0 + \Delta T] \); it represents our first state variable; \( x_2(t) \) is the instantaneous jerk at a generic time instant \( t \in [t_0, t_0 + \Delta T] \); it represents our second state variable.

We assume that the trajectory of the vehicle can be described as a (linear, time-invariant in \( \Delta T \)) dynamic system according to the following model:

\[
\dot{x}_1(t) = \hat{e}_1 x_1(t) + \hat{e}_2 x_2(t) + \hat{f} u \\
\dot{x}_2(t) = e_1 x_1(t) + e_2 x_2(t) + f u
\]
Of course, the state variables have to respect the physical consistency between jerk and acceleration:
\[ \dot{x}_1(t) = x_2(t) \]

One of the possible solutions that ensures consistency is:
\[ \dot{\epsilon}_1 = 0 \quad \dot{\epsilon}_2 = 1 \quad \dot{f} = 0 \]

This is equivalent to rewriting the dynamic system in the form:
\[ \dot{x}_1(t) = x_2(t) \]
\[ \dot{x}_2(t) = \epsilon_1 x_1(t) + \epsilon_2 x_2(t) + f u \]

Or, by using a matrix notation:
\[ \dot{x}(t) = A x(t) + c u \]

where
\[ A = \begin{bmatrix} 0 & 1 \\ \epsilon_1 & \epsilon_2 \end{bmatrix} \quad c = \begin{bmatrix} 0 \\ f \end{bmatrix} \quad x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} \]

As usual for dynamic systems, it is important that the model parameters ensure the stability of the system. This depends on the eigenvalues of matrix \( A \):
\[ \lambda_1 = \frac{1}{2} \left( \epsilon_2 - \sqrt{4 \epsilon_1 + \epsilon_2^2} \right) \]
\[ \lambda_2 = \frac{1}{2} \left( \epsilon_2 + \sqrt{4 \epsilon_1 + \epsilon_2^2} \right) \]

In particular, the stability is ensured if the eigenvalues are real, distinct and negative; this enables the fixed point (regime) to be viewed as a so-called sink-node (in the gradient field). This is ensured if:
\[ \epsilon_1 < 0 \quad \epsilon_2 < 0 \quad \epsilon_2^2 > -4 \epsilon_1 \quad (10) \]

Eigenvalues can be related to the so-called time constants \( (\tau_1, \tau_2) \) that can be used to compute (with excellent approximation) the so-called settling time \( (t_{s1}, t_{s2}) \), defined as the time elapsing from the application of an instantaneous step input to the time at which the state variable has entered a \( \delta \) bound around the final (fixed-point) value. In formal terms:
\[ \tau_i = \frac{1}{\lambda_i} \quad \forall i \in \{1, 2\} \]
\[ t_{s_i} \cong 2 \tau_i \ln \left( \frac{1}{\delta} \right) \quad \forall i \in \{1, 2\} \]

This means that if, for instance, \( \delta = 10\% \) then \( t_{s_1} \cong 4.6 \tau_1 \) and if \( \delta = 5\% \) then \( t_{s_1} \cong 6 \tau_1 \)

In practice \( t_{s_i} \cong \alpha \tau_i \quad \forall i \in \{1, 2\} \), where \( \alpha \in \{4.5, 6\} \) for an attained system response varying in the range [90%,95%] of the final response.

On the other hand, the settling times can be imposed to be equal to a predefined part of the whole transition period \( \Delta T \). This can be set as a function of two parameters \( (\omega, \psi) \):
\[ t_{s1} = \omega \Delta T \quad \omega \in [0, 1] \]
\[ t_{s2} = \psi \Delta T \quad \psi \in [0, 1] \]

Finally, it results that:
\[ \lambda_1 = -\frac{1}{\tau_1} = -\frac{1}{t_{s1}/\alpha} = -\frac{\alpha}{\omega \Delta T} = \frac{1}{2} \left( \epsilon_2 - \sqrt{4 \epsilon_1 + \epsilon_2^2} \right) \]
\[ \lambda_2 = -\frac{1}{\tau_2} = -\frac{1}{t_{s2}/\alpha} = -\frac{\alpha}{\psi \Delta T} = \frac{1}{2} \left( \epsilon_2 + \sqrt{4 \epsilon_1 + \epsilon_2^2} \right) \]

It can be easily verified that previous equations are satisfied by:
\[ e_1 = -\frac{\alpha^2}{\Delta T^2 \psi \omega} \]
\[ e_2 = -\frac{\alpha (\psi + \omega)}{\Delta T \psi \omega} \quad (11) \]
By using the previous values of $e_1$ and $e_2$ in Eq. (10), it can be noted for stability conditions that:

$e_1 < 0$ and $e_2 < 0$, because $\psi \omega > 0$ and $\psi + \omega > 0$, being $\psi > 0$ and $\omega > 0$;

$e_2^2 > 4e_1$, is ensured if $(\psi - \omega)^2 > 0$ that is if $\psi \neq \omega$; this can be verified by substituting Eqs. 11 and 12 in $e_2^2 + 4e_1$, thus obtaining $e_2^2 + 4e_1 = e^2(\psi - \omega)^2$, where $e^2 = x^2/(\Delta T^2 \psi^2 \omega^2)$.

It results that the stability of the trajectory is ensured by very mild assumptions on the parameters ($\psi$ and $\omega$) governing the settling time of the system. In practice, they just have to be admissible (values from 0 to 1) and distinct.

Stability is evaluated around the fixed point (the regime). Regime values for the state variables can be evaluated as:

$$0 = Ax' + cu - x' = -A^{-1}cu$$

where

$$-A^{-1}cu = \begin{bmatrix} -f/e_1 \\ 0 \end{bmatrix}$$

It results that the jerk assumes at the fixed point a null value, while the acceleration is finite and assumes a value that depends on the step ($u$) requested by the sampler:

$$x'_1 = -f/e_1 u \quad x'_2 = 0$$

Given the imposed stability of the system and assuming that the settling times have been properly set by means of parameters $\psi$ and $\omega$, the status of the system at time $t_0 + \Delta T$ can be reasonably considered as being attained:

$$x_1(t_0 + \Delta T) = x'_1 = -f/e_1 u$$

$$x_2(t_0 + \Delta T) = x'_2 = 0$$

(13)

(14)

Now, consider the variation of speed ($\Delta v$) in the time interval $[t_0, t_0 + \Delta T]$:

$$\Delta v = \int_{t_0}^{t_0 + \Delta T} a(t) dt = \int_{t_0}^{t_0 + \Delta T} x_1(t) dt = \int_{t_0}^{t_0 + \Delta T} \left[ \frac{1}{2} (x_2(t) - e_2 x_2(t) - fu) \right] dt = \int_{t_0}^{t_0 + \Delta T} \left[ \frac{1}{2} (x_2(t) - e_2 x_1(t) - fu) \right] dt$$

$$= \frac{1}{e_1} [x_2(t_0 + \Delta T) - x_2(t_0)] - \frac{e_2}{e_1} \left[ x_1(t_0 + \Delta T) - x_1(t_0) \right] - \frac{f}{e_1} u \Delta T$$

From Eq. (14) it is also evident that $x_2(t_0) = 0$, because it represents the regime status of a previous time step: $t_0 = (t_0 - \Delta T) + \Delta T$

Then:

$$\Delta v = -\frac{e_2}{e_1} [x'_1 - x_1(t_0)] - \frac{f}{e_1} u \Delta T = -\frac{e_2}{e_1} [x'_1 - x_1(t_0)] + x'_1 \Delta T$$

(15)

Eq. (15) above, once $x'_1$ is known, allows the computation of the variation in speed imposed by the profiler when the step requested by the sampler is imposed. Now consider the actual variation of space ($\Delta s$) in the time interval $[t_0, t_0 + \Delta T]$:

$$\Delta s = \int_{t_0}^{t_0 + \Delta T} v(t) dt$$

To compute it, the expression of the speed has to be derived:

$$v(t) = v(t_0) + \int_{t_0}^{t} a(z) dz = v(t_0) + \int_{t_0}^{t} x_1(z) dz = v(t_0) + \int_{t_0}^{t} \left[ \frac{1}{e_1} (x_2(z) - e_2 x_2(z) - fu) \right] dz$$

$$= v(t_0) + \int_{t_0}^{t} \left[ \frac{1}{e_1} (x_2(z) - e_2 x_1(z) - fu) \right] dz = v(t_0) + \frac{1}{e_1} (x_2(t) - x_2(t_0)) - \frac{e_2}{e_1} (x_1(t) - x_1(t_0)) - \frac{f}{e_1} u(t - t_0)$$

Therefore:

$$\Delta s = \int_{t_0}^{t_0 + \Delta T} v(t) dt = \int_{t_0}^{t_0 + \Delta T} \left[ v(t_0) + \frac{1}{e_1} x_2(t) - \frac{e_2}{e_1} (x_1(t) - x_1(t_0)) - \frac{f}{e_1} u(t - t_0) \right] dt$$

$$= \int_{t_0}^{t_0 + \Delta T} \left[ v(t_0) + \frac{1}{e_1} x_2(t) - \frac{e_2}{e_1} x_1(t) + \frac{e_2}{e_1} x_1(t_0) - \frac{f}{e_1} u(t - t_0) \right] dt$$

$$= v(t_0) \Delta T + \frac{1}{e_1} \int_{t_0}^{t_0 + \Delta T} x_1(t) dt + e_2 \int_{t_0}^{t_0 + \Delta T} x_1(t) dt + e_2 x_1(t_0) \Delta T - \frac{f}{e_1} u \int_{t_0}^{t_0 + \Delta T} (t - t_0) dt$$

$$= v(t_0) \Delta T + \frac{1}{e_1} x'_1 + \frac{e_2}{e_1} x_1(t_0) - \frac{e_2}{e_1} x_1(t_0) + \frac{e_2}{e_1} x_1(t_0) \Delta T + x'_1 \Delta T^2 \frac{1}{2}$$
Substituting the expression of $\Delta v$ (Eq. (15)), it is possible to obtain:

$$\Delta s = v(t_0)\Delta T + \frac{e_2}{e_1}x_1(t_0)\Delta T - \left(\frac{e_2}{e_1} + e_1\right)x_1(t_0) + x_1'(t_0)\left(\frac{\Delta T^2}{2} - \frac{e_2}{e_1}\Delta T + \frac{(e_2^2 + e_1^2)}{e_1^2}\right)$$  \hspace{1cm} (16)

Solving with respect to $x_1'$ and considering that the aim of the profiler is to force the driven distance to the step requested by the sampler ($\Delta s = u$):

$$x_1' = \frac{2\frac{e_2}{e_1}(u - v(t_0)\Delta T) + e_2^2x_1(t_0) + e_1x_1(t_0)(1 - e_2\Delta T)}{2e_1^2 + e_2(2 - 2e_1\Delta T + e_2\Delta T^2)}$$  \hspace{1cm} (17)

All the calculations stated above are applied by the profiler in the following way.

Step 1. Fix values for settling time parameters $\psi$ and $\omega$, use admissible values ($\in [0,1]$), ensure stability ($\psi \neq \omega$) and, if appropriate, try to have mild transitions (e.g. $\theta$; $\omega > 0.70$).

Step 2. Compute parameters $e_1$ and $e_2$ from Eqs. (11) and (12) and compute the regime acceleration ($x_1'$) by using Eq. (17) that ensures the driven distance equals that requested by the sampler ($\Delta s = u$).

Step 3. Check the resulting regime acceleration ($x_1'$). Only if the resulting value is inadmissible and/or judged to be inappropriate (e.g. $|x_1'| > 2 m/s^2$) fix an appropriate value for $x_1$ and compute by Eq. (16) the driven distance ($\Delta s$) the profiler will actually set (different from that $- u - \Delta s$ requested by the sampler).

Step 4. Compute parameter $f$ by solving from Eq. (13): $f = -\frac{x_1e_1}{\Delta s}$

Step 5. Run the profiler as a continuous state-space dynamic system from time $t_0$ to time $t_0 + \Delta T$; the resulting trajectory is the output of the profiler; in the overall multilayer architecture this is the input for the tutor.

The output is formally expressed in terms of acceleration and jerk (state variables); if required, other variables describing the trajectory (e.g. speed, position, etc.) can be easily derived from the state variables by using standard integration techniques. Note that the main purpose of step 3 above is to ensure the development of a robust algorithm. In theory, the step requested by the sampler imitates a human-like behaviour, that is it should be intrinsically consistent (for example) with admissible accelerations. If this is not the case, it is due to local errors in estimating the human-like spacing to be imposed, for instance due to an instantaneous malfunctioning of the on-board sensors. The role of step 3 is to avoid the propagation of such an error over the controlled vehicle trajectory.

It is worth noting that in all the above considerations, there are still some degrees of freedom in fixing some of the parameters of the profiler. In particular, the values of the parameters ($\psi$ and $\omega$) can be somehow arbitrarily chosen. In the numerical applications related to this paper, discussed in Section 5 below, they were set in order to obtain long settling times and support a smooth dynamics in terms of acceleration and jerking. Of course, smoothness is here intended as a pleasing transition from the starting point to the final point requested by the sampler. Pleasing transitions are favoured not only by fairly high values of $\psi$ and $\omega$ but also by the human likeness of the profiler. Higher values for parameters $\psi$ and $\omega$ could not only promote comfortable cruising but also a reduction in consumption and/or pollution. In future works, we will deal with the role of $\psi$ and $\omega$ as well as making some formal in-depth considerations about their suitability for formal optimisation purposes.

5. Results

In order to test the proposed approach, the trajectories collected by means of the instrumented vehicle (see Section 2) were used. For each of these, we first carried out the calibration of the sampler, performed on a small part of the trajectory. The sampler and the profiler were then tested in running mode: the simulated trajectories were compared with the experimental ones (in their parts not used for calibration) in order to check their ability to reproduce observations.

The overall framework was implemented by using Simulink/Matlab. A simplified tutor module was also included, essentially with the aim of overriding the trajectories provided by the profiler if unsafe. The simplified tutor implemented is not discussed here in detail. In practice, it is based on the concept of safe emergency braking for both the leader and the follower. Safety is tested by the tutor at a very high frequency (the maximum technically allowed by the sensors and by the on-board trajectory actuators), i.e. 10 Hz. Due to both the high control frequency and the magnitude of the actual cruising speeds, any lag between detection and actuation can be considered negligible and the admissibility of the trajectory depends on the different maximum decelerations likely to be applied by the leading and following vehicle in emergency conditions. In the results shown below, the maximum deceleration of the follower is assumed to be 70% of that of the leader. Of course, more complex (and perhaps appropriate) approaches can be incorporated in the tutor. In future research, direct control of spacing also in the tutor may be tested, constraining it to a safe range; such an approach could be implemented parallel to the modelling sequence sampler-profiler and could be based, for instance, on the approach by Bageshwar et al. (2004) or Martinez and Canudas-De-Wit (2007). In any event, the smaller of the position increments suggested by the tutor or by the sampler+profiler is applied by the performer.
Implementation and application of the performer (which is the detailed representation of the internal vehicle dynamics and of vehicle interaction with the road) lie outside the scope of this paper. In practice, the hypothesis is made that a performer is available and that it allows for the regulation of the actual vehicle trajectory around the reference one (produced by the profiler). It is worth noting that the characteristics of the profiler (and the tutor) we developed are compatible with commercial simulation tools that can play the role of performer. As an example, refer to CarSim (CarSim 8: Mechanical Simulation Corporation, website) that can also be integrated within the adopted Simulink/Matlab environment. The results shown in the following come from the application of the sampler, profiler and (simplified) tutor.

In our numerical experiments, prior to calibrating the sampler, Pearson’s tests were carried out to check the (linear) correlation between the independent variables of Eq. (1) and to exclude collinearity problems in parameter estimation. The tests evidenced a mild correlation among some variables, which should not give rise to practical problems. Table 2 below shows the Pearson’s coefficients for the case of the observed values of the current trajectory to which we refer in this section when showing the results.

Table 3 presents some results related to the calibration of the sampler. Estimation values and statistics are shown for the current trajectory (characterised by a good variety of the observed speed and spacing) but also the min, max, mean and standard deviation of the parameters over all estimated trajectories are shown, as well as the duration in seconds of the trajectory required by the calibration algorithm in the current trajectory, in the worst and best cases. Finally, values for the constraints in Eq. (8) are shown for the current case, as well as the min and max values over all calibrations.

For the current trajectory, Fig. 2 shows the evolution over time of the parameters during calibration. The chart is depicted by extending the calibration to the entire duration of the observed trajectory, even if it actually stops after 67.20 s (see Table 3). It seems evident from the figure that after a few seconds the estimation of the parameters is stable. Some experiments were made in order to test the effect of calibrating starting at different times (e.g. at time 30 or 45 or 60 instead of time 0). Results of these tests (not reported here) showed that the same values are obtained, as can also be argued from Fig. 2.

<table>
<thead>
<tr>
<th>Table 2</th>
<th>Pearson’s test of correlation between the independent variables of the sampler.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Δx</td>
</tr>
<tr>
<td>Δx</td>
<td>1</td>
</tr>
<tr>
<td>Δv</td>
<td>0.1088</td>
</tr>
<tr>
<td>ρ_1</td>
<td>0.3191</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3</th>
<th>Calibration result.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Current trajectory</td>
</tr>
<tr>
<td></td>
<td>Min</td>
</tr>
<tr>
<td>β₀</td>
<td>0.0762</td>
</tr>
<tr>
<td>β₁</td>
<td>0.9725</td>
</tr>
<tr>
<td>β₂</td>
<td>0.5382</td>
</tr>
<tr>
<td>β₃</td>
<td>0.0265</td>
</tr>
<tr>
<td>Calibration time (s)</td>
<td>67.20</td>
</tr>
<tr>
<td>β₀(1-β₁)</td>
<td>2.7709</td>
</tr>
<tr>
<td>β₁(1-β₁)</td>
<td>0.9636</td>
</tr>
</tbody>
</table>

![Fig. 2. Evolution of the sampler parameter during calibration.](image)
The results in terms of overall performance of the four-layer model are shown below; reference is made to the current trajectory but also the envelope of the worst and best performance is reported. In agreement with the approach proposed in other works based on experimental data collection (e.g. Marsden et al., 2001), comparison is made between the trajectory which would have been imposed by the ACC and that exhibited by the driver during manual driving. Fig. 3 shows the ability of the sampler to reproduce the observed data. For the first 67 s the figure shows only the dark line representing the observed spacing (in metres). This part of the trajectory is required by the calibration algorithm (learning-mode phase) in order to estimate the parameters of the linear car-following model. After 67 s the dark curve plays the role of reference data, plotted until 900 s (15 min) together with estimated data (grey line). From the dark line in Fig. 3, it can be seen that the current trajectory is representative of dense traffic conditions, the spacing between the vehicles ranges from about 12 to about 28 metres and the trajectory can be considered fairly variegated. The dark line in Fig. 4 represents the observed speed (m/s) of the controlled vehicle. It is worth noting that as the observed speed increases (Fig. 4) the spacing also increases (Fig. 3), as expected. The dark line in Fig. 4 ranges from 40 km/h (12 m/s) to 110 km/h (30 m/s) and confirms a variegated trajectory and the presence of dense traffic conditions.

Data are compared at the frequency of 1 Hz. Once the running mode starts (after 67 s), system evolution is based only on the sampler estimates and is not refreshed by observed data, that are used only for comparison purposes. Of course, in order to run the sampler also the trajectory of the leading vehicle is needed; this was obtained from observations and represents a boundary condition for the sampler. The sampler-estimated spacing follows the observed one quite closely, confirming the human likeness of the proposed approach. From the sampler-estimated spacing the speed of the controlled vehicle can be easily computed; this is depicted by the grey line in Fig. 4. The sampler-estimated speed agrees well with the observed speed. The reader should be aware that even if a satisfactory agreement with the observed spacing produces an almost perfect agreement of the speed, the reciprocal is not true. If the sampler were built to reproduce (for instance) speeds and not spacing, any error in this reproduction would have been recovered in terms of speed but not in terms of spacing due to integral error phenomena.

![Fig. 3. Sampler performance: spacing.](image1)

![Fig. 4. Sampler performance: speed.](image2)
Fig. 5 extends the comparison to the results of the overall modelling platform (including the profiler and the tutor) but is restricted (for graphical representation) within a shorter interval (a little more than 2 min, from 490 to 630 s) where the difference between the observed and estimated data is higher. In Fig. 5, the observed data are represented by an interpolating continuous (dark) line just in order to enhance the clarity of the representation, but they have to be intended as plotted at a frequency of about 1 Hz, similarly to the sampler output (grey points). The profiler, instead, produces continuous data and the tutor produces data at quite a high frequency. Hence the grey curve that represents the result of the overall framework is continuous. The most interesting (and positive) phenomenon is that the sampler is able to recover estimation errors (such as after points 520 or 550 in Fig. 5).

It should be noted that the profiler is built in order to fit the target spacing estimated by the sampler. The almost perfect overlapping of the grey points and the grey line (practically indistinguishable) on the left-hand side of Fig. 5 shows that the profiler was appropriately constructed and that it works exactly as required (it is also evident that the tutor does not need to actually operate in this time interval).

The difference between the observed data and the trajectory resulting from the application of the proposed ACC can also be shown by means of the cumulative error distributions, as depicted in Figs. 6 and 7. The left-hand side of Fig. 6 shows that

![Fig. 5. Performance of the overall modelling framework: spacing and speed (details).](image)

![Fig. 6. Cumulative error, sampler vs. observed data: spacing and speed.](image)

![Fig. 7. Cumulative error, profiler vs. sampler: spacing and speed.](image)
the error made by the sampler in estimating the observed spacing is never greater than 77%: it is less than 20% in 60% of cases and in 40% of cases is less than 10%. Similarly, the right-hand side of Fig. 6 shows that the error in terms of speed reproduction is never greater than 15%, is less than 10% in 95% of cases, is less than 5% in about 80% of cases and in many cases is negligible. The different perception of errors in terms of spacing and speed should also be considered, evaluated in absolute terms. For instance, a 40% error with respect to a real spacing of 20 m corresponds to an absolute error of 8 metres; the same with respect to a speed of 20 m/s (72 km/h) corresponds to an absolute error of about 30 km/h. Whether a difference in spacing of 8 m with respect to the human-like one is more or less perceived by the driver (and perceived as more or less acceptable) than a difference in speed of 30 km/h should be evaluated.

Fig. 7 shows the agreement of the continuous trajectory generated by the profiler with the discrete points estimated by the sampler; it is computed on the whole running mode part of the trajectory and not only on the time interval shown in Fig. 5. The agreement is almost perfect for spacing and very good for speed (in almost all cases the discrepancy is less than 10% and in 80% of cases less than 3%).

As a conclusive remark on comparison, the proposed framework is fairly satisfactory with respect to human likeness and, given that it can be easily calibrated in real time and on demand for different drivers, different contexts and/or different driving sessions, it can be judged to be fully adaptive.

### 6. Conclusions

Driving data were observed in the field and collected by means of an instrumented vehicle. A fully-adaptive cruise control system was developed, ably reproducing human driving also in dense traffic conditions. The control logic behind the ACC was designed to be fully-adaptive, in the sense that it can be easily adapted to different drivers and/or different driving contexts, it can be calibrated on demand by just driving for a few minutes and, finally, it reproduces the behaviour the driver would have had (human likeness). The possibility of calibrating and applying the developed framework in real time and on demand is strongly associated with the linear architecture of the embedded car-following model. Despite the simple linear approach the results are very satisfactory, thus confirming the previous findings of the authors, as well as some evidence in the literature, where simpler models outperform (especially in validation) more complex ones.

The developed framework was structured into four layers: the sampler, the profiler, the tutor and the performer. The sampler is responsible for establishing the ACC control logic, the profiler is intended to transform the control logic into a continuous kinematic profile to be followed by the controlled vehicle, the tutor is in charge of ensuring the ultimate safety of the vehicle’s kinematics, and the performer is in charge of applying the reference trajectory by using the vehicle’s on-board actuators (e.g. brakes, throttle, etc.) and by interacting with the internal dynamics of the vehicle and with the road. The proposed modelling framework was implemented in a Simulink/Matlab environment in order to release a standard implementation suitable in the future for possible fast-prototyping.

The core of the human-like model is the sampler. It assumes the time series of spacing (intravehicle distance) as a variable to be estimated. Two main reasons account for this choice: first of all, spacing is a very natural choice for ACC applications, given that it is one of the measurements typically detected by the on-board sensors of an ACC-enabled vehicle; moreover, it is well known that very good reproduction of, say, speed evolution over time does not ensure a good enough reproduction of spacing. On the contrary, reproducing spacing also ensures very good accommodation of speed evolution over time. This is due to the integral of the error: if vehicle speed is erroneously reproduced for some time and then the accuracy is recovered, the error in terms of position increases during the period of inaccuracy, then stops increasing but is not recovered. By contrast, errors in terms of spacing can be recovered if the spacing is directly controlled and an error in spacing affects only locally the speed computation.

In the proposed ACC-oriented car-following formulation the stimulus towards spacing adjustment not only depends on the relative speed difference but also on the consistency between the current spacing and the cruising speed. This effect is somehow embedded in the value of the parameters of the proposed linear model. In particular, future studies could investigate whether an appropriate combination of linear model parameters could represent a sort of footprint of the driver’s driving style.

Implementation of the sampler requires that some independent variables be known at each simulation time step (see Section 2). However, whether these variables are directly measured, or indirectly estimated by using other measurements or numerical techniques, depends on technological aspects of the implementation. Of course, any of the previous (measured or computed) data can be affected by errors and all employed measures should be subject to error-smoothing/recovering procedures, as has been done (even if not explicitly described) in the analyses reported in this paper. Definition of the minimum accuracy tolerances of these measurements in order to deal with useful data and/or development of automatic real-time smoothing/recovering procedures, preferably based on Kalman filtering and on a fusion of different sources of measurements/sensors, are tasks for further research.

The role of the profiler is also crucial for the practical application of the proposed control logic. The profiler produces a continuous reference trajectory, fully defined in terms of complete kinematic description, ensuring that the points estimated by the (human-like) logic of the sampler are fitted.

Even if the proposed modelling framework could well be adapted to a wider range of applications, including traffic simulation, it is developed and presented herein with specific reference to the case of ACC. However, ACC applications require...
that some strict requirements in terms of real-time applicability need to be satisfied and a major part of our research effort was devoted to precisely that. The authors' perception is that preparing answers to such real-time issues is also relevant to non-ACC applications, given that an increasing number of traffic problems need to be solved nowadays in real-time (or fast-time) contexts. However, extension of the approach to non-ACC oriented applications is another task for future research.

Acknowledgements

The authors would like to thank Anita Fiorentino, Giovanni Abbattista and Stefano Scala of the FGA (Fiat Group Automobile) Laboratories in Pomigliano D'Arco (Naples, Italy) for their precious contribution in the early development of the framework presented here. The authors would also like to thank the anonymous reviewers of this paper: their comments brought about significant improvements in its overall quality.

References


