IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 28 August 2023, accepted 25 September 2023, date of publication 2 October 2023, date of current version 6 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3321269

RESEARCH ARTICLE

Resource-Efficient Path-Following Control for a Self-Driving Car in a Networked Control System

GUILLERMO ALITE[1], ÁNGEL CUENCA[1], JULIÁN SALT[1], AND MASAYOSHI TOMIZUKA[2]

[1]Instituto Universitario de Automática e Informática Industrial, Universitat Politècnica de València (UPV), 46022 Valencia, Spain
[2]Mechanical Engineering Department, University of California at Berkeley, Berkeley, CA 94720, USA

Corresponding author: Guillermo Alite (guialce@doctor.upv.es)

ABSTRACT In recent years, in-vehicle networks are increasingly being incorporated to self-driving cars in order to interconnect spatially distributed devices such as sensors, actuators, and controllers, leading to networked control systems (NCS). The main aim of this work is to reduce the use of resources in a NCS (bandwidth, device batteries) while maintaining an accurate path following for a self-driving car. Some typical network-induced drawbacks such as time-varying delays, packet dropouts and packet disorder will also be coped with. In order to reach the goals, a systematic integration of periodic event-triggered sampling techniques, packet-based control strategies, and state estimation methods is proposed. A novel non-uniform dual-rate extended Kalman filter (NUDREKF) is formulated to estimate the system state at fast, control rate from scarce slow-rate measurements. Due to its mathematical simplicity and low computational cost, the dynamic control law is designed from an inverse kinematic bicycle model and a proportional feedforward controller. Interestingly, optimal parameters for the event-triggered conditions are reached, leading to a satisfactory trade-off between resource savings and control performance. Simulation results for a real trajectory considering actual limitations for the actuators reveal the benefits of the control proposal compared to a conventional control approach.

INDEX TERMS Event-triggered communication, Kalman filter, networked control system, resource efficiency, self-driving car.

I. INTRODUCTION

In the last years, with the growth of artificial intelligence and big data, autonomous vehicles are considerably attracting the scientific community [1], [2]. Their application covers a wide range of fields such as self-driving cars, agriculture, industry, and military tasks. Among their functionalities, it may be stood out obstacle avoidance and trajectory planning [3], motion prediction [4], and path tracking [5]. The present work is focused on a path tracking scenario where time is not considered, becoming a path-following problem.

Due to the development of network communication technologies, in-vehicle networks are increasingly being incorporated to self-driving cars [6]. By means of these specialised internal communication networks, different devices inside a vehicle (sensors, actuators, controllers) can be interconnected, leading to a networked control system (NCS) [7]. The main advantages of this control setup are wiring and weight reduction, easier maintenance, and resource sharing. However, NCS can be affected among others by time-varying delays [8], [9], packet dropouts [10], [11], packet disorder [12], [13], limited communication resources [14], [15], and sensor and actuator attacks [16], [17]. These drawbacks may degrade control performance.

As a consequence of the considerable amount of data that must be transmitted over the in-vehicle network, the occupancy of its resources may be intensified [14], [15]. Then, the use of dual-rate control and estimation techniques [18], [19] may be a viable solution. Assuming a slow rate for sensing and communication, and an M times faster rate for control and actuation, the amount of packets sent through the network, and hence, the bandwidth and battery usage,

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Wei.

can be reduced M times, while keeping an acceptable control performance. In addition, packet disorder can be avoided by defining the slow rate greater than the maximum time delay.

Integrating dual-rate techniques with event-triggered control and communication methods [20] enables to increase resource savings even more in the NCS, since only some of the packets will be sent through the network when some event-triggered conditions are satisfied. Event-triggered control and communication is a trending research area, which is being applied to diverse fields such as self-driving vehicles [15], [21], cyber-physical systems [22], robotic manipulators [23], multi-agent systems [24], hard-disk drives [25], and so on. In the present work, this technique will be used from the point of view of periodic event-triggered communication (PETC) [26], where the event-triggered conditions are periodically evaluated. In this way, the PETC mechanism ensures a minimum inter-event time, which avoids the well-known Zeno behaviour, and makes easier the digital implementation [27]. In our work, using Montecarlo-like methods, optimal parameters for the event-triggered conditions are achieved, which results in an acceptable trade-off between resource savings and control performance.

As a consequence of integrating dual-rate and event-based techniques, the control side will receive scarce and slow-rate data from the plant side. In order to implement a faster-rate dynamic controller to assure a desired control performance, the estimation of the non-available data is necessary. For vehicle dynamic state estimation purposes, different strategies can be used [28]. In this work, the celebrated extended Kalman filter (EKF) [29] is utilised, but including a novel non-uniform dual-rate mode (NUDREKF) with the aim of estimating the vehicle state at fast, control rate from the scarce, non-uniform slow-rate measurement signal. Then, the non-linear nature of the self-driving car and possible Gaussian-like measurement and modelling uncertainties can be coped with in a non-uniform dual-rate sampling scheme. Additionally, the NUDREKF incorporates an h-step ahead prediction stage in order to calculate a control signal with h estimates to be applied in future time instants. The control signal will be sent to the actuators using a packet-based control technique [30]. By using this technique, the set of h fast-rate control values can be sent in a single packet at the slower rate, reducing network usage. When the packet is received at the actuator, it replaces the previous one. If packet dropouts or delays are produced when new packets are sent, or the packets are not sent as a consequence of not satisfying the event-triggered conditions, the h control actions will be applied by the actuator in the corresponding instants. The use of smart actuators is required so as to carry out the described management of the received data.

The NUDREKF provides the path-tracking algorithm (Pure Pursuit [31] in our work) with the estimated position, orientation and velocity of the vehicle. Then, the Pure Pursuit algorithm is able to generate the next dynamic reference to be followed. In addition, the NUDREKF provides the dynamic controller with the estimated velocity and yaw rate. From these estimates and the reference, the dynamic controller is able to compute the steering angle to be applied to the vehicle. In this work, due to its mathematical simplicity and low computational cost, a straightforward dynamic control law based on the inverse kinematic bicycle model and a proportional feedforward controller is proposed. However, for this kind of scenarios, other approaches can be found in literature, as the widely used model predictive controller (MPC) [32], [33], [34], or other methods based on fuzzy and sliding mode control [35], robust H_∞ control [16], etc. These control alternatives usually imply greater mathematical and algorithmic complexity (see, e.g., in [36], where a comparison between MPC and the control law proposed in the present work is made).

To the best of the authors' knowledge, few works dealing with non-uniform dual-rate sampling via EKF can be found in literature. Only in [37] and [38], similar approaches to face delayed measurements are introduced, but not considering autonomous vehicle scenarios. In [39], a NUDREKF is employed for a four-wheeled holonomic robot, but using a more complex mathematical formulation, based on multi-rate system modelling, and not contemplating an NCS environment.

As a summary, the main contributions of the present work are:

- Consideration of a novel NUDREKF, which includes an h-step ahead prediction stage, for path-following control of a self-driving car.
- Integration of the NUDREKF in an NCS framework, using PETC and packet-based control.
- Obtainment of optimal parameters for the event-triggered mechanisms in both network links so as to reach a satisfactory trade-off between resource savings and control performance.

The paper is organised as follows. In Section II, the problem scenario and notation are introduced. In Section III, the kinematic and dynamic model of the self-driving car is presented. In our work, this formulation models the 2017 Lincoln MKZ car [40], which is available at the Mechanical System Control (MSC) Lab, University of California, Berkeley (UCB). In Section IV, the different elements that integrates the control solution are stated. In Section V, some simulations are carried out using Truetime [41] to implement the NCS environment. Optimal parameters for the event-triggered conditions are obtained, and the control proposal is compared to a conventional control solution, where a real trajectory located at Richmond Field Station (UCB) and actual limitations for the actuators are considered so as to give more realism to the simulation. Finally, in Section VI, the main conclusions of the paper and possible future works are highlighted.

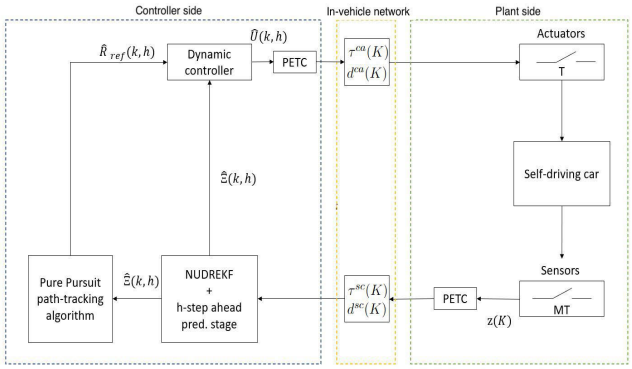

FIGURE 1. Problem scenario.

TABLE 1. Notation.

T	Control period
MT	Sensor period
k	Iteration at T
K	Iteration at MT
h	Prediction horizon
$z(K)$	Vehicle measurements
$\tau^{SC}(K)$	Sensor-to-controller delay
$\tau^{CA}(K)$	Controller-to-actuator delay
$d^{SC}(K)$	Sensor-to-controller dropouts
$d^{CA}(K)$	Controller-to-actuator dropouts
$\hat{\Xi}(k,h)$	Vehicle state estimates
$\hat{R}_{ref}(k,h)$	Path references
$\hat{U}(k,h)$	Control actions

II. PROBLEM SCENARIO AND NOTATION

Figure 1 shows the problem scenario. Notation employed is presented in Table 1. Let us consider the plant side as the network side that includes sensors and actuators of the self-driving car. The controller side is the network side where the control solution is implemented. Both sides are connected via a in-vehicle network.

Although the sensing devices usually work at different rates, let us only take into account the measurements that coincide at the slowest rate, being equivalent to the period MT. The actuators are able to apply control actions at period T under zero order hold (ZOH) conditions. Then, $M \in \mathbb{N}^+$ is the multiplicity between the periods involved in the dual-rate sampling scheme.

Let us denote $k \in \mathbb{N}$ and $K \in \mathbb{N}$ as the current time instant for variables defined at period T and MT, respectively. The event-triggered conditions will be evaluated at MT, and hence, the information can travel through the network at some instants KMT.

As commented, the control solution includes the NUDREKF with the h-step ahead prediction stage. The value of the prediction horizon h depends on the parameters considered in the NCS and can be set by carrying out some previous tests. As a dual-rate scheme, a minimum of M estimates at period T must be generated from an array of measurements taken at period MT. Then, $h \geq M$. The different vehicle variables involved in the control solution will be defined in Section III.

A. TIME-VARYING DELAYS, PACKET DROPOUTS, AND PACKET DISORDER

The round-trip time delay $\tau(K)$ induced by the NCS can be defined as

$$\tau(K) = \tau^{sc}(K) + \tau^{ca}(K) + \tau^{c}(K) \tag{1}$$

where $\tau^{sc}(K)$ is the sensor-to-controller delay, $\tau^{ca}(K)$ is the controller-to-actuator delay, and $\tau^{c}(K)$ is a possible computation delay caused by the control solution, which can be lumped together with the network delays. The delay $\tau(K)$ is assumed to be time-varying in the range $[0, \tau_{max}]$, being τ_{max} the maximum time delay. In this work, in order to avoid packet disorder, the sensing period is chosen to be greater than the maximum time delay, i.e. $MT > \tau_{max}$. The probability density function for the delay $\tau(K)$ is given by

$$P[\tau(K)] = \begin{cases} \dfrac{1}{\phi} e^{\frac{-(\tau(K)-\eta)}{\phi}}, & \tau(K) \geq \eta \\ 0, & \tau(K) < \eta \end{cases} \tag{2}$$

where the expected value of the delay is $E[\tau(K)] = \phi + \eta$, the variance of the delay is $V[\tau(K)] = \phi^2$, and η is the median value of the delay. From these parameters, ϕ can be calculated [42].

Packet dropouts can be modelled by means of a Bernoulli distribution [42]. Considering $d^{sc}(K)$ and $d^{ca}(K)$ respectively as the possible sensor-to-controller and controller-to-actuator dropouts, the probability of dropout for each network link can be expressed as

$$\begin{aligned} p^{sc} &= Pr[d^{sc}(K) = 0] \in [0, 1) \\ p^{ca} &= Pr[d^{ca}(K) = 0] \in [0, 1) \end{aligned} \tag{3}$$

B. PERIODIC EVENT-TRIGGERED COMMUNICATION (PETC)

The PETC mechanism is implemented at both network sides, being evaluated at period MT. Let us respectively consider $\lambda_z(K), \lambda_u(K) \in [0, 1]$ as the scheduling variable at the sensor and controller. When $\lambda_z(K) = 1$, the packet with the measurements at instants KMT, $z(K)$, is transmitted over the sensor-to-controller link, and when $\lambda_u(K) = 1$, the packet with a set of $h+1$ estimated control actions, $[\hat{u}(k), \hat{u}(k+1), \ldots, \hat{u}(k+h)]$, is transmitted over the controller-to-actuator link. Otherwise, when $\lambda_z(K) = 0$ or $\lambda_u(K) = 0$, no packet is transmitted. The last sent sensor data are stored in $\bar{z}(K)$, and the first control action of the last transmitted set is stored in $\bar{u}(K)$. Then

$$\begin{aligned} \bar{z}(K) &= \lambda_z(K) z(K) + (1 - \lambda_z)\bar{z}(K-1) \\ \bar{u}(K) &= \lambda_u(K) u(K) + (1 - \lambda_u)\bar{u}(K-1) \end{aligned} \tag{4}$$

for $K \in \mathbb{N}_{\geq 1}$, being $u(K) = \hat{u}(k)$, and considering $\bar{z}(1) = z(1)$ and $\bar{u}(1) = u(1)$.

The periodic event-triggered condition for the sensor is based on a mixed-triggered mechanism [43], and is evaluated by considering every sensor measurement of the array, $z_i(K)$:

$$\sum_{i=1}^{s_z} \|\bar{z}_i(K-1) - z_i(K)\|^2 > \sum_{i=1}^{s_z} \sigma_{zi}\|z_i(K)\|^2 + \mu_{zi} \quad (5)$$

where s_z=size($z(K)$), and $\sigma_{zi} \in [0,1]$, $\mu_{zi} \in \mathbb{R}^+$, [44]. In the same way, the periodic event-triggered condition for the controller is defined as follows

$$\sum_{i=1}^{s_u} \|\bar{u}_i(K-1) - u_i(K)\|^2 > \sum_{i=1}^{s_u} \sigma_{ui}\|u_i(K)\|^2 + \mu_{ui} \quad (6)$$

being s_u=size($u(K)$), $u_i(K)$ is the i-element of the array $u(K)$ (similarly for $\bar{u}_i(K)$), and $\sigma_{ui} \in [0,1]$, $\mu_{ui} \in \mathbb{R}^+$, [44].

C. CONTROL STRUCTURE

Next, the overall working mode of the control structure is briefly described:

- If the event-triggered condition at the sensors (5) is satisfied, the packet with the vehicle measurements taken at instants KMT, $z(K)$, is transmitted over the sensor-to-controller link.
- From $z(K)$, the vehicle state correction is carried out by the NUDREKF, taking into account the possible delay $\tau^{sc}(K)$. The estimated state is defined at period T, $\hat{\xi}(k)$. At time instants kT between two packet deliveries, or if the packet is lost ($d^{sc}(K) = 0$), the vehicle state prediction is computed by the NUDREKF. More details in Section IV-C.
- From $\hat{\xi}(k)$, the Pure Pursuit path-tracking algorithm can generate the next reference to be followed, $\hat{r}_{ref}(k)$, and then, the dynamic controller can compute the control signal $\hat{u}(k)$. For details, see Sections IV-A and IV-B.
- Then, the h-step ahead prediction stage iterates h times the control loop, resulting in the matrices with $h+1$ estimated references (7), $h+1$ state estimates (8), and $h+1$ estimated control actions (9). More details in Section IV-D.

$$\hat{R}_{ref}(k,h) = \begin{bmatrix} \hat{r}_{ref}(k) \\ \hat{r}_{ref}(k+1) \\ \cdots \\ \hat{r}_{ref}(k+h) \end{bmatrix} \quad (7)$$

$$\hat{\Xi}(k,h) = \begin{bmatrix} \hat{\xi}(k) \\ \hat{\xi}(k+1) \\ \cdots \\ \hat{\xi}(k+h) \end{bmatrix} \quad (8)$$

$$\hat{U}(k,h) = \begin{bmatrix} \hat{u}(k) \\ \hat{u}(k+1) \\ \cdots \\ \hat{u}(k+h) \end{bmatrix} \quad (9)$$

- If the event-triggered condition at the controller (6) is satisfied, the set of control actions $\hat{U}(k,h)$ is transmitted to the smart actuators. When the packet is received, it replaces the previous one, and then, the actuators are able to apply the correspondent control action at period T irrespective of the possible delay $\tau^{ca}(K)$ or future packet dropouts, $d^{ca}(K) = 0$. If the $h+1$ control actions were applied and no new packet were received at the plant side, the actuators would hold the last control action, $\hat{u}(k+h)$, until a new packet were received.

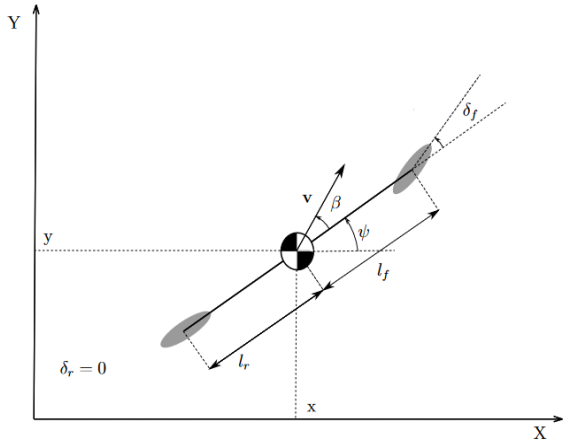

FIGURE 2. Bicycle model [45].

III. SELF-DRIVING CAR MODELLING

A. KINEMATIC MODEL

In order to represent the kinematic model of the self-driving car, let us consider the kinematic bicycle model (see Figure 2):

$$\dot{x} = V\cos(\psi + \beta) \quad (10)$$

$$\dot{y} = V\sin(\psi + \beta) \quad (11)$$

$$\dot{\psi} = r = \frac{V\cos(\beta)}{l_f + l_r}(\tan(\delta_f) - \tan(\delta_r)) \quad (12)$$

$$\beta = \arctan\left(\frac{l_f\tan(\delta_r) + l_r\tan(\delta_f)}{l_f + l_r}\right) \quad (13)$$

where (x, y) are the coordinates of the center of mass in an inertial frame, ψ is the inertial heading, V is the speed of the vehicle, l_f and l_r represent the distance from the center of the mass of the vehicle to the front and rear axles, respectively, and β is the angle of the current velocity of the center of mass with respect to the longitudinal axis of the car. Let us assume the rear steering angle as $\delta_r = 0$. Then, in the sequel, the front steering angle will be $\delta_f = \delta$.

B. DYNAMIC MODEL

With the aim of extending the kinematic model, it is required an equation to relate the derivative of the yaw rate in body frame coordinates of the vehicle, $\dot{r}$, with the lateral tire forces at the front and rear wheels, F_{yf} and F_{yr}, respectively. Different alternatives can be used (see, e.g., [45]). In this work, the Stanford model [46] is chosen:

$$\dot{r} = \frac{ml_f\tan\delta}{I_z}(a_x - rV_y) + \frac{l_fF_{yf}}{I_z\cos\delta} - \frac{l_rF_{yr}}{I_z} \quad (14)$$

$$F_{yf} = -C_{af} \arctan\left(\frac{V_y + rl_f}{\max(V_x, V_{min})}\right) - \delta \quad (15)$$

$$F_{yr} = -C_{ar} \arctan\left(\frac{V_y - rl_f}{\max(V_x, V_{min})}\right) \quad (16)$$

where V_x and V_y are respectively the longitudinal and lateral velocities (to be defined in (17)-(18)), I_z is the yaw inertia, m represents the vehicle's mass, C_{af} and C_{ar} are respectively the tire cornering stiffness of the front and rear axles, V_{min} is the minimum velocity ($V_{min} = 2.23\ m/s$), and a_x is the longitudinal acceleration, which together with δ will be considered as the control signal. In this work, let us assume a_x to remain constant along the simulation.

In order to give more robustness to the control solution, two different versions of the dynamic model have been used: one of them for state estimation by the NUDREKF and the h-step ahead prediction stage, and the other one to represent the plant in simulations. The discrete version at period T of the Stanford model used for state estimation is described by:

$$V_x(k) = V_x(k-1) + T \cdot a_x(k-1) \quad (17)$$

$$V_y(k) = V_y(k-1) + T[\tan(\delta(k-1))(a_x(k-1) - r(k-1)V_y(k-1)) + \frac{F_{yf}}{m\cos(\delta(k-1))} + \frac{F_{yr}}{m} - r(k-1)V_x(k-1)] \quad (18)$$

$$x(k) = x(k-1) + T[V_x(k-1)\cos(\psi(k-1)) - V_y(k-1)\sin(\psi(k-1))] \quad (19)$$

$$y(k) = y(k-1) + T[V_x(k-1)\sin(\psi(k-1)) - V_y(k-1)\cos(\psi(k-1))] \quad (20)$$

$$\psi(k) = \psi(k-1) + T \cdot r(k-1) \quad (21)$$

$$r(k) = r(k-1) + T\left[\frac{ml_f\tan(\delta(k-1))}{I_z}(a_x(k-1) - r(k-1)V_y(k-1)) + \frac{l_f F_{yf}}{I_z\cos(\delta(k-1))} - \frac{l_r F_{yr}}{I_z}\right] \quad (22)$$

where the vehicle model states and the control actions will be respectively defined as:

$$\xi(k) = (V_x(k), V_y(k), x(k), y(k), \psi(k), r(k))$$
$$u(k-1) = (a_x(k-1), \delta(k-1))$$

As commented, in this work, $a_x(k) = a_x(k-1), \forall k$.

The model used to represent the plant in simulations is similar to the previous one (see, e.g., in [45], [47]), but replacing (14) by

$$\dot{r} = \frac{l_f F_{yf}\cos(\delta) - l_r F_{yr}}{I_z} \quad (23)$$

and then, (22) gives place to

$$r(k) = r(k-1) + T\left[\frac{l_f F_{yf}\cos(\delta(k-1)) - l_r F_{yr}}{I_z}\right] \quad (24)$$

Additionally, the lateral acceleration of the vehicle, a_y, is also included

$$a_y(k) = -V_x(k-1)r(k-1) + \frac{F_{yf}\cos(\delta(k-1)) + F_{yr}}{m} \quad (25)$$

IV. RESOURCE-EFFICIENT PATH-FOLLOWING CONTROL SOLUTION

A. PURE PURSUIT PATH-TRACKING ALGORITHM

This algorithm is in charge of computing the yaw rate reference, $\hat{r}_{ref}$, to be followed by the dynamic controller. From an offline predefined path (x_{ref},y_{ref}) formed by n points, a coherent look ahead distance (LAD), and the estimated vehicle position, orientation and longitudinal velocity ($\hat{x}, \hat{y}, \hat{\psi}, \hat{V}_x$) provided by the NUDREKF, the algorithm follows these steps [31]:

1) Taking the index $i = 1..n$ to access up to the n points of the predefined path, the total distance $dist_{tot}(k)$ from the current position ($\hat{x}(k), \hat{y}(k)$) to the target point ($x_{ref}(i), y_{ref}(i)$) is calculated:

$$dist_x(k) = |\hat{x}(k) - x_{ref}(i)| \quad (26)$$

$$dist_y(k) = |\hat{y}(k) - y_{ref}(i)| \quad (27)$$

$$dist_{tot}(k) = \sqrt{(dist_x(k))^2 + (dist_y(k))^2} \quad (28)$$

When $dist_{tot}(k) > LAD$, the reference for the current iteration ($x_{ref}(k), y_{ref}(k)$) is saved:

$$x_{ref}(k) = x_{ref}(i) \quad (29)$$

$$y_{ref}(k) = y_{ref}(i) \quad (30)$$

2) Then, the next yaw rate reference, $\hat{r}_{ref}(k+1)$, is computed:

$$\alpha(k) = \arctan_2\left(\frac{y_{ref}(k) - \hat{y}(k)}{x_{ref}(k) - \hat{x}(k)}\right) - \hat{\psi}(k) \quad (31)$$

$$\hat{r}_{ref}(k+1) = \frac{2\hat{V}_x(k)\sin(\alpha(k))}{dist_{tot}(k)} \quad (32)$$

where the function $\arctan_2$ represents the fourth-quadrant inverse tangent.

B. DYNAMIC CONTROLLER

As a dynamic controller, a control law based on the inverse kinematic bicycle model and a proportional feedforward controller is used [36]:

$$\hat{\delta}(k+1) = [\arctan_2\left(\frac{\hat{r}_{ref}(k+1)L}{\hat{V}_x(k)}\right) + K_p(\hat{r}_{ref}(k+1) - \hat{r}(k))]\gamma \quad (33)$$

where L is the vehicle's longitudinal length (i.e., $L = l_r + l_f$), $\hat{V}_x(k)$ and $\hat{r}(k)$ are provided by the NUDREKF and were respectively defined in (17) and (22), $\hat{r}_{ref}(k+1)$ is provided by the Pure Pursuit path-tacking algorithm in (32), K_p is the gain of the proportional feedforward controller, and γ is a vehicle coefficient that translates the tire angle into the steering angle.

C. NON-UNIFORM DUAL-RATE EXTENDED KALMAN FILTER (NUDREKF)

In this work, a NUDREKF is employed in order to provide the dynamic controller with state estimations at period T from scarce vehicle measurements taken at instants KMT and sent over the network when the periodic event-triggered communication is successful. The filter only corrects its gain and the vehicle states when the data are received. Otherwise, state prediction is performed.

Possible process and measurement noises, $n_1(k)$ and $n_2(k)$, respectively, are coped with. The noises are assumed to be zero mean multivariate Gaussian, being their covariances $Q(k)$ and $R(k)$, respectively.

Let us use the notation $\hat{\xi}(j|i)$ to indicate the state estimates for the instant j computed at the instant i. Then, from the non-linear dynamic model described for the self-driving car in (17)-(22), adding the noises $n_1(k)$ and $n_2(k)$, and synthesizing in this way

$$\begin{cases} \xi(k) = f\left(\xi(k-1), n_1(k-1), u(k-1)\right) \\ z(k) = h\left(\xi(k), n_2(k)\right) \end{cases} \tag{34}$$

the NUDREKF is formulated as follows:

1) Computations at instants $kT \neq KMT$, or $kT = KMT$ when a sensor-to-controller dropout is produced ($d^{sc}(K) = 0$) or the event-triggered condition (5) is not satisfied:

- Prediction of the next state $\hat{\xi}(k|k-1)$ and propagation of the covariance $P(k|k-1)$

$$\begin{aligned} \hat{\xi}(k|k-1) &= f(\hat{\xi}(k-1|k-1), \\ &\quad \times n_1(k-1), \hat{u}(k-1)) \\ P(k|k-1) &= A(k)P(k-1|k-1)A(k)^\top \\ &\quad + L(k)Q(k-1)L(k)^\top \end{aligned} \tag{35}$$

where $(\cdot)^\top$ means transpose function, and

$$\begin{aligned} \hat{\xi}(0) &= E[\xi(0)] \\ P(0) &= E\left[(\xi(0) - E[\xi(0)])\,(\xi(0) - E[\xi(0)])^\top\right] \end{aligned}$$

being $E[\cdot]$ the expectation. Jacobian matrices $A(k)$ and $L(k)$ are respectively used to linearize the process model about the current state and about the process noise

$$\begin{aligned} A(k) &= \left.\frac{\partial f}{\partial \xi}\right|_{\hat{\xi}(k-1|k-1), n_1(k-1), \hat{u}(k-1)} \\ L(k) &= \left.\frac{\partial f}{\partial n_1}\right|_{\hat{\xi}(k-1|k-1), n_1(k-1), \hat{u}(k-1)} \end{aligned} \tag{36}$$

- Shift of the estimated state and covariance

$$\begin{aligned} \hat{\xi}(k|k) &= \hat{\xi}(k|k-1) \\ P(k|k) &= P(k|k-1) \end{aligned} \tag{37}$$

2) Computations at instants $kT = KMT$ when the event-triggered condition (5) holds and no sensor-to-controller dropout is produced ($d^{sc}(K) = 1$):

- Prediction of the output

$$\hat{z}(k) = h\left(\hat{\xi}(k|k-1), n_2(k)\right) \tag{38}$$

- Correction of the filter gain

$$\begin{aligned} \bar{K}(k) = P(k|k-1)H(k)^\top &(H(k)P(k|k-1)H(k)^\top \\ &+ M(k)R(k)M(k)^\top)^{-1} \end{aligned} \tag{39}$$

where the Jacobian matrices $H(k)$ and $M(k)$ enables to respectively linearize the output model about the next predicted state and about the measurement noise

$$\begin{aligned} H(k) &= \left.\frac{\partial h}{\partial \xi}\right|_{\hat{\xi}(k|k-1), n_2(k)} \\ M(k) &= \left.\frac{\partial h}{\partial n_2}\right|_{\hat{\xi}(k|k-1), n_2(k)} \end{aligned} \tag{40}$$

- Correction of the state $\hat{\xi}(k|k)$ and correction of the covariance $P(k|k)$

$$\begin{aligned} \hat{\xi}(k|k) &= \hat{\xi}(k|k-1) + \bar{K}(k)(z(k) - \hat{z}(k)) \\ P(k|k) &= \bar{K}(k)R(k)\bar{K}(k)^\top + (I - \bar{K}(k)H(k)) \\ &\quad \cdot P(k|k-1)(I - \bar{K}(k)H(k))^\top \end{aligned} \tag{41}$$

D. H-STEP AHEAD PREDICTION STAGE

This stage is included in order to generate the h estimated references for $\hat{R}_{ref}(k,h)$ in (7), the h state estimates for $\hat{\Xi}(k,h)$ in (8), and then, the h estimated control actions for $\hat{U}(k,h)$ in (9), which will be applied to the vehicle when no new information is received by the actuators. The algorithm results in an internal control loop, which is h times repeated, and works as follows:

1) From the state $\hat{\xi}(k)$ computed by the NUDREKF and the control signal $\hat{u}(k)$ calculated by the dynamic controller, the next state and output estimations are performed via

$$\begin{cases} \hat{\xi}(k+1) = f\left(\hat{\xi}(k), n_1(k), \hat{u}(k)\right) \\ \hat{z}(k+1) = h\left(\hat{\xi}(k), n_2(k)\right) \end{cases} \tag{42}$$

2) From the state $\hat{\xi}(k+1)$ in (42), the Pure Pursuit path-tacking algorithm generates a new estimated reference $\hat{r}_{ref}(k+1)$, and then, the dynamic controller computes the next estimated control signal $\hat{u}(k+1)$.
3) Iterating the previous two steps $h-1$ times, the matrices $\hat{R}_{ref}(k,h)$, $\hat{\Xi}(k,h)$ and $\hat{U}(k,h)$ are obtained.

V. SIMULATION

A. CASE STUDIES

These are the cases simulated in this section:

- Case 1 (nominal case): This case considers no network, and hence, neither delays nor packet dropouts.

TABLE 2. Parameters used for the model of self-driving car.

$l_f(m)$	$l_r(m)$	$m(kg)$	$C_{af}(kN/rad)$	$C_{ar}(kN/rad)$	$I_z(kg \cdot m^2)$
1.2	1.65	1800	140	120	3270

A conventional control approach actuating and sensing at period T is implemented (M=1). The performance of this control system is considered as the desired one.

- Case 2 (one of the worst cases): In search of resource savings, the control solution proposed in this work is implemented using M=10 and wide thresholds. A relevant reduction of resource usage is reached, but the path-following behavior is noticeably worsened.
- Case 3 (the best case): As a consequence of the previous conclusion, and in order to achieve a satisfactory trade-off between resource savings and control performance, the parameters of the event-triggered mechanism must be suitably set. Then, a Montecarlo-like procedure has been performed to reach optimal parameters. The path-following results for the best case will be shown.
- Case 4 (a realistic case): In this last case, an experiment considering a real trajectory located at Richmond Field Station (a facility of the University of California, Berkeley -UCB-) and including actual limitations for the actuators is presented.

The parameters used for the model of the self-driving car correspond to the 2017 Lincoln MKZ [40], and are shown in Table 2. For the dynamic controller in (33), it is used $K_p = 0.55$ and $\gamma = 1$, and for the NUDERKF, $Q(k) = diag(1 \cdot 10^{-4}, 1 \cdot 10^{-4}, 1 \cdot 10^{-4}, 1 \cdot 10^{-4}, 1 \cdot 10^{-4}, 1 \cdot 10^{-4})$, $R(k) = diag(1 \cdot 10^{-4}, 1 \cdot 10^{-6}, 1 \cdot 10^{-6}, 1 \cdot 10^{-6})$, $\forall k$, where the function $diag(\cdot)$ represents a diagonal matrix formed by the elements detailed as arguments.

B. COST INDEXES

The next cost indexes J_i, $i = 1..5$ are suggested to assess the savings-performance trade-off.

J_1 measures the average deviation from the path followed by the vehicle to the desired trajectory

$$J_1 = \frac{\sum_{k=1}^{l} \min_{1 \le k' \le l} \Omega}{t_{sim}} \tag{43}$$

being

$$\Omega = \sqrt{(x(k) - x_{ref}(k'))^2 + (y(k) - y_{ref}(k'))^2} \tag{44}$$

and where t_{sim} is the total simulation time, and is used to standardize the index; l is the total number of iterations needed to reach the final point of the trajectory; and as known, $(x(k), y(k))$ is the current vehicle position, and $(x_{ref}(k'), y_{ref}(k'))$ is the dynamic reference that is set by the path-tracking algorithm for the current position.

J_2 measures the greatest deviation of the trajectory followed by the vehicle with respect to the desired one

$$J_2 = \max_{1 \le k \le l} \left\{ \min_{1 \le k' \le l} \Gamma \right\} \tag{45}$$

TABLE 3. Weights and targets for J_4.

P_{J1}	P_{J3c}	P_{J3s}	O_{J1}	$O_{J3c}(\%)$	$O_{J3s}(\%)$
1.5	0.75	0.75	30	8	3

being

$$\Gamma = \sqrt{(x(k) - x_{ref}(k'))^2 + (y(k) - y_{ref}(k'))^2} \tag{46}$$

J_3 assesses the reduction of resource usage achieved by the control proposal (which includes PETC) with respect to a conventional, time-triggered control (TTC) working at period T

$$J_3 = \frac{NoT_{\text{PETC}}}{NoT_{\text{TTC}}} 100\% \tag{47}$$

where NoT_{PETC} and NoT_{TTC} are respectively defined as the number of transmissions over the network for the PETC and TTC strategies. Let us distinguish the sensor-to-controller network traffic by means of J_{3s} from the controller-to-actuator one, J_{3c}.

J_4 indicates the global behavior, i.e., a measure of the savings-performance trade-off, by weighting some of the previous indexes and setting some targets for them. In our study, for simplicity, J_1 was used as the representative value for control performance, although in other contexts J_2 could have been chosen (or both of them). Then J_4 is given by

$$J_4 = \frac{\frac{P_{J1}J_1}{O_{J1}} + \frac{P_{J3s}J_{3s}}{O_{J3s}} + \frac{P_{J3c}J_{3c}}{O_{J3c}}}{3} \tag{48}$$

where P_{J1}, P_{J3c} and P_{J3s} are the weights respectively considered for the indexes J_1, J_{3s} and J_{3c}, and O_{J1}, O_{J3c} and O_{J3s} are the targets established for the indexes. If $J_4 \le 1$, the targets are reached, and hence, a satisfactory global behavior can be considered. The lower J_4 is, the better the trade-off will be. Table 3 shows the weights and targets established for the evaluation of J_4. In our simulations, the weights have been set to give more relevance to an accurate path-following behavior instead of resource savings (but the contrary case could have been considered). Concretely, $P_{J1} = 1.5$ (i.e., 50% of relevance) and $P_{J3c} = P_{J3s} = 0.75$ (i.e., 25% of relevance for each of them). Alternatively, other possible weights could have been chosen. From the Montecarlo-like procedure performed in Section V-C3, the targets O_{J1}, O_{J3c} and O_{J3s} have been established as indicated in Table 3 (more details later).

J_5 evaluates comfort that may be experienced by possible passengers riding in the vehicle

$$J_5 = \frac{\sum_{k=1}^{l-1} \|\delta(k+1) - \delta(k)\|}{t_{sim}} \tag{49}$$

where the consecutive variation of the control signal δ has been considered. This index will be only evaluated in Section V-C4 (case 4), where a real scenario with actual limitations on δ has been studied.

TABLE 4. Cost indexes for the case 1 (nominal case).

J_1	$J_2(m)$	$J_{3c}(\%)$	$J_{3s}(\%)$	J_4
16.8172	1.8865	100	100	-

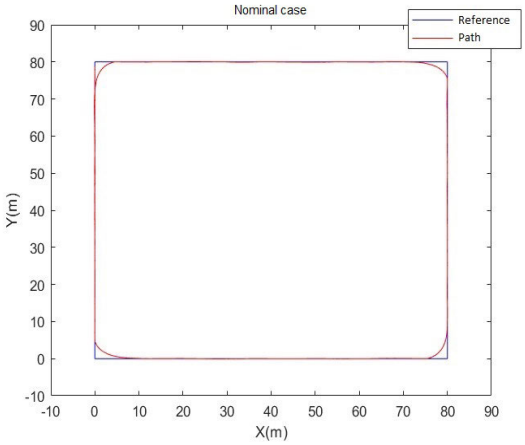

FIGURE 3. Trajectory for the case 1 (nominal case).

TABLE 5. Cost indexes for the case 2 (one of the worst cases).

J_1	$J_2(m)$	$J_{3c}(\%)$	$J_{3s}(\%)$	J_4
119.9259	6.2514	10	0.43636	2.5843

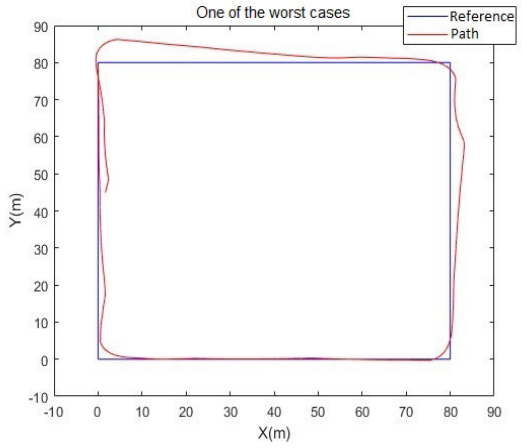

FIGURE 4. Trajectory for the case 2 (one of the worst cases).

C. RESULTS

1) CASE 1 (NOMINAL CASE)

Considering a conventional, time-triggered control at period $T = 0.01s$, and no network, the path-following behavior of the vehicle is depicted in Figure 3, where a squared trajectory is planned. The values obtained for the cost indexes are shown in Table 4 (J_4 is not calculated, since no event-triggered mechanism is used). This is the desired behavior to be compared with the one that will be obtained in cases 2 and 3.

2) CASE 2 (ONE OF THE WORST CASES)

In this case, the control solution proposed in this work is implemented considering $M = 10$, $T = 0.01s$, and $h = 50$, but defining too wide thresholds in the event-triggered mechanisms with the aim of significantly increasing resource savings. Neither delays nor packet dropouts are considered. Figure 4 and Table 5 present the results for this case. As shown, the trajectory followed by the vehicle is considerably far from the planned one. Due to the very scarce information received at the controller side ($J_{3s} = 0.43636\%$, which implies a reduction of around 99.6% with respect to the nominal case), the control solution is not able to reach a satisfactory behavior (J_1 and J_2 are clearly incremented with respect to the nominal case: J_1 around 613%, and J_2 around 231%). However, every packet with control actions is successfully delivered at period MT to the actuator ($J_{3c} = 10\%$, which implies a 90% of reduction with respect to the nominal case), but with not sufficiently updated information. $J_4 > 1$, which confirms a poor savings-performance trade-off.

3) CASE 3 (THE BEST CASE)

In this section, firstly, optimal parameters for the event-triggered mechanisms have been obtained by following a Montecarlo-like procedure. Secondly, some simulations using these optimal parameters have been performed, and the best results for the savings-performance trade-off are depicted.

The PETC conditions (5) and (6) show four different parameters to be set (σ_z, μ_z, σ_u, μ_u). At the same time, σ_z and μ_z include one parameter for each of the four outputs (σ_{zi}, μ_{zi}, $i = 1..4$). In σ_u and μ_u, as the acceleration a_x is assumed to be constant, let us consider only the parameter related to δ. In order to check the effect of every parameter on the NCS, the Montecarlo-like procedure will consider different combinations for these parameters. The combinations are based on keeping constant all the parameters except one of them, which will vary inside a range where the maximum value is 100 times the minimum one.

In the simulations, the NCS considers a delay distribution according to (2), where $E[\tau(K)] = 0.017$, $V[\tau(K)] = 0.01$, $\mu = 0.009s$, $\tau_{max} = 0.064s < MT$ ($T = 0.01s$, $M = 10$), and a packet dropout distribution such as in (3), where $p_{sc} = p_{ca} = 0.25$. As in the previous case, $h = 50$.

After performing the simulation procedure, these are the main conclusions:

1) Effect of the different parameters:
   - As Figure 6 shows, the percentage of packets sent from the controller to the actuator, J_{3c}, is much more affected by μ_u than by σ_u. Concretely, for $\mu_u \in [0.00001, 0.001]$, J_{3c} varies from 2.7% to 8% (see the third subplot in Figure 6). However, as Figure 5 depicts, for $\sigma_u \in [0.005, 0.5]$, J_{3c} varies only from 6.4% to 8% (see the second subplot in Figure 5).
   - Different from the previous conclusion, the percentage of packets sent from the sensors to the controller, J_{3s}, is much more affected by σ_z than

TABLE 6. Optimal values for the PETC conditions.

σ_u	0.05
μ_u	0.00001
σ_z	[0.01, 0.0015, 0.0015, 0.01]
μ_z	[0.1, 0.1, 0.1, 0.1]

by μ_z. For the sake of brevity, let us only show the study done for σ_{z1} and μ_{z1} by means of the Figures 7 and 8. As the second subplot in Figure 7 illustrates, for $\sigma_{z1} \in [0.001, 0.1]$, J_{3s} varies from 0.88% to 4.7%. However, as depicted in the third subplot of Figure 8, for $\mu_{z1} \in [0.01, 1]$, J_{3s} varies only from 1.8% to 2.5%.

2) As expected, when the percentage of sent packets (J_{3s}, J_{3c}) is excessively reduced in search of increasing resource savings, the values for J_1 are considerably incremented, which implies a worse path-following behaviour. In addition, the results become inconsistent (with noticeable variability). This behaviour can be seen, for example, in the first subplot of Figure 7, where for values lower than 2% in J_{3s}, very different values for J_1 are obtained. Unfortunately, most of the simulations are gathered in the inconsistent zone, as shown by the fourth subplot in Figures 5-8. Next, to obtain the targets and optimal parameters, the inconsistent zone will be avoided.
3) Obtainment of the targets O_{J1}, O_{J3s} and O_{J3c} for the cost index J_4 (indicated in Table 3):
   - As shown in the first subplot of Figure 7, J_1 gets more consistent values from $J_{3s} = 3\%$. Then, $O_{J3s} = 3\%$.
   - Similarly, from the first subplot of Figure 6, O_{J3c} may be approximately set as $O_{J3c} = 8\%$.
   - From the previous subplots, and considering the established values for O_{J3s} and O_{J3c}, O_{J1} may be approximately set as $O_{J1} = 30$.
4) Obtainment of the optimal parameters for the PETC conditions:
   - Setting the target O_{J3c} in the second subplot of Figure 5 and in the third subplot of Figure 6, the optimal values for σ_u and μ_u can be respectively deduced such as $\sigma_u = 0.05$, $\mu_u = 0.00001$.
   - Similarly, considering the target O_{J3s} and the Figures 7 and 8, the optimal values for σ_{z1} and μ_{z1} can be obtained as $\sigma_{z1} = 0.01$, $\mu_{z1} = 0.1$. Following a similar procedure (not presented for the sake of brevity), σ_{zi}, μ_{zi}, $i = 2..4$ can also be achieved. Table 6 gathers together all the optimal values.

From the optimal parameters deduced for the event-triggered conditions in Table 6, four different simulations for the NCS have been carried out. A different seed for the packet dropout distribution has been used in every simulation so as to check the consistency of the control solution. Figure 9 shows the best result obtained, which looks very similar to

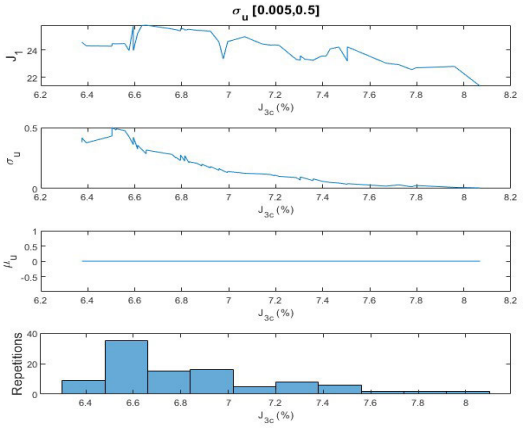

FIGURE 5. Targets and optimal parameters for σ_u.

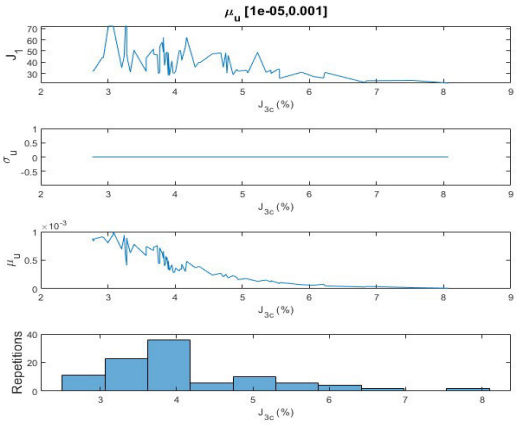

FIGURE 6. Targets and optimal parameters for μ_u.

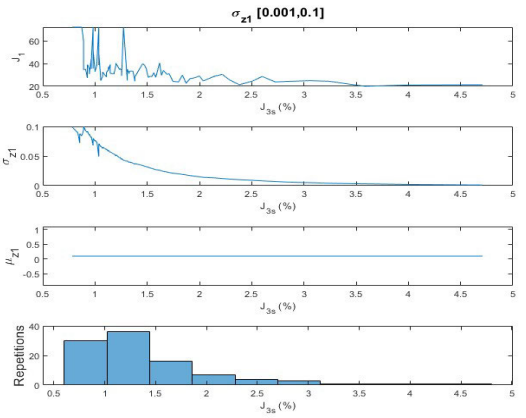

FIGURE 7. Targets and optimal parameters for σ_{z1}.

the nominal result (in Figure 3). The points on the trajectory followed by the vehicle represent the moment where the packet is sent by the sensors. The cost indexes obtained for every simulation are presented in Table 7, where J_1 and J_{3s} clearly fulfill the targets in Table 3, and J_{3c} lightly exceed it. The index J_2 worsens around 9% with respect to the nominal case. The index J_4 obtains similar values in every simulation,

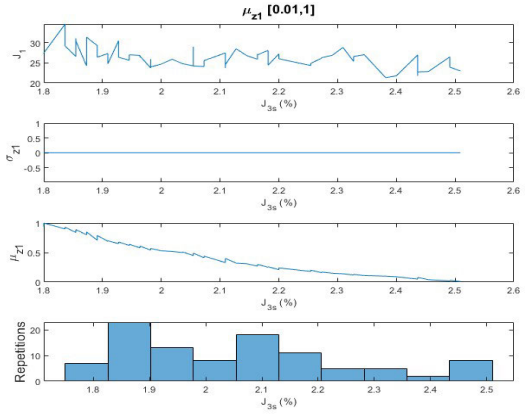

FIGURE 8. Targets and optimal parameters for μ_{z1}.

TABLE 7. Cost indexes for the four simulations.

J_1	$J_2(m)$	$J_{3c}(\%)$	$J_{3s}(\%)$	J_4
21.3808	2.0598	8.087	2.3636	0.8853
22.1588	2.0007	8.1056	2.3636	0.8957
21.6624	2.1205	8.0327	2.3818	0.8883
26.3736	2.0674	8.2331	2.3636	0.9532

TABLE 8. Variation of the cost indexes with respect to the nominal case and the desired targets.

$J_1(\%)$	$J_2(\%)$	$J_{3c}(\%)$	$J_{3s}(\%)$	$J_4(\%)$
27.1361	9.1863	-91.913	-97.6364	-11.47
31.7622	6.0535	-91.8944	-97.6364	-10.43
28.8105	12.4039	-91.9673	-97.6182	-11.17
56.8245	9.5891	-91.7669	-97.6364	-4.68

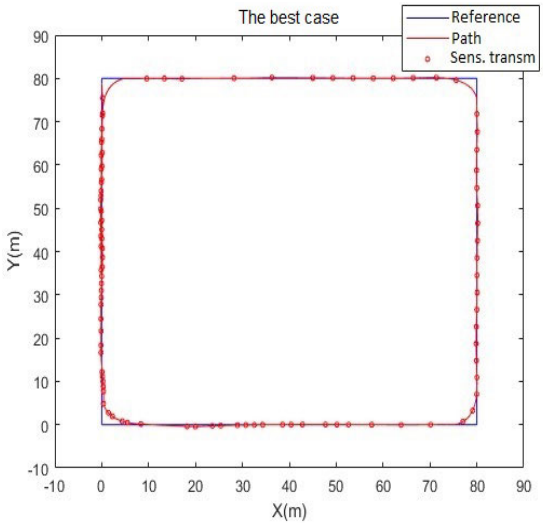

FIGURE 9. Trajectory for the case 3 (the best case).

and as $J_4 < 1$, a satisfactory savings-performance trade-off is always reached.

Table 8 points out the variation of the cost indexes got in the four simulations with respect to those obtained in the nominal case. The best case is easily detected by looking for the greatest negative variation of J_4. For every case, the amount of data travelling through the network has been reduced around

TABLE 9. Cost indexes for the case 4 (a realistic case).

$V_x(m/s)$	J_1	$J_2(m)$	$J_{3c}(\%)$	$J_{3s}(\%)$	J_4	J_5
5	28.9577	3.7151	7.6301	1.7927	0.8705	0.3321
8	37.9523	5.8736	7.6141	2.6434	1.0908	0.3644
12	72.4898	6.6056	7.51	3.7396	1.7548	0.3914

92% in the controller-to-actuator link, and around 97% in the sensor-to-controller link. For the best case, the control performance has been only worsened 27% in J_1 and 9% in J_2. As a conclusion, facing typical network drawbacks such as delays and dropouts, the control solution proposed in this work is able to significantly reduce the resource usage (bandwidth, battery), while maintaining an acceptable path-following behavior.

4) CASE 4 (A REALISTIC CASE)

In order to give more realism to the experiments, a case that considers a real trajectory and actual limitations for the actuators is contemplated. The trajectory used is located at Richmond Field Station (UCB). The actual limitations of the actuators result in the consideration of:

- A range of values for the wheel steering angle, which usually takes $\delta_{max} = 0.32rad$ as the standard maximum angle. So, $\delta(k) \in [-0.32, 0.32]rad$.
- A maximum variation ratio for the steering angle, $rat_\delta(k)$, which is related to the first derivative of δ as follows

$$rat_\delta(k) = \frac{\delta(k) - \delta(k-1)}{T} \tag{50}$$

Let us consider $rat_\delta(k) = 1, \forall k$.

In the experiments, three different velocities have been checked, $V_x = [5, 8, 12]m/s$, assuming $a_x(k) = 0, \forall k$. To better set the *LAD* used by the Pure Pursuit path-tacking algorithm according to the velocity, the next expression is utilized

$$LAD = \frac{s_f V_x^2}{R_{max}} \tag{51}$$

where s_f is a scale factor, and R_{max} is the maximum turning angle calculated as follows

$$R_{max} = \frac{L}{\tan(\delta_{max}) - \cos(\beta)} \tag{52}$$

where L is the vehicle's longitudinal length, and β was defined in (13). The rest of parameters, targets and distributions used in the experiments maintain the same values of the case 3.

In Figures 10-12 and 13-15, the results obtained by using the control solution proposed in this work (red path) are compared to the ones for a conventional control approach (green path). In this conventional approach, no h-step ahead prediction stage is implemented, and the NUDREKF is emulated to have a residual effect, being hardly used to correct the state estimates (an upper M such as $M = 20$ is considered). As shown, the conventional approach is not able to accurately follow the path (the higher the velocity is,

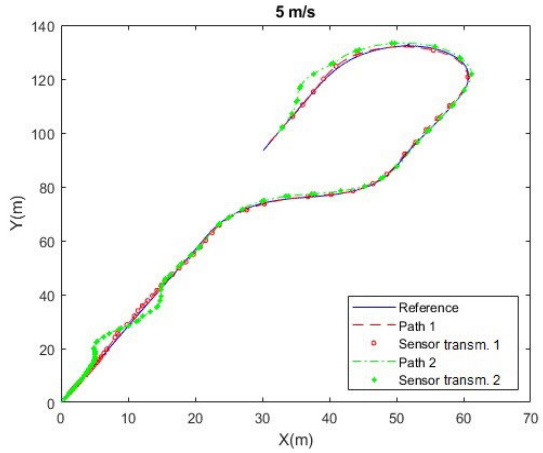

FIGURE 10. Trajectory for the case 4 (a realistic case): 5m/s.

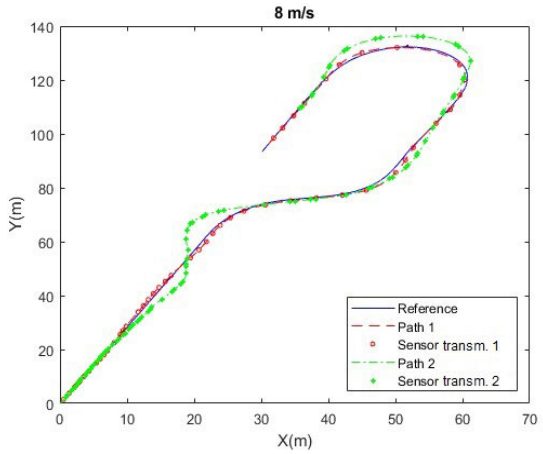

FIGURE 11. Trajectory for the case 4 (a realistic case): 8m/s.

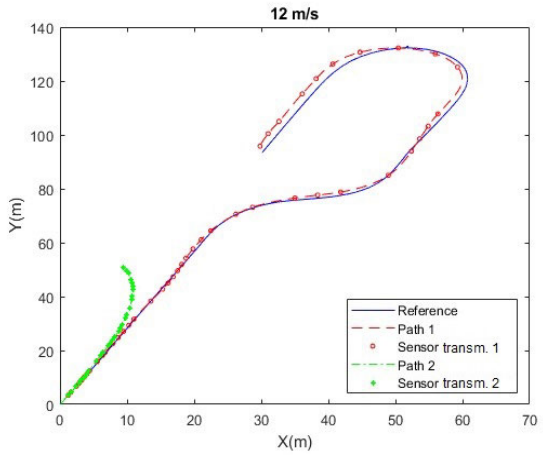

FIGURE 12. Trajectory for the case 4 (a realistic case): 12m/s.

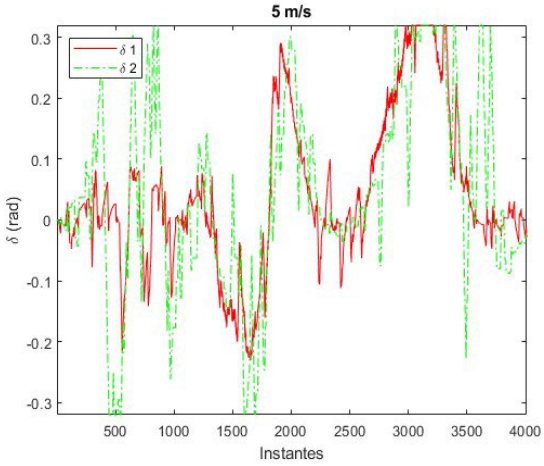

FIGURE 13. Control actions for the case 4 (a realistic case): 5m/s.

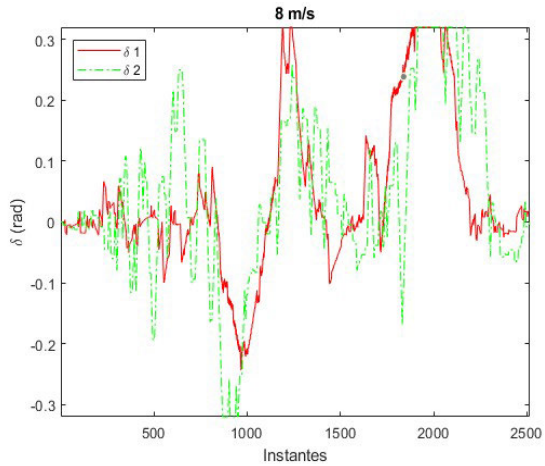

FIGURE 14. Control actions for the case 4 (a realistic case): 8m/s.

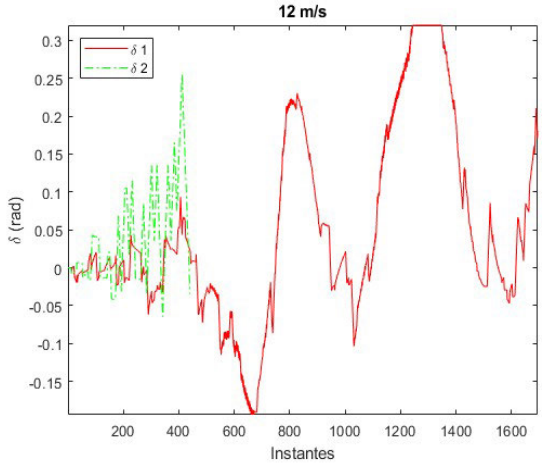

FIGURE 15. Control actions for the case 4 (a realistic case): 12m/s.

the worse the path-following behavior will be), even becoming unstable for the highest velocity ($V_x = 12m/s$). Nevertheless, our control proposal is able to reach an acceptable path-following behavior, despite the actual limitations on the actuators. As expected, the cost indexes J_1 and J_2 in Table 9 obtained for our proposal present better values for lower velocities. In average, the cost indexes J_{3c} and J_{3s} point out around 92.5% and 97.5% of resource savings, respectively. The lowest velocity ($V_x = 5m/s$) reaches the best J_4, which is

less than 1, and hence, for this velocity, the targets are achieved (for $V_x = 8m/s$, they are almost accomplished). As also expected, with the lowest velocity, the best comfort index (J_5) is obtained, which is 18% better than the reached one with the highest velocity.

The control actions depicted in Figures 13-15 show some saturated values when turning 180°, which result in a worse path-following behavior in this path segment. The conventional control solution (green line) shows more fluctuating control values than our control proposal (red line), becoming unstable for the highest velocity.

VI. CONCLUSION

In this work, an NCS approach for a self-driving car is proposed with the aim of achieving a satisfactory savings-performance trade-off. Typical NCS drawbacks such as time-varying delays, packet dropouts and packet disorder are additionally faced. Optimal parameters have been deduced for the periodic event-triggered communication mechanisms. By means of a realistic scenario, the benefits of the control proposal have been revealed in comparison with a conventional control approach.

As future works, some aspects may be extended or included such as throttle and brake pedal control, dynamic event-triggered communication, etc.

CONFLICT OF INTEREST

The authors declare that there is no competing financial interest or personal relationship that could have appeared to influence the work reported in this paper.

REFERENCES

[1] X. Di and R. Shi, "A survey on autonomous vehicle control in the era of mixed-autonomy: From physics-based to AI-guided driving policy learning," *Transp. Res. C, Emerg. Technol.*, vol. 125, Apr. 2021, Art. no. 103008.

[2] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. M. Paixao, F. Mutz, L. D. P. Veronese, T. Oliveira-Santos, and A. F. De Souza, "Self-driving cars: A survey," *Expert Syst. Appl.*, vol. 165, Mar. 2021, Art. no. 113816.

[3] X. Hu, L. Chen, B. Tang, D. Cao, and H. He, "Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles," *Mech. Syst. Signal Process.*, vol. 100, pp. 482–500, Feb. 2018.

[4] P. Karle, M. Geisslinger, J. Betz, and M. Lienkamp, "Scenario understanding and motion prediction for autonomous vehicles—Review and comparison," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 16962–16982, Oct. 2022.

[5] Q. Yao, Y. Tian, Q. Wang, and S. Wang, "Control strategies on path tracking for autonomous vehicle: State of the art and future challenges," *IEEE Access*, vol. 8, pp. 161211–161222, 2020.

[6] J. Huang, M. Zhao, Y. Zhou, and C.-C. Xing, "In-vehicle networking: Protocols, challenges, and solutions," *IEEE Netw.*, vol. 33, no. 1, pp. 92–98, Jan. 2019.

[7] X.-M. Zhang, Q.-L. Han, X. Ge, D. Ding, L. Ding, D. Yue, and C. Peng, "Networked control systems: A survey of trends and techniques," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 1, pp. 1–17, Jan. 2020.

[8] Z. Hu, H. Song, H. Ren, Y. Su, and F. Deng, "Stabilization of networked control systems subject to noisy sampling intervals and stochastic time-varying delays," *IEEE Trans. Control Netw. Syst.*, vol. 9, no. 3, pp. 1271–1280, Sep. 2022.

[9] F. Fang, H. Ding, Y. Liu, and J. H. Park, "Fault tolerant sampled-data H_∞ control for networked control systems with probabilistic time-varying delay," *Inf. Sci.*, vol. 544, pp. 395–414, Jan. 2021.

[10] Z. Hu, T. Luo, R. Yang, and F. Deng, "Stabilization of networked control systems subject to consecutive packet dropouts: The random clock offsets case," *IEEE Trans. Control Netw. Syst.*, vol. 10, no. 1, pp. 285–294, Mar. 2023.

[11] C. Li, X. Zhao, M. Chen, W. Xing, N. Zhao, and G. Zong, "Dynamic periodic event-triggered control for networked control systems under packet dropouts," *IEEE Trans. Autom. Sci. Eng.*, early access, Jan. 19, 2023, doi: 10.1109/TASE.2023.3235375.

[12] D. Liu, Z. Wang, Y. Liu, and F. E. Alsaadi, "Extended Kalman filtering subject to random transmission delays: Dealing with packet disorders," *Inf. Fusion*, vol. 60, pp. 80–86, Aug. 2020.

[13] D. Liu, Z. Wang, Y. Liu, and F. E. Alsaadi, "Recursive filtering for stochastic parameter systems with measurement quantizations and packet disorders," *Appl. Math. Comput.*, vol. 398, Jun. 2021, Art. no. 125960.

[14] H.-T. Sun, P. Zhang, C. Peng, and Y. Zhang, "Neural-network-based event-triggered adaptive security path following control of autonomous ground vehicles subject to abnormal actuator signal," *Int. J. Robust Nonlinear Control*, vol. 33, no. 14, pp. 8275–8289, 2023.

[15] L. Zhang, M. Hu, H. Zhang, Y. Bian, A.-T. Nguyen, and R. Ding, "Channel-level event-triggered communication scheme for path tracking control of autonomous ground vehicles with distributed sensors," *IEEE Trans. Veh. Technol.*, early access, May 8, 2023, doi: 10.1109/TVT.2023.3274111.

[16] M.-Y. Lee and B.-S. Chen, "Robust H_∞ network observer-based attack-tolerant path tracking control of autonomous ground vehicle," *IEEE Access*, vol. 10, pp. 58332–58353, 2022.

[17] H.-T. Sun, C. Peng, X. Ge, and Z. Chen, "Secure event-triggered sliding control for path following of autonomous vehicles under sensor and actuator attacks," *IEEE Trans. Intell. Vehicles*, early access, May 22, 2023, doi: 10.1109/TIV.2023.3278697.

[18] R. Carbonell, Á. Cuenca, V. Casanova, R. Pizá, and J. J. S. Llobregat, "Dual-rate extended Kalman filter based path-following motion control for an unmanned ground vehicle: Realistic simulation," *Sensors*, vol. 21, no. 22, p. 7557, Nov. 2021.

[19] A. Baños, J. Salt, and V. Casanova, "A QFT approach to robust dual-rate control systems," *Int. J. Robust Nonlinear Control*, vol. 32, no. 2, pp. 1026–1054, Jan. 2022.

[20] C. Peng and F. Li, "A survey on recent advances in event-triggered communication and control," *Inf. Sci.*, vols. 457–458, pp. 113–125, Aug. 2018.

[21] W. Li, Z. Xie, J. Zhao, and P. K. Wong, "Velocity-based robust fault tolerant automatic steering control of autonomous ground vehicles via adaptive event triggered network communication," *Mech. Syst. Signal Process.*, vol. 143, Sep. 2020, Art. no. 106798.

[22] A. Fu and J. A. McCann, "Dynamic decentralized periodic event-triggered control for wireless cyber–physical systems," *IEEE Trans. Control Syst. Technol.*, vol. 29, no. 4, pp. 1783–1790, Jul. 2021.

[23] S. E. Benitez-Garcia, M. G. Villarreal-Cervantes, J. F. Guerrero-Castellanos, and J. P. Sánchez-Santana, "Periodic event-triggered control for the stabilization of robotic manipulators," *IEEE Access*, vol. 8, pp. 111553–111565, 2020.

[24] R. Yang, H. Zhang, G. Feng, H. Yan, and Z. Wang, "Robust cooperative output regulation of multi-agent systems via adaptive event-triggered control," *Automatica*, vol. 102, pp. 129–136, Apr. 2019.

[25] Á. Cuenca, M. Zheng, M. Tomizuka, and S. Sánchez, "Non-uniform multi-rate estimator based periodic event-triggered control for resource saving," *Inf. Sci.*, vol. 459, pp. 86–102, Aug. 2018.

[26] K. Li, Q. Liu, and Z. Zeng, "Multiagent system with periodic and event-triggered communications for solving distributed resource allocation problem," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 53, no. 10, pp. 6245–6256, Oct. 2023.

[27] J. Sun, J. Yang, and Z. Zeng, "Predictor-based periodic event-triggered control for nonlinear uncertain systems with input delay," *Automatica*, vol. 136, Feb. 2022, Art. no. 110055.

[28] H. Guo, D. Cao, H. Chen, C. Lv, H. Wang, and S. Yang, "Vehicle dynamic state estimation: State of the art schemes and perspectives," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 2, pp. 418–431, Mar. 2018.

[29] D. Simon, *Optimal State Estimation: Kalman, H_∞, and Nonlinear Approaches*. Hoboken, NJ, USA: Wiley, 2006.

[30] Y.-B. Zhao, G.-P. Liu, Y. Kang, L. Yu, Y.-B. Zhao, G.-P. Liu, Y. Kang, and L. Yu, "Packet-based control design for networked control systems," in *Packet-Based Control for Networked Control Systems: A Co-Design Approach*. Singapore: Springer, 2018, pp. 15–32.

[31] R. C. Coulter, "Implementation of the pure pursuit path tracking algorithm," Robot. Inst., Carnegie-Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. ADA255524, 1992.
[32] H. Wang, B. Liu, X. Ping, and Q. An, "Path tracking control for autonomous vehicles based on an improved MPC," *IEEE Access*, vol. 7, pp. 161064–161073, 2019.
[33] M. A. Daoud, M. W. Mehrez, D. Rayside, and W. W. Melek, "Simultaneous feasible local planning and path-following control for autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16358–16370, Sep. 2022.
[34] C. Sun, X. Zhang, Q. Zhou, and Y. Tian, "A model predictive controller with switched tracking error for autonomous vehicle path tracking," *IEEE Access*, vol. 7, pp. 53103–53114, 2019.
[35] H. Taghavifar and S. Rakheja, "Path-tracking of autonomous vehicles using a novel adaptive robust exponential-like-sliding-mode fuzzy type-2 neural network controller," *Mech. Syst. Signal Process.*, vol. 130, pp. 41–55, Sep. 2019.
[36] J. M. S. Ducajú, J. J. S. Llobregat, Á. Cuenca, and M. Tomizuka, "Autonomous ground vehicle lane-keeping LPV model-based control: Dual-rate state estimation and comparison of different real-time control strategies," *Sensors*, vol. 21, no. 4, p. 1531, Feb. 2021.
[37] A. Gopalakrishnan, N. S. Kaisare, and S. Narasimhan, "Incorporating delayed and infrequent measurements in extended Kalman filter based nonlinear state estimation," *J. Process Control*, vol. 21, no. 1, pp. 119–129, Jan. 2011.
[38] J. Wang, Y. Alipouri, and B. Huang, "Multirate sensor fusion in the presence of irregular measurements and time-varying time delays using synchronized, neural, extended Kalman filters," *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–9, 2022.
[39] R. Pizá, R. Carbonell, V. Casanova, Á. Cuenca, and J. J. S. Llobregat, "Nonuniform dual-rate extended Kalman-filter-based sensor fusion for path-following control of a holonomic mobile robot with four mecanum wheels," *Appl. Sci.*, vol. 12, no. 7, p. 3560, Mar. 2022.
[40] Ford Motor Company. (Jan. 2017). *Technical Specifications for 2017 Lincoln MKZ*. [Online]. Available: https://media.lincoln.com/content/dam/lincolnmedia/lna/us/product/2016/17MKZ-TechSpecs.pdf
[41] A. Cervin, D. Henriksson, B. Lincoln, J. Eker, and K.-E. Arzén, "How does control timing affect performance? Analysis and simulation of timing using Jitterbug and TrueTime," *IEEE Control Syst. Mag.*, vol. 23, no. 3, pp. 16–30, Jun. 2003.
[42] J. Alcaina, A. Cuenca, J. Salt, V. Casanova, and R. Pizá, "Delay-independent dual-rate PID controller for a packet-based networked control system," *Inf. Sci.*, vol. 484, pp. 27–43, May 2019.
[43] D. P. Borgers and W. P. M. H. Heemels, "Event-separation properties of event-triggered control systems," *IEEE Trans. Autom. Control*, vol. 59, no. 10, pp. 2644–2656, Oct. 2014.
[44] L. Xing, C. Wen, Z. Liu, H. Su, and J. Cai, "Event-triggered adaptive control for a class of uncertain nonlinear systems," *IEEE Trans. Autom. Control*, vol. 62, no. 4, pp. 2071–2076, Apr. 2017.
[45] J. M. S. Ducaju, C. Tang, M. Tomizuka, and C.-Y. Chan, "Application specific system identification for model-based control in self-driving cars," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 384–390.
[46] M. Montemerlo et al., "Junior: The Stanford entry in the urban challenge," *J. Field Robot.*, vol. 25, no. 9, pp. 569–597, 2008.
[47] Á. Domina and V. Tihanyi, "Modelling the dynamic behavior of the steering system for low speed autonomous path tracking," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2019, pp. 535–540.

GUILLERMO ALITE received the B.Sc. and M.Sc. degrees in industrial technology engineering and the M.Sc. degree in industrial automation and informatics from the Technical University of Valencia (UPV), Valencia, Spain, in 2018, 2020, and 2022, respectively, where he is currently pursuing the Ph.D. degree in event-triggered networked control for autonomous vehicles. He is working in a tech company in the automotive industry.

ÁNGEL CUENCA received the M.Sc. degree in computer science and the Ph.D. degree in control engineering from the Technical University of Valencia (UPV), Valencia, Spain, in 1998 and 2004, respectively. He was a Visiting Scholar with the Lund Institute of Technology, Lund, Sweden, in 2008, the North Carolina State University, Raleigh, NC, USA, in 2012, the Eindhoven University of Technology, Eindhoven, The Netherlands, in 2014, and the University of California at Berkeley, Berkeley, CA, USA, in 2016 and 2018. He is currently an Associate Professor with the Department of Systems Engineering and Control, UPV. He has coauthored more than 60 technical articles in journals and conferences. His research interests include networked and event-triggered control systems, multirate control systems, and autonomous vehicles.

JULIÁN SALT received the M.Sc. degree in industrial engineering and the Ph.D. degree in control engineering from the Technical University of Valencia (UPV), in 1986 and 1992, respectively. Since 1987, he has been a Full Professor of automatic control with UPV, teaching a wide range of subjects in the area from continuous and discrete simulation to automation and programmable logic controller applications. He was a Visiting Scholar in multirate control of hard disk drives with UC Berkeley, USA. He is currently the Head of the Department of Systems Engineering and Control, UPV. He has been the director of eight Ph.D. dissertations. He has taken part in research projects funded by local industries, the Spanish Government, and the European Science Foundation. He has coauthored more than 90 technical articles in journals and technical meetings. His research interests include non-conventionally sampled control systems and network-based control systems. His research activity is currently involved in network-based control energy saving.

MASAYOSHI TOMIZUKA received the B.S. and M.S. degrees in mechanical engineering from Keio University, Tokyo, Japan, in 1968 and 1970, respectively, and the Ph.D. degree in mechanical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA, in February 1974.

In 1974, he joined as a Faculty Member with the Department of Mechanical Engineering, University of California at Berkeley, Berkeley, CA, USA, where he currently holds the Cheryl and John Neerhout, Jr., Distinguished Professorship Chair. His current research interests include optimal and adaptive control, digital control, motion control, and their applications to robotics and vehicles.

Dr. Tomizuka is a Life Fellow of the ASME and a fellow of the International Federation of Automatic Control (IFAC) and the Society of Manufacturing Engineers. He was a recipient of the Best Journal of Dynamic Systems, Measurement and Control Best Paper Award, in 1995 and 2010, the ASME/DSCD Outstanding Investigator Award, in 1996, the Charles Russ Richards Memorial Award (ASME), in 1997, the Rufus Oldenburger Medal (ASME), in 2002, the John R. Ragazzini Award (AACC), in 2006, the Richard Bellman Control Heritage Award (AACC), in 2018, the Honda Medal (ASME), in 2019, and the Nathaniel B. Nichols Medal (IFAC), in 2020. He served as the Program Director of the Dynamic Systems and Control Program of the Civil and Mechanical Systems Division of NSF, from 2002 to 2004. He served as a Technical Editor of the *Journal of Dynamic Systems, Measurement, and Control* and *Journal of Dynamic Systems, Measurement and Control*, from 1988 to 1993, and the Editor-in-Chief of the IEEE/ASME TRANSACTIONS ON MECHATRONICS, from 1997 to 1999.