Multi Model Adaptive Control for CACC applications

Francisco Navas, Vicente Milanés, Carlos Flores and Fawzi Nashashibi

***Abstract*—This paper proposes a multi-model adaptive control (MMAC) algorithm based on Youla-Kucera (YK) theory to deal with heterogeneity in cooperative adaptive cruise control (CACC) systems. The main idea of MMAC is to choose the plant in a predefined set that best approximates the system dynamics, applying the corresponding predesigned controller. A set of linear plants describing different vehicle dynamics is defined. Different CACC controllers are designed depending on these linear plants. Simulation and experimental results prove how MMAC determines the closest plant in the set, choosing the CACC system able to ensure string stability.**

***Index Terms*—Heterogeneous string, Homogeneous string, Cooperative Adaptive Cruise Control, Youla-Kucera parameterization, Multi-Model Adaptive Control.**

I. Introduction

Traffic demand has increased significantly over recent years. According to the European Commission, road transport constitutes 82.4% of the whole passengers' transport in the European Union (EU) [1]. This leads to traffic congestion problems, which if not solved, could have a cost up to 213 billion euros in the next 10 years [2].

To overcome the problem of this huge rising traffic volume, intelligent transportation systems (ITS) technology provides efficient solutions without additional infrastructure cost. An example of such a system is the commercially available adaptive cruise control (ACC) [3], that can maintain a minimum safety time gap with the vehicle ahead. Recently, research focuses on the cooperative version of this system, so-called cooperative ACC (CACC) [4]. Vehicle-to-vehicle (V2V) communication is added to ACC, providing a lower minimum safety time gap, which could increment highway capacity in almost a 100 % [5].

The main criteria to evaluate a CACC system is the minimum time gap achievable that guarantees string stability. The latter is defined as the attenuation of disturbances along the string of vehicles [6]. As it needs to be ensured to avoid collisions, improve highway capacity and reduce fuel consumption, string stability analysis for a string of vehicles has been a hot topic in research.

When vehicles within the string have identical dynamics (i.e. homogeneous string), disturbances attenuates/amplifies uniformly along the string. String stability for homogeneous string of vehicles has been widely studied [6], being ensured for any positive time gap value if no delay is present in the communication link between vehicles. This statement is validated through experimental results [7] [8].

On the contrary, when vehicles in the string have not identical dynamics, disturbances do not attenuate/amplify in the same way downstream; e.g. a vehicle with slower dynamics will have difficulties to follow a vehicle with faster dynamics. This was first noticed during the GCDC 2011, where ATeam remarked how string stability is explicitly affected by preceding vehicle's dynamics–the minimum time gap achievable goes from $0.6s$ to $1.5s$ [9]. Guidelines are given in [10] [11] about how to change variables of a CACC controller depending on whether slower or faster dynamics vehicle is in front of the ego-vehicle.

String stability for a heterogeneous string of vehicles has been mainly addressed in two different ways: Robust control– a fixed controller able to work under bounded uncertainties in the plant. Dynamics difference between ego and preceding vehicles is seen as uncertainty in the platoon model; Adaptive control– controller changes with the unknown uncertainties in the system, e.g. control is modified depending on ego and preceding dynamics. Adaptive control is needed when robust control is not able to offer decent behaviour due to all the parameters changes that can occur in the system.

Related to robust control, [12] added some local conditions for heterogeneity to a robust controller designed for a homogeneous string of vehicles. Dynamics of the different agents within the strings need to be known to validate if the required local conditions are fulfilled. [13] proposed a controller design based on linear matrix inequality (LMI) to ensure individual stability, while an H_∞ control algorithm considers disturbances that affect string stability. Again some conditions based on the components of the string should be validated. A H_∞ robust control design was presented in [14] to overcome uncertainty between model connections. The range of model parameters needs to be known apriori. [15] presented a H_∞ control method which can guarantee both individual and string stabilities to parameters uncertainties and communication delay. Heterogeneity is reflected by different vehicle mass and time lag of powertrain dynamics. Wind gust and road slopes are also considered as external disturbances.

Related to adaptive control, almost all the existing work takes as baseline controller a CACC feedforward/feedback proportional-derivative (PD) controller. This type of controller has been extensively used due its simplicity and performance. [16] designed a new feedforward controller for a heterogeneous string of vehicles to minimize spacing error and make independent string stability from preceding vehicles dynamics. Models of ego and preceding vehicles are included in the feedforward, so string stability remains as in a homogeneous string of vehicles. However, there is not mention of how to obtain the dynamics of both vehicles in order to adapt the controller. More recently, one-vehicle look-ahead topology with model reference adaptive control is considered in [17]. They augmented a normal feedforward/feedback CACC system working in the homogeneous case with an adaptive

F. Navas is with AKKA Research Department (France); but work was developed with RITS Team at INRIA Paris, France. {francisco-martin.navas-matos@akka.eu}.

term compensating unknown driveline vehicle dynamics. The idea is the same, but dynamics estimation for subsequent adaptation is included. Preceding vehicle dynamics estimation is also developed in [18] with a particle filter algorithm. [19] proposes an adaptive optimal control solution able to learn the optimal feedback loop based on online data. However, vehicle dynamics are considered as a first-order function with differences of 0.04s in their response time, which seems unreal to simulate heterogeneity. Finally, an adaptive robust control solution is proposed in [20]. It deals with different information delays, communication topologies, ego-vehicle uncertainty and external disturbances. However, dynamics differences between vehicles are expressed as a maximum speed, distance and different controller parameters, leading to simplified vehicle models that again could not represent the real difference in a heterogeneous string of vehicles.

Adaptive control based on identification covers a greater dynamic range of vehicle dynamics than robust control. With the idea of avoiding an identification process that could slow down the control loop, a set of linear plants that describes a range of vehicle dynamics can be defined. Plants should be general enough to cover a wide range of possible dynamics, properly emulating the intended string heterogeneity. Likewise, a string stability analysis should be carried out in order to define application limits, which is missing in the literature.

Controller reconfiguration depending on a set of linear plants it is called in the literature Multi Model Adaptive Control (MMAC) [21]. Different applications of this methodology are found in the literature: clinical stroke rehabilitation [22], attitude tracking control for a spacecraft [23] or large-scale photovoltaic plants [24]. The idea of MMAC is a supervisor who chooses the proper controller among pre-designed candidates controllers once more information is known about the plant. Pre-designed controllers are conceived with the set of linear plants. MMAC is able to determine the closest plant in the set, switching to its corresponding controller to maintain the desired performance. [25] proposed an indirect adaptive control based on the identification of linear plants by using the v-gap metric. As the metric is difficult to obtain in real time, [26] proposed a similar approach but using model unfalsification: If a model and a controller are unable to reproduce the observed behaviour in closed-loop, then the set plant-controller is not the correct representation of the system. However, noise correlation problems are affecting system performance. The noise correlation problem is later solved in [27] by using the dual Youla-Kucera (YK) parameterization.

Here the MMAC algorithm based on YK in [27] is used together with the widely used CACC feedforward/feedback control structure. A set of linear plants describing different vehicle dynamics is defined representing heterogeneous strings for MMAC validation. Stability analysis is provided, clearly defining application limits. MMAC should estimate the closest plant in this set to ego and preceding vehicles, so the predesigned CACC controller able to ensure string stability is chosen. An outline of this algorithm is in Fig. 1. Encouraging results are obtained both in simulation and in real tests.

In brief, main contributions in this paper are the following:

- Adaptive control solution for CACC heterogeneous/homogeneous string of vehicles without need of identification algorithm.
- First and second order models are employed with wider parameter variations in order to properly emulate string heterogeneity.
- Application limits are well-known, and applicable to a wider range just by setting more plants in the set.
- String stability is ensured for a a specific dynamics range in both ego and preceding vehicles.
- YK-based MMAC approach has been adapted to CACC applications and applied for the first time experimentally.
- Dynamics estimation results faster than other estimation/identification processes in the literature [17].
- Experimental results belong to the very first heterogeneous CACC algorithm results in the literature together with [28].

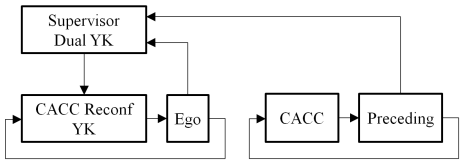


Fig. 1. Outline MMAC for CACC applications.

The paper is structured as follows. Section 2 gives some background for a good understanding of the paper. Section 3 describes MMAC for a general set of linear plants and controllers. The same is modified in Section 4 for its application to CACC systems. Simulation and experimental results are analysed in Sections 5 and 6. Finally, some concluding remarks are given in Section 7.

II. Preliminaries

This section describes some basic notation, which will be used extensively in the sequel. MMAC relies on the YK parameterization [29] for controller reconfiguration and supervision.

A general description of a set of linear plants and its corresponding predesigned controllers is given, to later apply the doubly coprime factorization used by the YK parameterization.

A. System description and assumptions

Let's consider a set of nominal plants represented as $\{G\} = \{\mathbf{G_0}..., \mathbf{G_i}..., \mathbf{G_n}\}$, describing different dynamics of a system. $\mathbf{G_i}$ denotes the i^{th} linear, time-invariant, discrete plant mapping input signals $u_i \in \mathbb{R}^m$ in output signals $y_i \in \mathbb{R}^p$. For each of these plants a discrete feedback controller $\mathbf{K_i}$ mapping error signals e_i in input signals u_i is designed such that the closed-loop (CL) behavior of the system is the desired one. Thus, a set of candidate controllers is defined relying on the nominal set of plants $\{K\} = \{\mathbf{K_0}..., \mathbf{K_i}, ..., \mathbf{K_n}\}$. Plants and controllers are represented in state space as:

$$\mathbf{G_i} = \left[\begin{array}{c|c} \mathbf{A_i} & \mathbf{B_i} \\ \hline \mathbf{C_i} & \mathbf{D_i} \end{array}\right]; \mathbf{K_i} = \left[\begin{array}{c|c} \mathbf{A_i^c} & \mathbf{B_i^c} \\ \hline \mathbf{C_i^c} & \mathbf{D_i^c} \end{array}\right] \quad (1)$$

Time variations in the real plant $\mathbf{G_{real}}$ are considered slow compared to input-output dynamics. The longitudinal low level

of the vehicle is considered to be properly designed, keeping the same LTI response with uncertainties as slope, wind gust, ... Nominal plants are defined, such that once a candidate controller is selected, it remains unchanged–i.e. the variations in $\mathbf{G_{real}}$ are smaller than those needed to change from one nominal plant to other within the set.

B. Doubly coprime factorization

$[\mathbf{G_i}, \mathbf{K_i}]$ needs to be factorized through the doubly coprime factorization [30] to apply later the basis of MMAC, that is, the YK parameterization.

$$\begin{bmatrix}\tilde{\mathbf{M}}_\mathbf{i} & \tilde{\mathbf{N}}_\mathbf{i}\end{bmatrix}\begin{bmatrix}\mathbf{X_{l,i}}\\ \mathbf{Y_{l,i}}\end{bmatrix} = \tilde{\mathbf{M}}_\mathbf{i}\mathbf{X_{l,i}} + \tilde{\mathbf{N}}_\mathbf{i}\mathbf{Y_{l,i}} = \mathbf{I} \tag{2}$$

These coprime factors should be such that G_i and K_i are:

$$\begin{aligned}\mathbf{G_i} &= \mathbf{N_i}\mathbf{M_i}^{-1} = \tilde{\mathbf{M}}_\mathbf{i}^{-1}\tilde{\mathbf{N}}_\mathbf{i}\\ \mathbf{K_i} &= \mathbf{U_i}\mathbf{V_i}^{-1} = \tilde{\mathbf{V}}_\mathbf{i}^{-1}\tilde{\mathbf{U}}_\mathbf{i}\end{aligned} \tag{3}$$

At the same time, these coprime factors $\mathbf{U_i} \in \mathbb{R}H_\infty^{mxp}$, $\tilde{\mathbf{U}}_\mathbf{i} \in \mathbb{R}H_\infty^{mxp}$, $\mathbf{V_i} \in \mathbb{R}H_\infty^{pxp}$, $\tilde{\mathbf{V}}_\mathbf{i} \in \mathbb{R}H_\infty^{pxp}$, $\mathbf{N_i} \in \mathbb{R}H_\infty^{pxm}$, $\tilde{\mathbf{N}}_\mathbf{i} \in \mathbb{R}H_\infty^{pxm}$, $\mathbf{M_i} \in \mathbb{R}H_\infty^{pxm}$ and $\tilde{\mathbf{M}}_\mathbf{i} \in \mathbb{R}H_\infty^{pxm}$ should satisfy the double Bézout's identity [31].

[32] allows to obtain coprime factors for $[\mathbf{G_i}, \mathbf{K_i}]$, when $\mathbf{G_i}$ and $\mathbf{K_i}$ are described in state-space form, Eq. (1). These factors satisfy (3) and the double Bézout's identity.

III. Multi Model Adaptive Control

Multi-model adaptive control (MMAC) is a supervisor switching among prespecified candidate controllers as new data about the plant is known, until one of these controllers finally remains unchanged.

It considers that the real plant $\mathbf{G_{real}}$ belongs to a set of nominal plants $\{G\} = \{\mathbf{G_0}..., \mathbf{G_i}..., \mathbf{G_n}\}$, or at least is close to one of them. Each of the nominal plants is associated to a controller to give a desired performance $\{K\} = \{\mathbf{K_0}..., \mathbf{K_i}, ..., \mathbf{K_n}\}$.

The supervisor is at a higher level, specifying which is the switching sequence γ that makes the system converge to the best controller for the unknown real plant $\mathbf{G_{real}}$. If $\mathbf{G_{real}}$ coincides with one of the nominal plants in the set $\{G\}$, a good candidate controller $\mathbf{K_i}$ is straightforward. Otherwise, the closest nominal plant in the set should be chosen, switching to the corresponding controller. Both supervisor and switching are related to Youla-Kucera (YK) parameterization.

YK parameterization is divided in two: Parameterization of all the controllers that stabilize a given plant; useful for performing switching between the candidate controllers [33]. And its dual formulation (also called Hansen scheme), all the plants stabilized by a given controller, which recasts CL identification into an open-loop-like problem [34]; employed within the supervisor to determine the switching sequence γ based on signals u and y.

Both theorems are briefly explained in their corresponding sections. Once the basis is known, subsection MMAC algorithm explains how they are used together, examining the set plant-controller $[\mathbf{G_i}, \mathbf{K_i}]$ and choosing the controller that is best able to fulfill performance requirements.

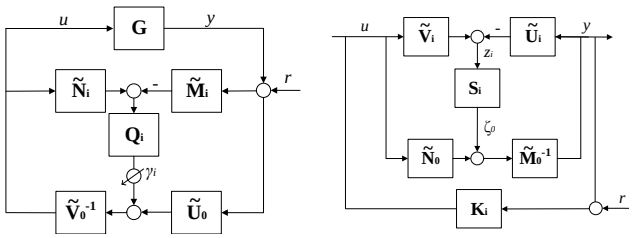


Fig. 2. (a) YK parameterization for controller reconfiguration and (b) Dual YK parameterization for CL identification.

A. YK parameterization. Switching

YK parameterization provides all stabilizing controllers for a given plant $\mathbf{G_i}$ within the set $\{G\}$ by interconnecting an initial controller $\mathbf{K_0}$ with $\mathbf{Q_i}$, called YK parameter. The initial controller could be any in the set $\{K\}$. Doubly coprime factors should be obtained, so the corresponding $\mathbf{Q_i}$ can be obtained as in [35].

Thus, different controllers $\mathbf{K_i}$ can be implemented just by getting the YK parameter $\mathbf{Q_i}$. Different $\mathbf{Q_i}$'s are obtained for each controller in the set $\{K\}$; so the YK parameter set is $\{Q\} = \{\mathbf{Q_0}, ..., \mathbf{Q_i}, ..., \mathbf{Q_p}\}$. As the initial controller is K_0, its corresponding $\mathbf{Q_0} = \mathbf{0}$. The control structure able to switch from an initial controller $\mathbf{K_0}$ to any in the set $\{K\}$ is depicted in Fig. 2 (a).

When doing controller transitions, the scalar factor γ_i plays a key role. It regulates the different level of activation of the YK parameter $\mathbf{Q_i}$. γ_i may vary from 0 to 1, being 0 a 100% contribution of $\mathbf{K_0}$ and 1 a 100% contribution of $\mathbf{K_i}$. If the set of controllers is greater than 2, $n > 2$, a linear combination of all the controllers could be implemented as:

$$\mathbf{Q} = \sum_{i=1}^{n} \gamma_i \mathbf{Q_i} \quad with \quad \sum_{i=1}^{n} \gamma_i = 1 \tag{4}$$

In the present work, just one of the candidate controllers is activated at the same time. Controller reconfiguration depends on the supervisor.

B. Dual YK parameterization. Supervisor

Dual YK parameterization is used to recast CL identification into an OL-like problem [36]. It provides all the plants stabilized by a given controller. Again, once double coprime factors are obtained for an initial plant $\mathbf{G_0}$ and a fixed controller $\mathbf{K_i}$ in the set $\{K\}$, all the plants stabilized by the controller can be represented by the dual YK parameter $\mathbf{S_i}$.

From the general description of any plant stabilized by $\mathbf{K_i}$, it is possible to identify the real plant $\mathbf{G_{real}}$ connected to the controller, just by identifying the dual YK parameter $\mathbf{S_i}$. Then the set of all plants stabilized by $\mathbf{K_i}$ is represented by the scheme in Fig. 2 (b). To identify $\mathbf{S_i}$, output ζ_0 and input z_i need to be obtained from measurable data:

$$z_i = \tilde{\mathbf{V}}_\mathbf{i} u - \tilde{\mathbf{U}}_\mathbf{i} y \tag{5}$$

$$\zeta_0 = \tilde{\mathbf{M}}_\mathbf{0} y - \tilde{\mathbf{N}}_\mathbf{0} u \tag{6}$$

Even when u and y are measured in CL, the identification of $\mathbf{S_i}$ is OL-like by employing the filtered signals ζ_0 and z_i. Identification algorithms as [37] [38] [39] are suitable for $\mathbf{S_i}$ identification.

$\mathbf{S_i}$ can be interpreted as the difference between real $\mathbf{G_{real}}$ and initial plant $\mathbf{G_0}$. Its output, ζ_0, gives an insight of this difference. When $\mathbf{G_{real}} = \mathbf{G_0}$ the value of ζ_0 is zero. MMAC algorithm uses this property, so an identification algorithm is no longer needed to figure out the closest plant in the set $\{G\}$.

C. MMAC algorithm

Let $\{K\} = \{\mathbf{K_0}..., \mathbf{K_i}..., \mathbf{K_n}\}$ be a set of predesigned controllers for different plants $\{G\} = \{\mathbf{G_0}..., \mathbf{G_i}..., \mathbf{G_n}\}$. γ is the switching sequence: $\{\gamma\} = \{\gamma_0..., \gamma_i..., \gamma_n\}$. It determines which controller $\mathbf{K_i}$ is activated by setting the corresponding γ_i to 1 at the YK reconfiguration controller structure shown in Fig. 2 (a). Only one γ_i can be set to 1 at the same time.

The switching sequence γ is specified by the supervisor. The goal is to figure out which plant in the set $\{G\}$ is the closest to the real plant $\mathbf{G_{real}}$. As outlined above, S_i does not need to be directly identified to know the closest plant in the set [27]. If $\mathbf{G_{real}}$ coincides with the initial plant $\mathbf{G_0}$, ζ_0 should be zero for any value of u and y. By choosing different coprime factors $\tilde{\mathbf{M}}_\mathbf{i}$ and $\tilde{\mathbf{N}}_\mathbf{i}$ for every nominal plant in the set $\{G\}$, $\zeta_i = \tilde{M}_i y - \tilde{N}_i u$ gives the closeness to these plants. The smallest truncated 2-norm $J_i = (\|\zeta_i\|_2)^2$ will activate the signal γ_i corresponding to the controller $\mathbf{K_i}$ able to fulfill performance requirements.

The MMAC algorithm for a real plant $\mathbf{G_{real}}$ with a set of stabilizing controllers $\{K\}$ designed for a set of nominal plants $\{G\}$ is described in Algo. 1. h should be positive, and expresses the mandatory difference between two norms for controller change.

Algorithm 1 Multi Model Adaptive Control

1. Initialization
$\gamma[n] = [0]$ ▷ Switching sequence initialization
$K[n] = [\mathbf{K_0}, ..., \mathbf{K_j}, ..., \mathbf{K_n}]$ ▷ Candidate controllers
$\tilde{M}[n] = [\tilde{\mathbf{M}}_\mathbf{0}, ..., \tilde{\mathbf{M}}_\mathbf{i}, ..., \tilde{\mathbf{M}}_\mathbf{n}]$ ▷ Left coprime factor $\mathbf{M}$ for $\mathbf{G_i}$
$\tilde{N}[n] = [\tilde{\mathbf{N}}_\mathbf{0}, ..., \tilde{\mathbf{N}}_\mathbf{i}, ..., \tilde{\mathbf{N}}_\mathbf{n}]$ ▷ Left coprime factor $\mathbf{N}$ for $\mathbf{G_i}$
$\zeta[n] = [0]$ ▷ S_i output initialization
$J[n] = [0]$ ▷ Truncated 2-norm initialization
loop
2. YK Controller reconfiguration
UpdateController(γ) ▷ Apply controller $\mathbf{K_i}$, with $i = \gamma$
Get(u,y) ▷ Obtain measurements u and y
3. Supervisor
3.1 Closeness to plants in set
for i in $(0,n)$ **do**
$\zeta[i] = \tilde{M}[i]y - \tilde{N}[i]u$ ▷ Output of $\mathbf{S_i}$
$J[i] = (norm2(\zeta[i]))^2$ ▷ Compute truncated 2-norm/ Closeness to nominal plants
$i_{min} = argmin_i\{i \in n \mid J[i]\}$ ▷ The smallest norm corresponds to the closest plant
3.2 Evaluate switching sequence
if $(J[\gamma[i] == 1] \leq J[i_{min}] + h)$ **then**
$\gamma = \gamma$ ▷ Previous controller remains
else
$\gamma[i_{min}] = 1$ ▷ Controller changes
$\gamma[\forall i \quad except \quad i = i_{min}] = 0$
end loop

IV. MMAC for CACC applications

This section describes how MMAC can be used together with a CACC feedforward/feedback structure to address the problem of heterogeneity in CACC string of vehicles.

This work proposes a feedforward/feedback fractional-order (FO) PD controller [40], able to keep performance but increasing flexibility design by moving from integer to fractional order differentiation. This CACC FOPD controller is extended with the feedforward filter in [16] to ensure string stability even if different dynamics are present in the string. A model of each of the vehicles in the string would be necessary. Further details are in subsection CACC control structure.

In order to avoid any identification process that could slow down the control loop, MMAC is used as a supervisor able to choose the proper controller among a pre-designed CACC controller set $\{K\}$. The task of the supervisor will be to estimate the closest plant to the ego and preceding vehicles, so the controller ensuring string stability will be chosen. Details how Algo. 1 is modified for a CACC application are in subsection Supervisor and switching.

A. CACC control structure

A string of z vehicles driving in the same lane is considered. j determines the order of a vehicle inside the string, $j \in [1, z]$. Vehicle j denotes ego-vehicle and vehicle $j-1$ preceding vehicle. The solution here focuses on ego-vehicle situation depending on preceding's dynamics. From now on, $\mathbf{G_{real}}$ is equivalent to ego-vehicle $\mathbf{G_j}$.

Vehicle j tracks its preceding vehicle, regulating the distance with respect to the desired inter-vehicle distance provided by a constant time gap spacing policy [6]. As seen in Eq. (7) the desired intervehicle distance $d_{ref,j}$ is the addition of two terms, a fixed standstill distance s_j and a variable term composed by the vehicle's velocity v_j multiplied by the constant time gap h_j.

$$d_{ref,j} = s_j + h_j v_j \tag{7}$$

This desired distance is compared with the real one d_j, so a distance error needs to be regulated to zero: $e_{d,j} = d_j - d_{ref,j}$. A FOPD controller (see Eq. (8)) is designed following guidelines in [41].

$$\mathbf{K_j} = K_{p,j} + K_{d,j} s^{\alpha_j} = \left[\begin{array}{c|c} \mathbf{A_j^c} & \mathbf{B_j^c} \\ \hline \mathbf{C_j^c} & \mathbf{D_j^c} \end{array}\right] \tag{8}$$

String stability is defined as the attenuation of speed disturbances along the string of vehicles. A sufficient condition is presented in [42]: The absolute position of each vehicle cannot be amplified as it propagates along the string:

$$SS = \|x_j / x_{j-1}\|_\infty \leq 1 \quad for \quad i > 1 \tag{9}$$

When ideal communication is considered (no delays are present in the communication link), the choice of a feedforward filter as $\mathbf{F_j} = 1/(h_j s + 1) = [\mathbf{A_j^f} \quad \mathbf{B_j^f}; C_j^f \quad \mathbf{D_j^f}]$ ensures string stability for any positive value of h_j [43]. The statement above is no longer valid when a heterogeneous string of

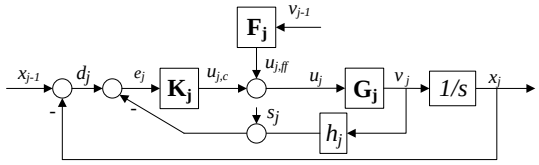


Fig. 3. CACC SISO structure.

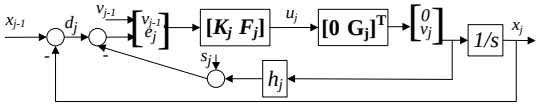


Fig. 4. CACC MISO structure.

vehicles is considered. [16] included dynamics of ego and preceding vehicles in F_j (see Eq. 10), proving string stability even when dynamics are not the same.

$$\mathbf{F_j} = \frac{1}{h_j s+1}\frac{G_{j-1}}{G_j} = \left[\begin{array}{c|c} \mathbf{A_j^f} & \mathbf{B_j^f} \\ \hline \mathbf{C_j^f} & \mathbf{D_j^f} \end{array}\right] \tag{10}$$

Both ego $\mathbf{G_j}$ and preceding vehicle $\mathbf{G_{j-1}}$ need to be known to ensure string stability. When $\mathbf{G_j} = \mathbf{G_{j-1}}$, the feedforward controller reduces to the homogeneous case.

Figure 3 shows the classical SISO structure for CACC systems. Notice how the control signal u_j is the addition of $u_{j,ff}$ and $u_{j,c}$. Here the controller structure is modified to MIMO, so FOPD controller $\mathbf{K_j}$ and feedforward filter $\mathbf{F_j}$ can be changed at once by using the YK parameterization. The modified MIMO structure is shown in Fig. 4.

Notice that there are as many controllers as there are possible dynamics combinations between ego and preceding vehicles.

B. Supervisor and switching

Let's consider n number of nominal plants in the set $\{G\} = \{\mathbf{G_0}...,\mathbf{G_i}...,\mathbf{G_n}\}$. Preceding $\mathbf{G_{j-1}}$ and ego $\mathbf{G_j}$ vehicles can be described by any of these nominal plants; or have a close behavior to one of them. Depending on ego and preceding dynamics combination, a set of $(n+1)^2$ CACC controllers is created $\{K\} = \{\mathbf{K_{xr}}\}$, where x is the closest plant in $\{G\}$ to the preceding vehicle $\mathbf{G_{j-1}}$, and r the closest to the ego-vehicle $\mathbf{G_j}$. Similarly, γ is the switching sequence $\{\gamma\} = \{\gamma_{xr}\}$, specifying which controller in $\{K\}$ is activated by modifying to 1 the corresponding γ_{xr} at the YK controller reconfiguration structure. Only one γ_{xr} can be set to 1 at the same time.

The switching sequence γ is specified by the supervisor. The goal is to determine which of the plants in the set $\{G\}$ is the closest to the real one $\mathbf{G_j}$. As already outlined, this is done through the signal ζ_i. The smaller its truncated 2-norm J_i the closer to the plant $\mathbf{G_i}$. The same kind of system is installed in the preceding vehicle, sending to ego-vehicle which is the closest plant to $\mathbf{G_{j-1}}$. Once x and r are known, the corresponding switching signal γ_{xr} is activated, so string stability is ensured.

The general MMAC algorithm in Algo. 1 is modified for CACC applications. It remains the same, except for a switching sequence γ that depends on two different indexes, a communication link added to get the value of x when obtaining measurements u and y, and an extra condition when ego-vehicle is exactly between two plants in the set $\{G\}$. Even if unlikely, the fastest model is chosen in order to provide the most string stable controller in the most critical situation. The mandatory difference between two norms h is set to 0.4.

V. Simulation results

This section presents the MMAC CACC performance when the string of vehicles is heterogeneous. In the following subsections, the sets of nominal plants and corresponding CACC controllers are introduced. These sets are chosen according to the requirements of convergence and stability. Different simulations have been carried out. The first of them (subsection Matching case), dynamics of both vehicles, preceding and ego, coincide exactly with one of the plants in the set $\{G\}$. Subsection Non-matching plant considers that preceding and/or ego-vehicle have dynamics close to one of the plants in the set.

A. Plant and controller sets

Convergence and stability for MMAC algorithms are assured by assuming that the true plant is sufficiently close to the identified model in the set [44] [45]. Thus, determination of the correct set of nominal plants ensures convergence and stability [25]. The idea is to determine the stability criterion for each pair plant/model, analyzing the maximum uncertainty around the plant without affecting the performance/stability of the system. In that way, one can define the required number of plants for the desired application range.

Vehicles with fast or slow dynamics, and with a certain overshoot are considered through second-order transfer functions: They are LTI plants mapping control velocity signals in output velocity signals:

$$\mathbf{G_i} = \frac{w_{n,i}^2}{s^2 + 2f_{damping,i}w_{n,i}s + w_{n,i}^2} = \left[\begin{array}{c|c} \mathbf{A_i} & \mathbf{B_i} \\ \hline \mathbf{C_i} & \mathbf{D_i} \end{array}\right] \tag{11}$$

where $f_{damping,i}$ is the damping factor and $w_{n,i}$ is the natural frequency in rad/s. Application range is defined by limits in $f_{damping,i}$ and $w_{n,i}$: $f_{damping,i} \in [0.55, 0.65]$ and $w_{n,i} \in [0.9524, 6.667]$. Notice that these limits could be extended in order to deal with every possible vehicle in the market. These limits are in consistency with real vehicle models in the literature as [46], [47] and [48], validating the selected application range.

Once the application range is defined, string stability in Eq. (9) is extended in order to get the minimum number of nominal plants in $\{G\}$ able to cover the whole application range with string stability guarantee. For doing so, uncertainties in damping factor and natural frequency of both ego and preceding vehicles are included into the string stability condition (see Eqs. (12) and (13)) as $\Delta f_{damping,j}$, $\Delta f_{damping,j-1}$, $\Delta w_{n,j}$ and $\Delta w_{n,j-1}$. MMAC uses the same LTI plant model for a given vehicle operational range. When doing so, there are

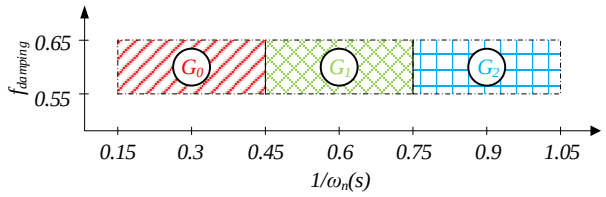


Fig. 5. Uncertainty range string stable for pairs $(\mathbf{G_0},\mathbf{K_0})$, $(\mathbf{G_1},\mathbf{K_1})$ and $(\mathbf{G_2},\mathbf{K_2})$.

inaccuracies between the associated model and the real model. This can be seen as heterogeneous string stability problem within the given operational range. String stability must be guaranteed in the worst-case scenario, providing the number of nominal plants for the MMAC design.

For the specific range defined above, three models in the set $\{G\}$ are enough to ensure string stability in the whole application range. String stable areas for pairs $(\mathbf{G_0},\mathbf{K_0})$, $(\mathbf{G_1},\mathbf{K_1})$ and $(\mathbf{G_2},\mathbf{K_2})$ are depicted in Fig. 5. Small overlapping between them is carried out in order to have a safer solution.

$$\frac{X_j}{X_{j-1}}(\Delta f_{damping,j},\Delta f_{damping,j-1},\Delta w_{n,j},\Delta w_{n,j-1}) = \\ = \frac{\frac{G_j(\Delta f_{damping,j},\Delta w_{n,j})F_j}{G_{j-1}(\Delta f_{damping,j-1},\Delta w_{n,j-1})} - \frac{G_j(\Delta f_{damping,j},\Delta w_{n,j})K_j}{s}}{1-(\frac{G_j(\Delta f_{damping,j},\Delta w_{n,j})K_j}{s} + G_j(\Delta f_{damping,j},\Delta w_j)K_j h_j)} \tag{12}$$

$$\mathbf{G_i}(\Delta f_{damping,i},\Delta w_{n,i}) = \\ = \frac{(w_{n,i}+\Delta w_{n,i})^2}{s^2+2(f_{damping,i}+\Delta f_{damping,i})(w_{n,i}+\Delta w_{n,i})s+(w_{n,i}+\Delta w_{n,i})^2} \\ with \quad i=j,j-1 \tag{13}$$

Values for nominal plants in the set are in Table I. $\mathbf{G_0}$, $\mathbf{G_1}$ and $\mathbf{G_2}$ go from faster to slower dynamics. Plants $\mathbf{G_{x1}}$, $\mathbf{G_{x2}}$ and $\mathbf{G_{x3}}$ are also present in Table I, and they will be the non-matching cases in the corresponding subsection. Notice how $\mathbf{G_{x3}}$ have first order dynamics corresponding to the production vehicle in [47]. The latter allows to test how the algorithm behaves in a high-speed scenario and with a different order model. τ_i refers to the time constant of the first order transfer function.

TABLE I
PLANTS AND PARAMETERS.

$\mathbf{G_0}$	$f_{damping,0}=0.6$	$w_{n,0}=3.3333$
$\mathbf{G_1}$	$f_{damping,1}=0.6$	$w_{n,1}=1.6667$
$\mathbf{G_2}$	$f_{damping,2}=0.6$	$w_{n,2}=1.1111$
$\mathbf{G_{x1}}$	$f_{damping,2}=0.65$	$w_{n,2}=6.6667$
$\mathbf{G_{x2}}$	$f_{damping,2}=0.55$	$w_{n,2}=0.9524$
$\mathbf{G_{x3}}$	$\tau_3=0.2$	

For every plant in the set $\{G\}=\{\mathbf{G_0},\mathbf{G_1},\mathbf{G_2}\}$ a FOPD controller has been designed with the objective of being robust to uncertainties. Guidelines in [41] are followed, resulting in a controller set $\{K\}=\{\mathbf{K_0},\mathbf{K_1},\mathbf{K_2}\}$. Controller parameters are shown in Table II. This set is extended including the feedforward filter; Eq. (10) depends on possible combinations between plants in $\{G\}$, yielding $\{K_{xr}\}=\{\mathbf{K_{00}},\mathbf{K_{01}},\mathbf{K_{02}},\mathbf{K_{10}},\mathbf{K_{11}},\mathbf{K_{12}},\mathbf{K_{20}},\mathbf{K_{21}},\mathbf{K_{22}}\}$, where x is the plant in the preceding vehicle, and r the plant in the ego-vehicle.

TABLE II
FOPD CONTROLLER PARAMETERS

	K_p	K_p	α	h
$\mathbf{K_0}$	$K_{p,0}=0.35$	$K_{d,0}=0.15$	$\alpha_0=0.3847$	$h_0=1s$
$\mathbf{K_1}$	$K_{p,1}=0.5$	$K_{d,1}=0.225$	$\alpha_1=0.3847$	$h_1=1s$
$\mathbf{K_2}$	$K_{p,2}=0.6$	$K_{d,2}=0.3$	$\alpha_2=0.3847$	$h_2=1s$

Once nominal plants and candidate controllers are defined, Theorem 1 is applied to get the coprime factors needed in Theorem 3, so the set of YK parameters is obtained as $\{Q_{xr}\}=\{\mathbf{Q_{00}},\mathbf{Q_{01}},\mathbf{Q_{02}},\mathbf{Q_{10}},\mathbf{Q_{11}},\mathbf{Q_{12}},\mathbf{Q_{20}},\mathbf{Q_{21}},\mathbf{Q_{22}}\}$. Each of them permits controller reconfiguration from an initial controller to a controller in the set $\{K_{xr}\}$. This set is reduced by choosing an initial controller $\mathbf{K_{00}}$. Thus, the new set of YK parameters is $\{Q_{xr}\}=\{\mathbf{Q_{01}},\mathbf{Q_{02}},\mathbf{Q_{10}},\mathbf{Q_{11}},\mathbf{Q_{12}},\mathbf{Q_{20}},\mathbf{Q_{21}},\mathbf{Q_{22}}\}$. Controller transition is done through the scalar factor γ_{xr} associated to $\mathbf{Q_{xr}}$. In the switching sequence γ only one γ_{xr} can be set to one at the same time. If none of them is activated, the inital controller $\mathbf{K_{00}}$ will be applied. Table III gathers the information related to the controller set and the corresponding YK parameter.

TABLE III
CONTROLLER SET AND YK PARAMETER

$\mathbf{K_{xr}}$	$\mathbf{G_{j-1}}$	$\mathbf{G_j}$	$\mathbf{K_j}$	$\mathbf{F_j}$	$\mathbf{Q_{xr}}$	γ_{xr}
$\mathbf{K_{00}}$	$\mathbf{G_0}$	$\mathbf{G_0}$	$\mathbf{K_0}$	$\frac{1}{hs+1}$	–	–
$\mathbf{K_{01}}$	$\mathbf{G_0}$	$\mathbf{G_1}$	$\mathbf{K_1}$	$\frac{1}{hs+1}\frac{G_0}{G_1}$	$\mathbf{Q_{01}}$	γ_{01}
$\mathbf{K_{02}}$	$\mathbf{G_0}$	$\mathbf{G_2}$	$\mathbf{K_2}$	$\frac{1}{hs+1}\frac{G_0}{G_2}$	$\mathbf{Q_{02}}$	γ_{02}
$\mathbf{K_{10}}$	$\mathbf{G_1}$	$\mathbf{G_0}$	$\mathbf{K_0}$	$\frac{1}{hs+1}\frac{G_1}{G_0}$	$\mathbf{Q_{10}}$	γ_{10}
$\mathbf{K_{11}}$	$\mathbf{G_1}$	$\mathbf{G_1}$	$\mathbf{K_1}$	$\frac{1}{hs+1}$	$\mathbf{Q_{11}}$	γ_{11}
$\mathbf{K_{12}}$	$\mathbf{G_1}$	$\mathbf{G_2}$	$\mathbf{K_2}$	$\frac{1}{hs+1}\frac{G_1}{G_2}$	$\mathbf{Q_{12}}$	γ_{12}
$\mathbf{K_{20}}$	$\mathbf{G_2}$	$\mathbf{G_0}$	$\mathbf{K_0}$	$\frac{1}{hs+1}\frac{G_2}{G_0}$	$\mathbf{Q_{20}}$	γ_{20}
$\mathbf{K_{21}}$	$\mathbf{G_2}$	$\mathbf{G_1}$	$\mathbf{K_1}$	$\frac{1}{hs+1}\frac{G_2}{G_1}$	$\mathbf{Q_{21}}$	γ_{21}
$\mathbf{K_{22}}$	$\mathbf{G_2}$	$\mathbf{G_2}$	$\mathbf{K_2}$	$\frac{1}{hs+1}$	$\mathbf{Q_{22}}$	γ_{22}

B. Matching case

The matching case considers that ego and preceding vehicles have dynamics that coincide exactly with one of the plants in the set $\{G\}$. Specifically, preceding vehicle has dynamics corresponding to plant $\mathbf{G_{j-1}}=\mathbf{G_0}$, while ego-vehicle has faster dynamics like $\mathbf{G_j}=\mathbf{G_2}$.

The dual YK parameterization is used in the supervisor, so identification algorithms are not needed to determine which is the closest plant in the set. ζ_i is used instead. By obtaining the signal value for every plant in the set, one can easily determine closeness. The top graph in Fig. 6 shows the evolution of these signals through time for the ego-vehicle $\mathbf{G_j}$, while the bottom graph depicts the truncated 2-norm J_i related to ζ_i. Different

system activation times are considered to see how the system reacts to different initial conditions. Solid-line indicates the case where MMAC is activated at 0s, while dotted line shows the situation where MMAC is activated at 29s. In both cases, ζ_2 and J_2 are always zero, as $\mathbf{G_j} = \mathbf{G_2}$. This plant index r is used together with the received index x to specify the switching sequence γ.

Figure 7 depicts the performance of MMAC CACC algorithm when preceding vehicle is $\mathbf{G_{j-1}} = \mathbf{G_0}$ and ego-vehicle $\mathbf{G_j} = \mathbf{G_2}$. A comparison is made between an ego-vehicle with an erroneous controller and with the controller that makes the system string stable, observing the transition from one to another when using MMAC. The initial and erroneous controller in the YK controller reconfiguration is $\mathbf{K_{00}}$. The top graph plots vehicles' speeds when using $\mathbf{K_{00}}$ (dotted red line), $\mathbf{K_{02}}$ (dotted blue line), MMAC activated at 0s (solid green line) and MMAC activated at 29s (solid pink line). Preceding vehicle speed is also shown (black solid line). The second graph plots the switching sequence γ given by the supervisor. The switching sequence is determined by the signals in Fig. 6. Dot-line indicates the case where MMAC is activated at 29s. Only γ_{02} is shown to have a lighter graph; the rest are zero even when MMAC is activated at 29s. Finally, the bottom graph represents the distance error for each of the controllers.

At second 0 Ego-incorrect, Ego-MMAC 1 and Ego-MMAC 2 are overlapping; it is not until second 1.8 that the corresponding γ_{02} is activated, making a transition of Ego-MMAC 1 from Ego-Incorrect to Ego-Correct. From second 1.5 to 32 Ego-MMAC 2 matches with Ego-Incorrect as the system has been activated at second 29, and the correct γ_{02} is not identified until second 32. It is then when Ego-MMAC 2 starts its transition from Ego-Incorrect to Ego-Correct.

From these results, one observes the importance of using different controllers depending on vehicle dynamics. When using $\mathbf{K_{00}}$ the string of vehicles results unstable. Ego-vehicle speed is amplified in comparison with the preceding one, and distance error tracking is large. No matter the activation time, MMAC is able to specify in a few seconds the correct switching sequence that makes the string stable. Thus, distance error tends to zero and the transition between controllers is really soft. To the best of authors' knowledge, the only work mixing controller reconfiguration and dynamics estimation for an heterogeneous string of vehicles is the one in [17]. Parameters estimation takes 31s, which is much slower than the presented work.

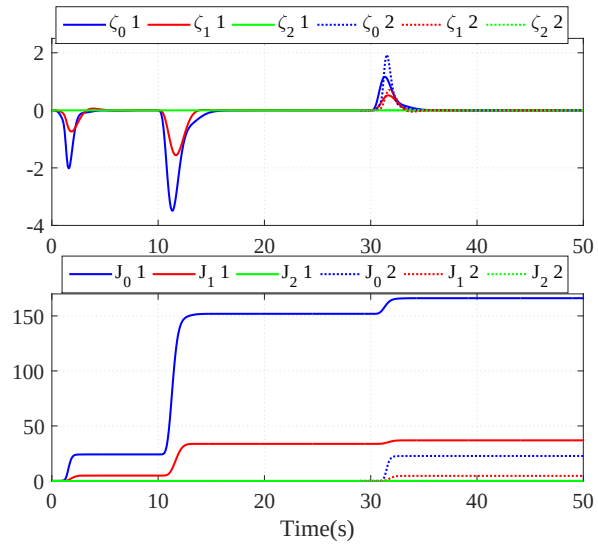


Fig. 6. ζ_i and J_i comparison. $\mathbf{G_{j-1}} = \mathbf{G_0}$ and $\mathbf{G_j} = \mathbf{G_2}$. Solid-line indicates the case where the system is activated at 0s, while dotted line shows the case where the system is activated at 29s.

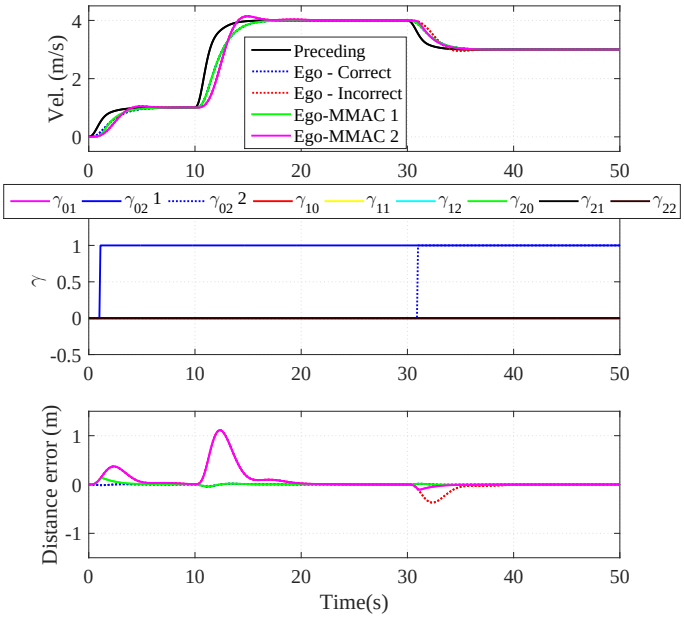


Fig. 7. Simulation results MMAC CACC. $\mathbf{G_{j-1}} = \mathbf{G_0}$ and $\mathbf{G_j} = \mathbf{G_2}$.

C. Non-matching case

The non-matching case considers that ego-vehicle and/or preceding vehicle are not one of the plants in the set $\{G\}$, but within the application range: $\mathbf{G_{x1}}$, $\mathbf{G_{x2}}$ and $\mathbf{G_{x3}}$. The Vinnicombe ν-gap in [49] is employed to figure out which is the closest plant in the set, so later can be verified if the proposed MMAC algorithm is properly working. ν-gap goes from 0 to 1, and expresses the difference between two plants; the closer to zero the more the two plants look alike. In the case of $\mathbf{G_{x1}}$, the lower value of ν-gap is for $\mathbf{G_0}$ (ν-gap($\mathbf{G_{x1}}, \mathbf{G_0}$)= 0.5336), so it is the closest plant to $\mathbf{G_{x1}}$ in the set. In the case of $\mathbf{G_{x2}}$, the closest plant results $\mathbf{G_2}$ with ν-gap($\mathbf{G_{x2}}, \mathbf{G_2}$)= 0.1449. Finally, in the case of the first order plant $\mathbf{G_{x3}}$ the closest results $\mathbf{G_0}$ with ν-gap($\mathbf{G_{x3}}, \mathbf{G_0}$)= 0.5722.

Figure 8 shows the evolution through time of ζ_i and J_i when $\mathbf{G_{j-1}} = \mathbf{G_1}$ and ego-vehicle $\mathbf{G_j} = \mathbf{G_{x3}}$ (Citroen C4 model used in [47]). Again different system activation times are considered, 10s and 35s in solid and dotted lines respectively. In both cases, one can observe how the minimum value corresponds to ζ_0. Even if $\mathbf{G_{x3}}$ is a production vehicle model and not a candidate plant in the set $\{G\}$, dual YK parameterization gives a good insight of the closest plant. Notice that ζ_0 is not zero

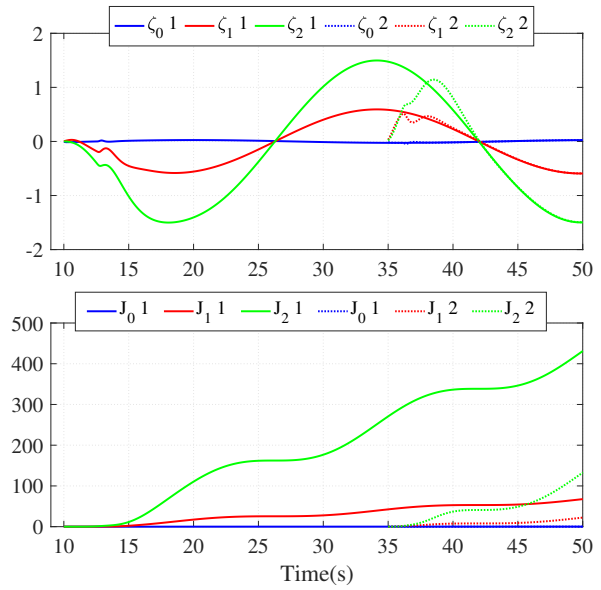


Fig. 8. ζ_i and J_i comparison. $\mathbf{G_{j-1} = G_1}$ and $\mathbf{G_j = G_{x1}}$. Solid-line indicates the case where the system is activated at 0s, while dotted line shows the case where the system is activated at 20s.

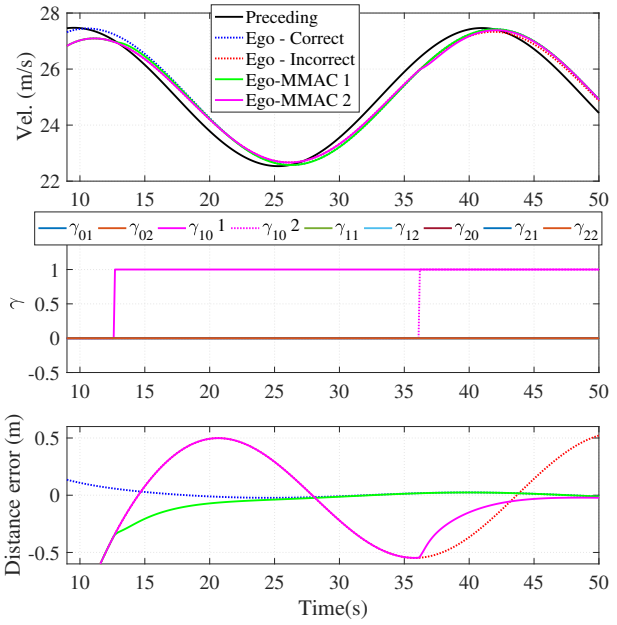


Fig. 9. Simulation results MMAC CACC. $\mathbf{G_{j-1} = G_1}$ and $\mathbf{G_j = G_{x3}}$.

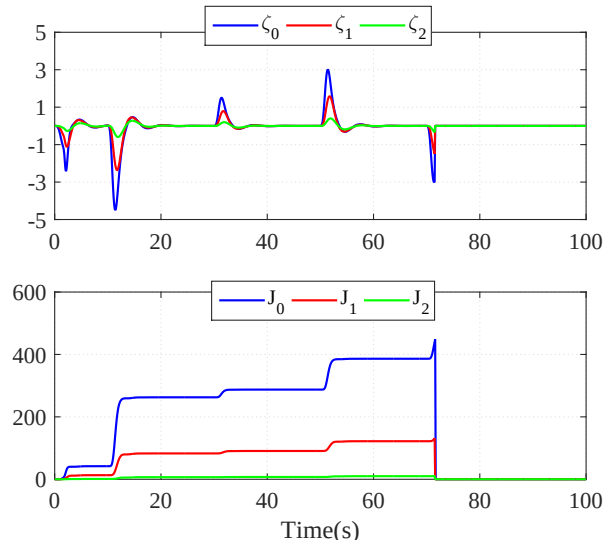


Fig. 10. ζ_i and J_i comparison. $\mathbf{G_{j-1} = G_{x1}}$ and $\mathbf{G_j = G_{x2}}$. System shutdown at 71.5s.

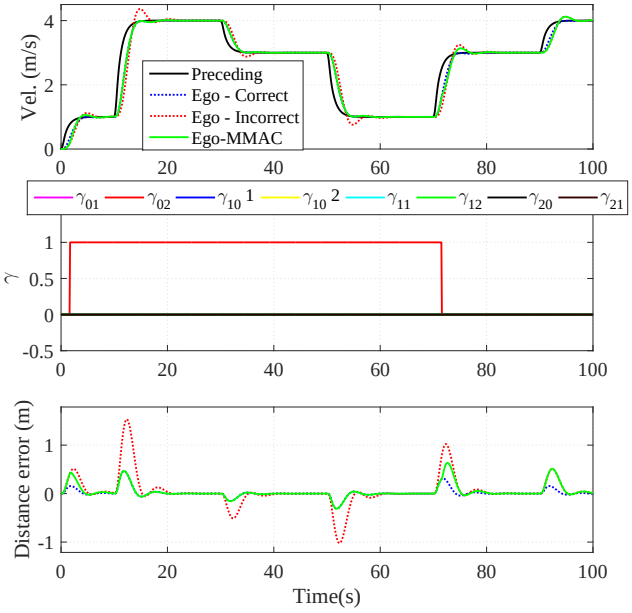


Fig. 11. Simulation results MMAC CACC. $\mathbf{G_{j-1} = G_{x1}}$ and $\mathbf{G_j = G_{x2}}$. System shutdown at 71.5s.

in these cases, but it is much more smaller than ζ_1 and ζ_2.

Figure 9 depicts the performance of MMAC CACC algorithm when preceding vehicle is $\mathbf{G_{j-1} = G_1}$ and ego-vehicle is the production vehicle $\mathbf{G_j = G_{x3}}$. Again incorrect ($\mathbf{K_{00}}$, dotted red line) and correct controller ($\mathbf{K_{10}}$, dotted blue line) are compared with MMAC activated at 10s (solid green line) and MMAC at 35s (solid pink line). Preceding vehicle (black solid line) follows a speed profile that goes between 22 and 28m/s. Indexes are identified in preceding and ego vehicles, so the corresponding γ_{10} is activated in few seconds no matter the activation time, making the distance error zero in a smooth way.

At second 10 Ego-incorrect, Ego-MMAC 1 and Ego-MMAC 2 are overlapping; it is not until second 12.8 that the corresponding γ_{10} is activated, making Ego-MMAC 1 goes from Ego-Incorrect to Ego-Correct. It is not until second 36 that Ego-MMAC 2 starts its transition from Ego-Incorrect to correct as the system has been activated at second 35, and the correct γ_{10} is identified at second 36.

Finally, a case with two non-matching models in preceding and ego vehicles is considered: $\mathbf{G_{j-1} = G_{x1}}$ and ego-vehicle $\mathbf{G_j = G_{x2}}$. This is the most critical case possible, as it considers

the fastest preceding vehicle together with the slowest ego-vehicle, and different damping factors in the limits of the application range. Evolution through time of ζ_i and J_i for ego-vehicle is in Fig. 10. The closest plant is $\mathbf{G_2}$ as ζ_2 is the lower one. Differences between the unstable controller and stable one are remarkable in the distance error tracking.

Figure 11 depicts the performance of MMAC CACC algorithm when preceding vehicle is $\mathbf{G_{j-1} = G_{x1}}$ and ego-vehicle $\mathbf{G_j = G_{x2}}$. Incorrect ($\mathbf{K_{00}}$, dotted red line) and correct controllers ($\mathbf{K_{02}}$, dotted blue line) are compared with MMAC solution (solid green line). Indexes are correctly identified in preceding and ego vehicles, so the corresponding γ_{02} is activated, preserving string stability. MMAC shut down is also analysed in the same figure; the system shuts off at $71.5s$ passing smoothly from a string stable behaviour to an unstable one. Notice how in the unstable case, the distance error tracking gets bigger than in previous cases. This distance error is directly associated with an ego-vehicle speed that is amplified.

At second 0 Ego-incorrect and Ego-MMAC are overlapping; it is at second 3 that the corresponding γ_{02} is activated, making Ego-MMAC goes from Ego-Incorrect to Correct. At second 71.5 MMAC system is switched off, making Ego-MMAC comes back to Ego-Incorrect.

It has been proved the correct behaviour of MMAC for CACC applications even if the real vehicle $\mathbf{G_j}$ is not exactly one of the candidate plants in the set $\{G\}$. Different orders and speed scenarios have been used. Real vehicle dynamics should be within the application range. A larger set of models and controllers would be needed in practice to cover the different dynamics in a fleet of vehicles.

VI. Experimental results

A string of two cycabs is used as the experimental test platform. Cycabs are mobile platforms used in several research labs conceived for urban applications [50]. The vehicle is controlled by a commanded velocity. The velocity is limited up to $4m/s$, because it is mainly designed for crowded areas where higher speeds can lead to unsafe situations. MMAC CACC algorithm is implemented in C++ using RTMaps[1] prototyping software.

Experimental results serve as convergence validation of a real system. This is a non-matching case. Since both vehicles in the string have similar dynamics, the string will always be homogeneous, not being required the activation of any γ_{xr} in the switching sequence γ. The initial controller is modified to $\mathbf{K_{01}}$; in that way, a modification of γ will be mandatory to obtain the best performance possible.

Figure 12 depicts the evolution through time of ζ_i and J_i during the experimental test. MMAC algorithm is activated at $0s$. The minimum value corresponds to ζ_0, so $\mathbf{G_0}$ is the closest plant to the real dynamics of a cycab. It is not until $2.5s$ that the differences between J_0, J_1 and J_2 are big enough to determine the switching sequence γ.

Figure 13 depicts the performance of MMAC CACC algorithm when applied to a string of two cycabs. A comparison is

[1]https://intempora.com/

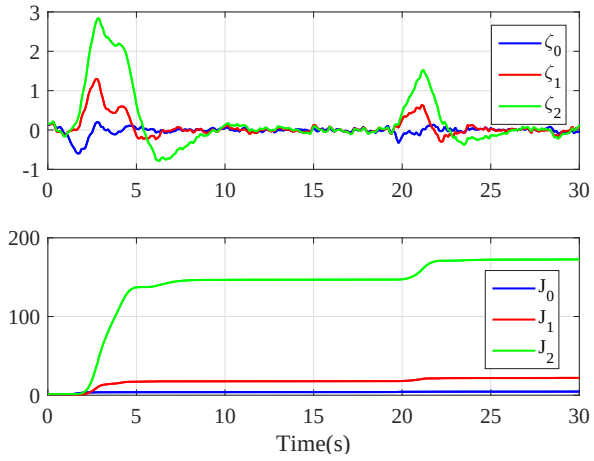


Fig. 12. ζ_i and J_i comparison. Experimental results. Homogeneous string of two cycabs.

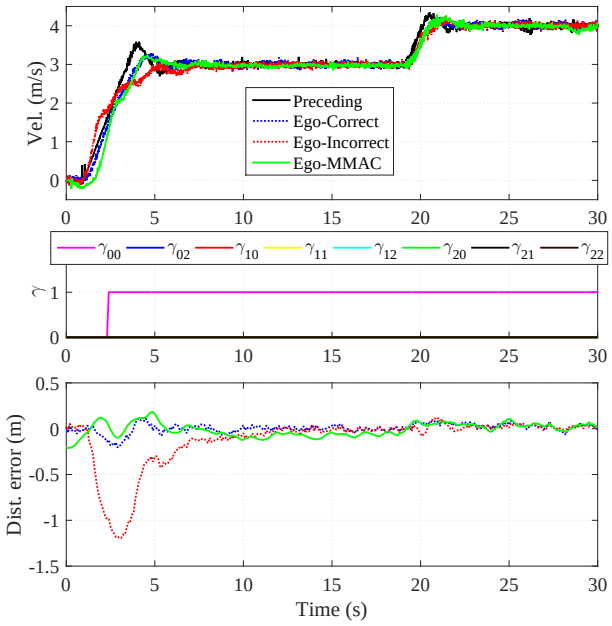


Fig. 13. Experimental results MMAC CACC. Homogeneous string of two cycabs.

made between the second cycab in the string with an erroneous controller and with the controller that makes the system string stable, showing the transition between both when using MMAC. The top graph plots cycabs' speeds when using $\mathbf{K_{10}}$ (dotted red line), $\mathbf{K_{00}}$ (dotted blue line) and MMAC activated at $0s$ (solid green line). First cycab's speed is also shown (black solid line). The middle graph shows the switching sequence γ determined by signals in Fig. 12. The bottom graph plots the corresponding distance errors when using correct, erroneous and MMAC controllers. Notice that initial distance errors are not the same for the 3 tests; a maximum error of $20cm$ was considered to start each of the tests.

From these results, one can see how important it is to take into account vehicle dynamics to ensure string stability. Inappropriate performance can be clearly appreciated on the

distance error tracking. Even if this is less remarkable in the speed response, distance error is longer when the velocity step is big enough. The difference in the second step is less remarkable, as the speed step is smaller, but oscillations are present in the distance error. For the MMAC algorithm, it takes 2.5s to determine that both vehicles are closer to G_0, switching to the homogeneous controller $\mathbf{K_{00}}$ through γ_{00}. The transition between both controllers is smooth. The resulting transient is a combination of the initial distance error and the controller reconfiguration induced by γ_{00}. These results validate the simulation results seen in the previous section.

Experimental results serve as a convergence proof of a real vehicle with some of the plants in $\{G\}$. As vehicles are homogeneous, the most important task is to detect that both vehicles have similar dynamics, in order to switch to the classical homogeneous CACC controller.

VII. Conclusions

This paper explores the use of MMAC for CACC applications in heterogeneous/homogeneous string of vehicles.

Youla-Kucera theory is explained as the basis of a general MMAC algorithm based on a set of nominal plants and predefined controllers. MMAC acts as a supervisor determining the closest plant to the real system in the set, switching to the predesigned controller that will give the best performance possible. No identification algorithm is needed, the dual YK parameterization is used instead.

The general MMAC algorithm is later applied to a feedforward/feedback FOPD CACC system, so a structure able to handle different dynamics in a string without the need of an identification algorithm is obtained.

Performance of MMAC CACC algorithm is analysed through simulation results. A set of 3 linear plants is considered for validation purposes. Real vehicles models are considered within the application range. Dynamics matching and non-matching (even with different model order) cases with a plant in the set are considered, verifying how the supervisor is able to provide the closest plant in the set, activating the controller that ensures string stability. Dynamics estimation results much faster than other estimation processes in the literature. A test setup of two cycabs is used, showing that MMAC CACC design is not only theoretically, but also practically feasible.

Although the set $\{G\}$ is composed by 3 plants, this could be extended to n plants, so a larger dynamic range can be covered in heterogeneous CACC applications.

Finally, as communication delays have a direct impact in the string stability of the system, the developed YK controller structure will be extended in order to deal with such delays and different communication topologies.

VIII. Acknowledgments

Authors express their gratitude to the french project VALET (ANR-15-CE22-0013) and the RITS Team for its support in the development of this work.

References

[1] E. Energy, "Transport in figures'–statistical pocketbook 2005," *European Commission, DG TREN in co-operation with Eurostat*, 2006.

[2] G. Cookson. (2016, November) Europe's traffic hotspots.

[3] S. Moon, I. Moon, and K. Yi, "Design, tuning, and evaluation of a full-range adaptive cruise control system with collision avoidance," *Control Engineering Practice*, vol. 17, no. 4, pp. 442–455, 2009.

[4] J. Ploeg, B. T. Scheepers, E. Van Nunen, N. Van de Wouw, and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, 2011, pp. 260–265.

[5] S. Shladover, D. Su, and X.-Y. Lu, "Impacts of cooperative adaptive cruise control on freeway traffic flow," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2324, pp. 63–70, 2012.

[6] D. Swaroop and K. Rajagopal, "A review of constant time headway policy for automatic vehicle following," in *Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE*. IEEE, 2001, pp. 65–69.

[7] G. Naus, "Model-based control for automotive applications," 2010.

[8] S. Öncü, "String stability of interconnected vehicles: Network-aware modelling, analysis and experiments," *Eindhoven University of Technology*, 2014.

[9] M. R. Nieuwenhuijze, T. van Keulen, S. O?ncu?, B. Bonsen, and H. Nijmeijer, "Cooperative driving with a heavy-duty truck in mixed traffic: Experimental results," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1026–1032, 2012.

[10] E. Shaw and J. K. Hedrick, "Controller design for string stable heterogeneous vehicle strings," in *Decision and Control, 2007 46th IEEE Conference on*. IEEE, 2007, pp. 2868–2875.

[11] ——, "String stability analysis for heterogeneous vehicle strings," in *American Control Conference, 2007. ACC'07*. IEEE, 2007, pp. 3118–3125.

[12] I. Lestas and G. Vinnicombe, "Scalability in heterogeneous vehicle platoons," in *American Control Conference, 2007. ACC'07*. IEEE, 2007, pp. 4678–4683.

[13] G. Guo and W. Yue, "Hierarchical platoon control with heterogeneous information feedback," *IET control theory & applications*, vol. 5, no. 15, pp. 1766–1781, 2011.

[14] F. Gao, D. Dang, and S. E. Li, "Control of a heterogeneous vehicular platoon with uniform communication delay," in *Information and Automation, 2015 IEEE International Conference on*. IEEE, 2015, pp. 2419–2424.

[15] F. Gao, S. E. Li, Y. Zheng, and D. Kum, "Robust control of heterogeneous vehicular platoon with uncertain dynamics and communication delay," *IET Intelligent Transport Systems*, vol. 10, no. 7, pp. 503–513, 2016.

[16] C. Wang and H. Nijmeijer, "String stable heterogeneous vehicle platoon using cooperative adaptive cruise control," in *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*. IEEE, 2015, pp. 1977–1982.

[17] Y. A. Harfouch, S. Yuan, and S. Baldi, "Adaptive control of interconnected networked systems with application to heterogeneous platooning," in *Control & Automation (ICCA), 2017 13th IEEE International Conference on*. IEEE, 2017, pp. 212–217.

[18] C. Flores, V. Milanés, and F. Nashashibi, "Online feedforward/feedback structure adaptation for heterogeneous cacc strings," in *2018 Annual American Control Conference (ACC)*. IEEE, 2018, pp. 49–55.

[19] Y. Zhu, D. Zhao, and Z. Zhong, "Adaptive optimal control of heterogeneous cacc system with uncertain dynamics," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 4, pp. 1772–1779, 2018.

[20] L. Zhang, J. Sun, and G. Orosz, "Hierarchical design of connected cruise control in the presence of information delays and uncertain vehicle dynamics," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 1, pp. 139–150, 2017.

[21] J. Lourenco and J. Lemos, "Learning in multiple model adaptive control switch," *IEEE instrumentation & measurement Magazine*, vol. 9, no. 3, pp. 24–29, 2006.

[22] O. Brend, C. Freeman, and M. French, "Multiple-model adaptive control of functional electrical stimulation," *IEEE Transactions on Control Systems Technology*, vol. 23, no. 5, pp. 1901–1913, 2015.

[23] Y. Ma, G. Tao, B. Jiang, and Y. Cheng, "Multiple-model adaptive control for spacecraft under sign errors in actuator response," *Journal of Guidance, Control, and Dynamics*, vol. 38, no. 7, pp. 628–641, 2015.

[24] L. Zhou, X. Yu, B. Li, C. Zheng, J. Liu, Q. Liu, and K. Guo, "Damping inter-area oscillations with large-scale pv plant by modified multiple-model adaptive control strategy," *IEEE Transactions on Sustainable Energy*, vol. 8, no. 4, pp. 1629–1636, 2017.

[25] B. Anderson, T. Brinsmead, D. Liberzon, and A. Stephen Morse, "Multiple model adaptive control with safe switching," *International journal of adaptive control and signal processing*, vol. 15, no. 5, pp. 445–470, 2001.

[26] S. Baldi, G. Battistelli, D. Mari, E. Mosca, and P. Tesi, "Multiple-model adaptive switching control for uncertain multivariable systems," in *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*. IEEE, 2011, pp. 6153–6158.

[27] J. Bendtsen and K. Trangbaek, "Multiple model adaptive control using dual youla-kucera factorisation," *IFAC Proceedings Volumes*, vol. 45, no. 13, pp. 63–68, 2012.

[28] E. van Nunen, J. Reinders, E. Semsar-Kazerooni, and N. Van De Wouw, "String stable model predictive cooperative adaptive cruise control for heterogeneous platoons," *IEEE Transactions on Intelligent Vehicles*, vol. 4, no. 2, pp. 186–196, 2019.

[29] B. D. Anderson, "From youla–kucera to identification, adaptive and nonlinear control," *Automatica*, vol. 34, no. 12, pp. 1485–1506, 1998.

[30] J. Y. Ishihara and R. M. Sales, "Doubly coprime factorizations related to any stabilizing controllers in state space," *Automatica*, vol. 35, no. 9, pp. 1573–1577, 1999.

[31] J. F. Pommaret and A. Quadrat, "Generalized bezout identity," in *Applicable Algebra in Engineering, Communication and Computing*, vol. 9, no. 2, 1998, pp. 91–116.

[32] T.-T. Tay, I. Mareels, and J. B. Moore, *High performance control*. Springer Science & Business Media, 2012.

[33] J. B. Moore, K. Glover, and A. Telford, "All stabilizing controllers as frequency-shaped state estimate feedback," *IEEE Transactions on Automatic Control*, vol. 35, no. 2, pp. 203–208, 1990.

[34] F. Hansen, G. Franklin, and R. Kosut, "Closed-loop identification via the fractional representation: Experiment design." in *American Control Conference*, 1989, pp. 1422–1427.

[35] H. Niemann and J. Stoustrup, "An architecture for implementation of multivariable controllers," in *In American Control Conference*, vol. 6, 1999, pp. 4029–4033.

[36] S. Dasgupta and B. D. Anderson, "A parametrization for the closed-loop identification of nonlinear time-varying systems," *Automatica*, vol. 32, no. 10, pp. 1349–1360, 1996.

[37] S. Karaboyas and N. Kalouptsidis, "Efficient adaptive algorithms for arx identification," *IEEE Transactions on signal processing*, vol. 39, no. 3, pp. 571–582, 1991.

[38] O. TUTSOY, D. E. BARKANA, and N. F. OZDİL, "Design and evaluation of an adaptive estimator in the presence of un-known system dynamics and armax type noise."

[39] O. Tutsoy, D. E. Barkana, and H. Tugal, "Design of a completely model free adaptive control in the presence of parametric, non-parametric uncertainties and random control signal delay," *ISA transactions*, vol. 76, pp. 67–77, 2018.

[40] H. Li, Y. Luo, and Y. Chen, "A fractional order proportional and derivative (fopd) motion controller: tuning rule and experiments," *IEEE Transactions on control systems technology*, vol. 18, no. 2, pp. 516–520, 2010.

[41] C. Flores, V. Milanés, and F. Nashashibi, "Using fractional calculus for cooperative car-following control," in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 2016, pp. 907–912.

[42] R. Rajamani, *Vehicle dynamics and control*, S. S. . B. Media., Ed., 2011.

[43] S. Öncü, N. Van De Wouw, and H. Nijmeijer, "Cooperative adaptive cruise control: Tradeoffs between control and network specifications," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, 2011, pp. 2051–2056.

[44] B. D. Anderson, T. S. Brinsmead, F. De Bruyne, J. Hespanha, D. Liberzon, and A. S. Morse, "Multiple model adaptive control. part 1: Finite controller coverings," *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, vol. 10, no. 11-12, pp. 909–929, 2000.

[45] M. Stefanovic, R. Wang, and M. G. Safonov, "Stability and convergence in adaptive systems," in *American Control Conference, 2004. Proceedings of the 2004*, vol. 2. IEEE, 2004, pp. 1923–1928.

[46] V. Milanés, S. E. Shladover, J. Spring, C. Nowakowski, H. Kawazoe, and M. Nakamura, "Cooperative adaptive cruise control in real traffic situations," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 1, pp. 296–305, 2014.

[47] G. Naus, R. Vugts, J. Ploeg, R. van de Molengraft, and M. Steinbuch, "Cooperative adaptive cruise control, design and experiments," in *American Control Conference (ACC), 2010*. IEEE, 2010, pp. 6145–6150.

[48] R. Kianfar, P. Falcone, and J. Fredriksson, "A receding horizon approach to string stable cooperative adaptive cruise control," in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2011, pp. 734–739.

[49] V. Glenn, *Uncertainty And Feedback, H Loop-shaping And The V-gap Metric*. World Scientific, 2000.

[50] S. Sekhavat and J. Hermosillo, "The cycab robot: a differentially flat system. in intelligent robots and systems," in *In Intelligent Robots and Systems, 2000.(IROS 2000). Proceedings. 2000 IEEE/RSJ International Conference on*, vol. 1, 2000, pp. 312–317.



Francisco Navas received his PhD degree inside the RITS team at INRIA (Paris) and MINES ParisTech, PSL Research University, in November 2018. Since 2018 he is with the Research Department at AKKA Technologies. His research interests include autonomous vehicles, Youla-Kucera parameterization, switching controllers, and vehicle-infrastructure cooperation.



Vicente Milanés received his Ph.D. degree in electronic engineering from University of Alcalá, Madrid, Spain, in 2010. He was with the AUTOPIA program at the Center for Automation and Robotics (UPM-CSIC, Spain) from 2006 to 2011. Then, he was awarded with a two-years Fulbright fellowship at California PATH, UC Berkeley. In 2014, he joined the RITS team at INRIA, France. Since 2016, he is with the Research Department at Renault, France. Dr. Milanés has been awarded with the Best Paper Award in three conferences and his PhD has received three major awards. His research interests include autonomous vehicles, vehicle dynamic control, intelligent traffic and transport infrastructures, and vehicle-infrastructure cooperation.



Carlos Flores received his Ph.D. degree in the RITS Team, INRIA (Paris, France) and MINES ParisTech, PSL Research University (France). He is currently working with PATH at University of California, Berkeley. His research interests include automated vehicles, cooperative driving, system modelling and control, and fractional-order calculus.



Fawzi Nashashibi is a senior researcher and the program manager of RITS team of INRIA (France) and a senior researcher and program manager in the robotics centre of MINES ParisTech. He has a Ph.D. degree in robotics from Toulouse University prepared in LAAS/CNRSlaboratory. His current interest focuses on advanced urban mobility through the design and development of highly Automated Transportation Systems.