Communication-Aware Bandwidth-Optimized Predictive Control of Motor Drives in Electric Vehicles

Davide Quaglia, *Member, IEEE*, and Riccardo Muradore, *Member, IEEE*

***Abstract*—To address the ever growing complexity of electric vehicles, several control functions can be integrated over a single shared intravehicle network, e.g., controller area network (CAN). However, sharing communication resources may lead to communication delays that affect control performance. Well-known communication standards allow to introduce different quality-of-service guarantees on a message-by-message basis. This study focuses on predictive control of motor drives; the model predictive control (MPC) approach is extended to jointly decide the value of each control command and its transmission priority on the CAN bus. A mixed integer quadratic problem is derived in which the transmission strategies are modeled by a two-class priority scheme and the corresponding values of delay variation are transformed into loss probabilities by introducing receiver buffers. System-level simulations show that the proposed approach uses high priority when the system behavior is farther from the desired one or bus is congested and the tracking error is reduced with respect to the traditional MPC and unprioritized transmission.**

***Index Terms*—Automotive, Bernoulli distribution, controller area network (CAN), differentiated services, in-vehicle network, intravehicle network, jitter, message drop probability, message loss, model predictive control (MPC), playout buffer, quality-of-service (QoS), transmission delay, transmission priority.**

I. Introduction

ELECTRIC vehicles (EVs) are at the center of a highly connective industry that focuses on serving mobility needs under the aspect of sustainability with a vehicle using a portable energy source and an electric drive that can vary in the degree of electrification [1].

The evolution of EVs meets the one of traditional vehicles in which more and more control activities are integrated and coordinated by using *intra-vehicle networks*. A huge literature focuses on the performance evaluation and optimized design of intravehicle control networks [2]–[4]. Intravehicle network is crucial when it is used to close a control loop. For instance, this paper addresses the scenario depicted in Fig. 1 in which the motor drive of an EV is controlled through an intravehicle network. The electric control unit compares the desired reference speed with its actual value measured on the wheels and sets the required torque to minimize the error.

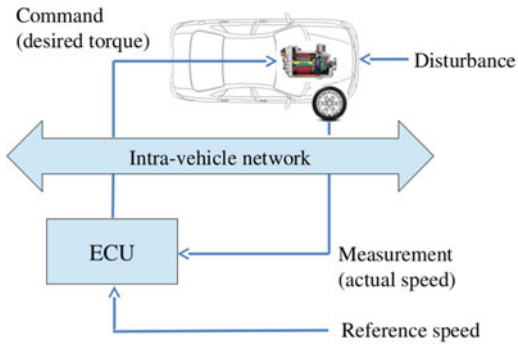


Fig. 1. EV with distributed motor drive controller.

The main issue in assuring controllability in this kind of system is related to the presence of a shared packet-based network that may introduce time-varying communication delay. The research was mainly focused on the controller design by adapting the linear quadratic (LQ) approach and the model predictive control (MPC) [5], [6]. The crucial point is about the analytical model of the communication issues. In [7] and [8], packet dropout has been modeled as a Bernoulli random process. In [9], a method for determining a worst-case upper bound on the controller area network (CAN) delays is presented, which enables the usage of a polytopic approximation technique to obtain a discrete-time model of the closed-loop CAN system.

We address the control problem on a shared network by considering a smart communication architecture where different transmission strategies are provided by the adopted protocol. This approach recalls techniques to introduce quality-of-service (QoS) guarantees in IP networks such as the differentiated services (DiffServ) architecture (see [10]) according to which packets are marked depending on their importance and sent through the network by using either a high-priority low-loss class, or a regular, unguaranteed class. DiffServ architecture has been proposed initially for multimedia communications (see [11]) and recently extended to networked control systems (e.g., [12]). Such architecture can be easily mapped onto CAN networks where message ID can be used as priority mark. In [13], the proposed torque-vectoring control has been validated through a simulation model that includes CAN-based communication

Manuscript received June 26, 2015; revised September 27, 2015, November 23, 2015, January 29, 2016, and February 20, 2016; accepted February 27, 2016. Date of publication April 27, 2016; date of current version August 9, 2016.

The authors are with the Department of Computer Science, University of Verona, 37129 Verona, Italy (e-mail: davide.quaglia@univr.it; riccardo.muradore@univr.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIE.2016.2558485

delays. Such delays are not considered by the formulation of the control strategy. In [14], Shuai *et al.* proposed an H_∞-based delay-tolerant LQ regulator control method. The problem is described in the form of an augmented discrete-time model with uncertain elements determined by the delays. However, message priority is not assigned by the control strategy. In [15], Shuai *et al.*, presented a dynamic message priority scheduling procedure, which evaluates each message's importance and dynamically assigns CAN IDs to them based on the vehicle states and control inputs in real time. Command values and priorities are assigned in two separate steps, thus, reducing optimization.

The contributions of this paper are as follows:

1) a new control architecture that explicitly takes into account the importance of the message content and the status of the communication channel in order to assign high/low priority;
2) a stochastic MPC controller where signals (states and commands) and binary conditions (high/low priority) are considered within the same performance index and jointly optimized;
3) the same channel bandwidth is used in an optimal way.

The starting point of this study is [16] that we improved in the following aspects strictly related with the vehicular scenario:

1) a generalized control architecture that separates the speed control subsystem and the power control subsystem (hierarchical control);
2) simulation results with a model of the permanent-magnet synchronous machine (PMSM);
3) a more complex control task (i.e., tracking instead of regulation to zero);
4) the use of the standard priority mechanism provided by CAN; the corresponding delay variation is transformed into a constant and known delay by using a buffering technique, which introduces some loss probability that simplifies the design of a stochastic MPC where packet loss probability is modeled as Bernoulli process.

This paper is organized as follows. Section II introduces the concepts involved in this study and the corresponding literature. In Section III, the MPC problem is stated with reference to QoS-based transmission. The solution of the problem based on the mixed-integer quadratic programming (MIQP) is described in Section IV. Simulation results are reported in Section V. Finally, conclusions are drawn in Section VI.

II. Background and State of the Art

A. Intravehicle Networks

Automotive systems are rapidly advancing in complexity and diversity [17]. A multitude of sensors and processors are used in different parts of the vehicle for various functions, e.g., antilock braking system, camera, and distance sensors to support drivers. This fact leads to the development of intravehicle network standards for the efficient interconnection of the various systems [17]. Among them, CAN [18] is adopted in the following discussion because of its widespread use for the most critical control tasks in vehicles.

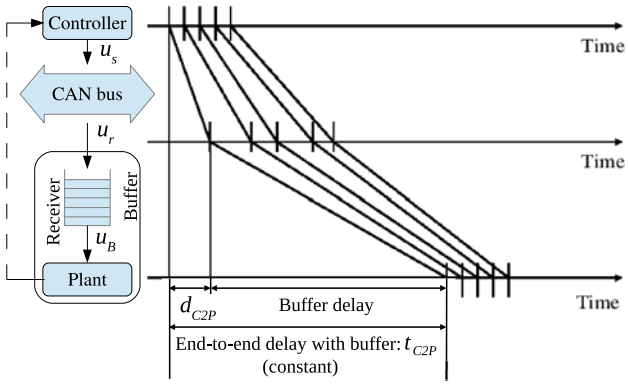


Fig. 2. Transmission architecture with antijitter buffer at the receiver side and the corresponding effect on end-to-end delay.

CAN is a shared bus. When the line is free and two or more nodes want to transmit, a bit-wise arbitration scheme is used following a wired-AND behavior with dominant (0) and recessive (1) bit symbols. The transmission starts with the message identifier from the most significant to the least significant bit. The node transmitting the message with the highest identifier has to send a recessive 1 bit that is overwritten by a dominant 0 bit. Therefore, it notices the collision and stops sending. Therefore, the message identifiers create an implicit hierarchy of priorities. The nodes losing the contention will retry again when the line returns free, thus, increasing their transmission delay.

This kind of medium access control does not provide deterministic guarantees on transmission delay and a huge literature has been devoted to evaluate the worst-case delay and its statistical distribution [19]. All agree on the fact that the smaller is the priority, the larger is delay variation. This variability is a well-known challenge for networked control of motor drives in EVs [9], [13]–[15].

B. Antijitter Buffer

A possible way to eliminate delay variation consists in implementing an antijitter buffer, i.e., a first-in-first-out queue, which stores received messages and extracts them with a constant periodicity. In this way, transmission delay variations are compensated at the cost of an increased (but constant and known) delay. Such approach is quite common in multimedia and also within the control community, e.g., [20].

Fig. 2 shows the classic transmission architecture with antijitter buffer at the plant side. Let T_s be the sample time of the system. Each command is sent on the network in a different data packet at $t = kT_s$, $k \in \mathbb{N}$. Let u_s and u_r be the commands sent/received to/from the network, respectively. Let d_{C2P} be the delay in the controller-to-plant path. The following equation holds:

$$u_r(t) = u_s(t - d_{\mathrm{C2P}}).$$

As in multimedia, to reduce the probability of underflow/overflow, buffers are partially filled at the beginning (prebuffering), and then, packets are extracted at $1/T_s$ rate. The

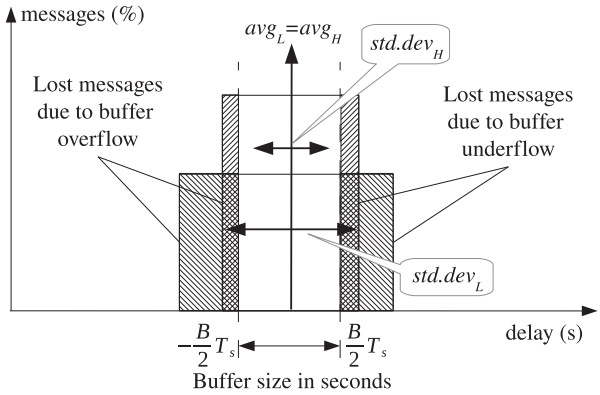


Fig. 3. Relationship between delay distribution, buffer size, and loss probability as a function of CAN message priority (low L or high H).

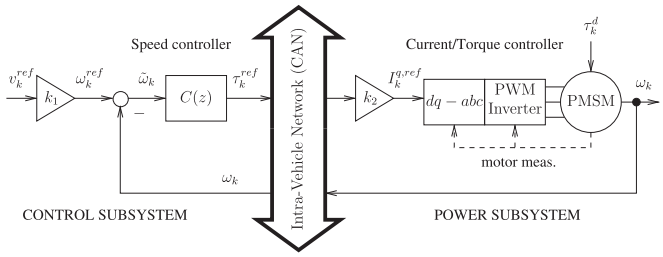


Fig. 4. Block diagram of the speed control architecture.

presence of the buffer makes the total delay constant but larger than the actual network delay because of the presence of the prebuffering time. Let u_B be the commands extracted from the buffer (of size B). The following equation holds:

$$u_B(t) = u_s(t - \text{avg}(d_{\text{C2P}}) - t_{\text{PRE}}) \tag{1}$$

where $\text{avg}(d_{\text{C2P}})$ is the average value of the controller-to-plant delay, while t_{PRE} is the prebuffering time usually equal to $\frac{B}{2}T_s$. The same reasoning also holds for the plant-to-controller path, thus, leading to a constant overall loop delay that simplifies the design of the controller.

A message is dropped if either its delay exceeds the current buffer level (buffer underflow) or it finds the buffer full (buffer overflow). Therefore, loss probability depends on both buffer size and standard deviation of delay distribution as depicted in Fig. 3 for two different CAN priorities and buffer size B (leading to prebuffering time $\frac{B}{2}T_s$). Buffer size B should be carefully chosen as a tradeoff between maximum tolerated constant delay and loss probability.

C. Electric Motor in EVs

In EVs, different kinds of electric motor can be used, e.g., the dc brushless PMM drive, the PMSM motor drive (e.g., Toyota, Lexus), the induction motor drive (e.g., Tesla), and the switched reluctance drive.

The system in Fig. 1 is equivalent to the block diagram in Fig. 4 where the *power subsystem* consists of the PMSM motor, the PWM, the inverter, and the vector control drive, whereas the *control subsystem* contains the speed controller $C(z)$. The speed control subsystem receives the speed error $\tilde{\omega}$ and sends to the other side of the network the reference torque τ^{ref}. The constant k_1 maps the desired linear speed of the car v^{ref} into the desired angular velocity of the motor ω^{ref} (taking also into account the gears). The constant k_2 maps the reference torque into the reference current $I^{q,\text{ref}}$ for the vector control drive.

The equation of motion of a basic vehicle model mapping forces into the vehicle velocity v takes the form

$$M_{\text{car}}\dot{v} = F_{\text{trac,wheel}} - F_{\text{aero}} - F_r - F_{\text{grav}}$$

where M_{car} is the unknown time-varying mass of the vehicle, $F_{\text{trac,wheel}}$ is the force due to the electric motor torque, F_{aero} is the aerodynamic drag force, F_r is the rolling resistance, and F_{grav} is the force due to the gravity [21]. The motor's electrical subsystem cannot disentangle the mechanical effects due to its rotor and to the car. The equivalent equation of motion *seen* by the electric motor side is

$$J\dot{\omega} + B\omega = \tau^{\text{ref}}$$

where ω is the angular velocity, J is the sum of the rotor inertia and the car mass *as seen* by the electric motor, B is the rotor damping coefficient and the car frictions *as seen* by the electric motor, and τ^{ref} is the torque applied by the motor. Unfortunately it is not possible to have reliable values for J and B because they depend on many unknown and time-varying variables. It is also rather risky to estimate all these parameters at run time. The solution is to use conservative coefficients J_m and B_m for J and B, and to model uncertainty and disturbances through the exogenous input τ^d in (2).

With this approach, the mechanical subsystem seen by the PMSM is described by

$$J_m\dot{\omega} + B_m\omega = \tau^{\text{ref}} - \tau^d. \tag{2}$$

It is worth highlighting the following points.

1) An MPC controller based on a simple model is suitable to be executed in real time in today's embedded platforms.
2) The MPC can consider a time-varying model. This means that if robust and reliable estimators can be used for the unknown coefficients, it is well possible to update J_m and B_m at run time.

Since the dynamics of the mechanical subsystem dominates the dynamics of the electrical subsystem, in the following, the design of the controller $C(z)$ will be based on the discrete-time state-space representation of the system in (2)[1]

$$P(z): \begin{cases} x_{k+1} = Ax_k + Bu_k \\ y_k = x_k \end{cases} \tag{3}$$

where $x_k = \omega_k$ and $u_k = \tau_k^{\text{ref}}$. At each sample time kT_s, the controller has to provide the torque commands u_k to the current/torque controller of the motor and receives measurements y_k from the sensors. These signals are exchanged through the intravehicle network as shown in Figs. 1 and 4.

[1]Without τ_k^d that is unknown.

In Section V, we will show that the controller $C(z)$ based on $P(z)$ works well even though the plant model is an approximation of the real power subsystem.

Remark 1: It is worth highlighting that the approximation of the system with a first-order mechanical system shows that the proposed control approach is independent of the motor type. It is always possible to model the vehicle, the electric motor and its inner controller as (2).

D. Review of MPC

For the linear plant (3) without taking into account the network, the quadratic cost function $J_{\mathrm{MPC}}(\cdot)$ is defined as

$$J_{\mathrm{MPC}}(k) = \sum_{i=0}^{N_p} \|\hat{x}_{k+i|k}\|^2_{Q_i} + \sum_{i=0}^{N_c} \|\hat{u}_{k+i|k}\|^2_{R_i} \tag{4}$$

where $Q_i > 0$, $R_i > 0$ are weighting matrices, N_p is the prediction horizon, N_c is the control horizon, and $\hat{x}_{k+i|k}$ and $\hat{u}_{k+i|k}$ are the i-ahead predictors of the state and of the command, respectively. It is common to assume $N_c \leq N_p$ and $\hat{u}_{k+i|k} = 0$ for $i \geq N_c$). The MPC control u_k at time k is computed based on the following algorithm ([22]):

1) get the new state measurement x_k;
2) solve the constrained optimization problem

$$\begin{aligned}\{\hat{u}^\star(k+\cdot|k)\} = \ & \arg \min_{\{\hat{u}(k+\cdot|k)\}} J_{\mathrm{MPC}}(k) \\ & \text{subject to: } m_u \leq \hat{u}_{k+\cdot|k} \leq M_u \\ & m_x \leq \hat{x}_{k+\cdot|k} \leq M_x\end{aligned}$$

where $\{m_u, M_u\}$ and $\{m_x, M_x\}$ are upper and lower bounds for the input and the state, respectively;
3) set $u_k = \hat{u}^\star_{k+0|k}$.

By rewriting the evolution of the system over the optimization horizon N_p as

$$\hat{X}(k) = \mathcal{A}x_k + \mathcal{B}\hat{U}(k)$$

where $\hat{X}(k)$ and $\hat{U}(k)$ are column vectors collecting $\hat{x}_{k+i|k}, i = 0, \ldots, N_p$ and $\hat{u}_{k+i|k}, i = 0, \ldots, N_c$, respectively,

$$\hat{X}(k) := \ \mathsf{colvec}\left\{\hat{x}_{k+i|k}\Big|_{i=0}^{N_p}\right\}$$

$$\hat{U}(k) := \ \mathsf{colvec}\left\{\hat{u}_{k+i|k}\Big|_{i=0}^{N_c}\right\}$$

and $\mathcal{A}$, $\mathcal{B}$ are appropriate block matrices, the cost function becomes

$$\begin{aligned}J_{\mathrm{MPC}}(k) = \ & \hat{U}^T(k) \underbrace{\left[\mathcal{B}^T \mathcal{Q}\mathcal{B} + \mathcal{R}\right]}_{\mathcal{H}} \hat{U}(k) \\ & + \underbrace{2x_k^T \mathcal{A}^T \mathcal{Q}\mathcal{B}}_{\mathcal{G}_k} \hat{U}(k) + \underbrace{x_k^T \mathcal{A}^T \mathcal{Q}\mathcal{A}x_k}_{\mathcal{K}_k}\end{aligned} \tag{5}$$

where $\mathcal{Q} = \mathrm{diag}\{Q_i\}_{i=0}^{N_p}$, $\mathcal{R} = \mathrm{diag}\{R_i\}_{i=0}^{N_c}$ and the last term $\mathcal{K}$ is independent of $\hat{U}(k)$. The constraints on the command and on the state can be easily written as

$$\underline{U} \leq \hat{U}(k) \leq \bar{U}, \ \underline{X} - \mathcal{A}x_k \leq \mathcal{B}\hat{U}(k) \leq \bar{X} - \mathcal{A}x_k \tag{6}$$

where $\underline{U}$, $\bar{U}$, $\underline{X}$, and $\bar{X}$ are matrices containing the upper and lower values. The optimal control $\hat{U}^\star(k)$ on the horizon $[0, N_c]$ is computed by solving the constrained quadratic programming problem

$$\begin{aligned}\hat{U}^\star(k) = \ & \arg \min_{\hat{U}(k)} \ \hat{U}^T(k)\mathcal{H}\hat{U}(k) + \mathcal{G}_k\hat{U}(k) \\ & \text{s. to } (6).\end{aligned}$$

The optimal control $\hat{u}^\star_k$ at time k is the first component in $\hat{U}^\star(k)$, i.e., $\hat{u}^\star_{k|k}$.

This study is based on the following stability assumption that is satisfied in the present scenario:

Assumption 2: The matrix A is asymptotically stable.

Under this assumption, it is possible to show that by setting $Q_1 = \cdots = Q_{N_p} \geq 0$ and Q_{N_p} equal to the solution of the Lyapunov equation $A^T P A - A = Q$, the closed loop system with the aforementioned MPC controller is asymptotically stable (for details, see [23]–[26]). Other approaches to guarantee stability when A is not stable can be found in [22] and [27].

Moreover, it is assumed without loss of generality that

Assumption 3: The matrix B has full column rank.

The prediction of state and command values is the basis of the design of the optimal MPC controller; it could be improved by knowing the reliability level of the network upon which commands and measurements are sent. This original contribution of the study will be described in Section III.

Remark 4: The aforementioned formulation refers to the regulation problem (i.e., steering the state to zero) and not to the tracking problem (i.e., steering the tracking error to zero). The second case, more interesting in our scenario, has the following performance index

$$J_{\mathrm{MPC}}(k) = \sum_{i=0}^{N_p} \|\hat{x}_{k+i|k} - x^{\mathrm{ref}}_{k+i}\|^2_{Q_i} + \sum_{i=0}^{N_c} \|\hat{u}_{k+i|k}\|^2_{R_i} \tag{7}$$

where x^{ref}_{k+i} is the reference state over the horizon N_p. It is easy to see that this problem can be cast in the regulation problem by using $\tilde{x}_k = x_k - x^{\mathrm{ref}}_k$ in (5) instead of x_k if $x^{\mathrm{ref}}_{k+i} = x^{\mathrm{ref}}_k$ $\forall i \in [0, N_p]$, [22].

E. Differentiated Services on the Network

With reference to Fig. 4, when the network is congested, commands τ^{ref}_k and measurements ω_k may be affected by transmission delay, which compromises stability and performance. The differentiated services network architecture can be used to control such congestion-induced delays without the need to increase the network bandwidth. It consists in assigning a different forwarding priority to each packet. Without loss of generality, this study deals with two transmission priorities, i.e., H and L. Medium access control (e.g., in CAN) and intermediate network systems (e.g., IEEE 802.1 AVB/Ethernet switches) favor the forwarding of H packets that, therefore, experience lower delay than L packets. The control of the fraction of H packets on the overall traffic amount is crucial for the success of the mechanism; clearly, if all packets are sent with high priority, the undifferentiated case is reestablished.

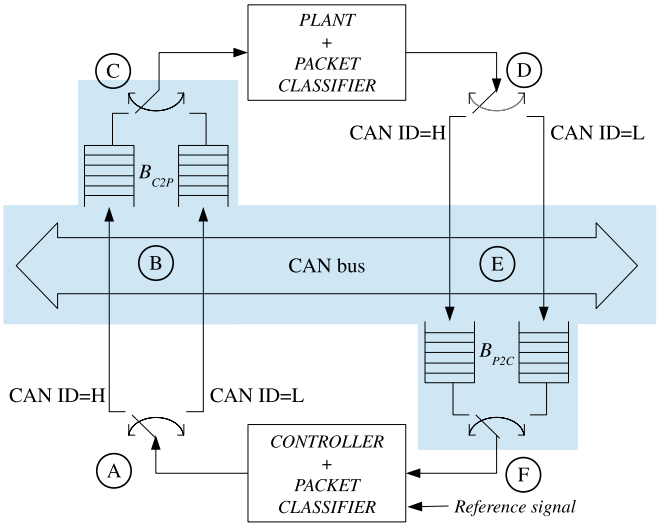


Fig. 5. Proposed architecture for the distributed control of motor drive with Differentiated Services communication approach.

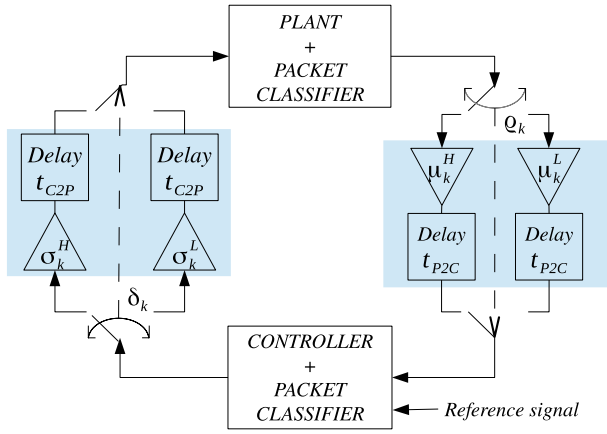


Fig. 6. Abstract block diagram of the proposed distributed control of motor drive with differentiated services network.

The exploitation of transmission priority to improve control performance was used in past literature. In [28], CAN priorities are preassigned to classes of control flows according to their importance and, within each class, earliest-deadline criterion is used to further differentiate priority assignment. This approach does not use the content of each specific message to decide its priority. In [15], CAN priority is assigned to each control command according to its relative importance among five concurrent flows devoted to the control of four independent wheel drives; nevertheless, control command and priority are decided in two independent steps.

In the proposed work, each message (i.e., commands or measurements) is marked and transmitted as either high- or low-priority packet. From the control perspective, packets sent with the H policy lead to better control performance than packets sent with the L policy, provided that the H fraction is kept low. Therefore, the optimal marking strategy can be formalized by assigning a cost to each priority and minimizing the total cost under performance constraints. Clearly, to maximize performance, the most important packets should be sent with the H policy. In this study, the MPC approach is extended to jointly decide the value of each control command and its transmission priority by considering its importance, the effect of priority assignment on its delivery and the instantaneous condition of the network (which implicitly depends on other concurrent flows).

III. Problem Statement

Fig. 5 shows the proposed architecture for the distributed control of motor drive with differentiated services communication approach. Two priority classes are used, i.e., H and L for both controller-to-plant and plant-to-controller paths. At the controller side, the command value and its priority are chosen by using an optimization process, which minimizes tracking error and the use of high-priority bandwidth. The priority is used to set the CAN ID of the corresponding message (labels A and D in Fig. 5). CAN messages with higher priority have more opportunities to access the bandwidth resources, which would cause less transmission delay (labels B and E). Delay variation is eliminated by using antijitter buffers (see Section II-B). With respect to [5], there are two different buffers for H and L traffic, respectively; they have the same size (B_{C2P} and B_{P2C}, according to the path) but different prebuffering time so that the end-to-end constant delay will be the same for both priority classes according to (1). This property simplifies the reconstruction of the original message sequence by the receiver (labels C and F).

Delay variation is higher for low-priority messages than for high-priority messages [19]. Fig. 3 shows a possible delay distribution as a function of message priority. By considering the hatched areas in Fig. 3, it is clear that buffering leads to a packet loss probability, which is higher for low-priority messages than for high-priority messages. The knowledge of the original delay distribution and of the buffer size allows to evaluate the packet loss probability to be analytically modeled in the MPC framework. Therefore, the block diagram in Fig. 5 can be conceptually replaced by the analytical model in Fig. 6 since the shaded areas of both diagrams are functionally equivalent. The data paths from the controller to plant and vice versa can be decomposed into two "virtual wires" representing the forwarding policies, i.e., H and L, characterized by different loss probabilities. The effect of packet loss on the transmitted data can be modeled by multiplying it with a binary random variable. Let $\sigma_k^H, \sigma_k^L, \mu_k^H,$ and μ_k^L be independent and identically distributed Bernoulli variables. For instance, σ_k^H and σ_k^L (μ_k^H and μ_k^L) take value 1 with probability values $\bar{\sigma}^H > \bar{\sigma}^L$ ($\bar{\mu}^H > \bar{\mu}^L$), respectively, because the H policy is more reliable than L policy. The original model (3) becomes

$$\begin{cases} x_{k+1} = Ax_k + \sigma_k^{\pi_f} Bu_k \\ y_k = \mu_k^{\pi_b} x_k \end{cases} \tag{8}$$

where $\pi_f, \pi_b \in \{H, L\}$, i.e., $\sigma_k^{\pi_f} \in \{\sigma_k^H, \sigma_k^L\}$ and $\mu_k^{\pi_b} \in \{\mu_k^H, \mu_k^L\}$. Let δ_k and ϱ_k be the binary variables representing the priority assignment for commands and measurements, respectively. For instance, $\delta_k = 1$ and $\varrho_k = 1$ mean high priority, whereas $\delta_k = 0$ and $\varrho_k = 0$ mean low priority.

To simplify the analysis and reduce computational effort, we assume that the controller has to choose the transmission policy also for the plant-to-controller path, which thus behaves in a specular manner, i.e., $\varrho_k = \delta_k$. In this way, we can ignore $\{\mu_k^H, \mu_k^L\}$ and focus only on $\{\sigma_k^H, \sigma_k^L\}$.

The state equation of the model (8) can be rewritten as

$$x_{k+1} = Ax_k + \left[(1-\delta_k)\sigma_k^L + \delta_k\sigma_k^H\right] Bu_k \tag{9}$$

where the marking strategy (represented by the binary variable δ_k), the channel behavior (represented by the random variables σ_k^L and σ_k^H) and the control command (represented by the value u_k) are explicitly indicated. In fact, with $\delta_k = 1$, the channel behavior is described by the Bernoulli variable σ_k^H, otherwise σ_k^L is considered.

The performance index $J_{\mathrm{MPC}}(k)$ defined in (4) has now to be modified for the following two reasons:

1) the performance index has to penalize the use of the high priority, otherwise the control architecture will always select the H priority, thus, wasting network resources and becoming unfair with respect to other control applications. The new index is

$$J_{\mathrm{MPC-QoS}}(k) = J_{\mathrm{MPC}}(k) + \sum_{i=0}^{N_c} \left[\|\delta_{k+i}\|_{W_i}^2\right]$$

where the weights W_i are positive;

2) the plant (9) is now a stochastic system due to the randomness introduced by the Bernoulli processes. This means that the performance index needs the expectation operator. However, since the state is known at time k, the conditional expectation given the state x_k is used as follows:

$$\ell_{\mathrm{MPC-QoS}}(k) := \mathbf{E}\left[J_{\mathrm{MPC-QoS}}(k) \,|\, x_k\right]. \tag{10}$$

Let MPC-QoS be the name of the following problem:

Problem 5: (MPC Problem Over Lossy Networks) Given the system (9), find the optimal control $u_k^\star$ and the optimal transmission strategy $\delta_k^\star$ for the corresponding packet by solving the stochastic MPC-QoS problem

$$\begin{cases} \{u_k^\star, \delta_k^\star\} = \arg\min \mathbf{E}[J_{\mathrm{MPC-QoS}}(k)|x_k] \\ \text{subject to } m_u \le \hat{u}_{k+i|k} \le M_u \\ m_x \le \hat{x}_{k+i|k} \le M_x \\ \delta_k \in \{0,1\} \\ \sigma_k^L, \sigma_k^H \text{ i.i.d. Bernoulli.} \end{cases}$$

In the following section, the conditional expectation will be computed in order to rewrite the aforementioned problem in a solvable MIQP problem.

Remark 6: In general, delay distribution of a given message depends not only on its priority but also on the effects of other traffic sources; therefore, the corresponding packet loss probability in the MPC-QoS approach also takes into account the instantaneous network condition to further optimize the use of high-priority bandwidth; for instance, even a very important command is sent in a low-priority packet if currently the network is not congested. Furthermore, the proposed MPC minimizes the use of high-priority bandwidth, thus, contributing to keep its overall utilization low. Finally, the Bernoulli model assumes statistically independent loss events, which is not the case if they are generated by buffer overflow/underflow. Nevertheless, well-known advanced queue management techniques can reduce this statistical dependence [29].

IV. Solution of MPC-QoS

This section presents a solution of the stochastic MPC-QoS problem defined previously by rewriting the minimization problem as an LQ programming problem. To reach this goal, the first step is to recognize that the system (9) belongs to the family of mixed logical dynamical (MLD) systems introduced in [30] as

$$\begin{cases} x_{k+1} = Ax_k + B_u u_k + B_\delta \delta_k + B_a a_k \\ y_k = Cx_k + D_u u_k + D_\delta \delta_k + D_a a_k \\ E \le E_x x_k + E_u u_k + E_\delta \delta_k + E_a a_k \end{cases}$$

where x_k is the state, u_k is the input, y_k is the output, δ_k is a logical variable (0-1 variable), and a_k is an auxiliary variable. The auxiliary variables usually take care of the products between decision variables and states or inputs.

By defining the auxiliary variable as $a_k = \delta_k u_k$, the state equation takes the form

$$\begin{aligned} x_{k+1} &= Ax_k + \left[(1-\delta_k)\sigma_k^L + \delta_k\sigma_k^H\right] Bu_k \\ &= Ax_k + \sigma_k^L Bu_k + \left[-\sigma_k^L + \sigma_k^H\right] Ba_k. \end{aligned}$$

The decision variable "disappears" and an auxiliary variable shows up [16]. The two equations are equivalent if and only if the following inequalities hold:

$$a_k \le M_u \delta_k \tag{11}$$

$$a_k \ge m_u \delta_k \tag{12}$$

$$a_k \le u_k - m_u(1-\delta_k) \tag{13}$$

$$a_k \ge u_k - M_u(1-\delta_k) \tag{14}$$

where $M_u = \max\{u(\cdot)\}$ and $m_u = \min\{u(\cdot)\}$ are vectors defining the range of the commands [30]. With respect to the original MLD model, the present state equation is inherently stochastic: there are products of stochastic variables with input and auxiliary variables.

The second step has to bring the constrained minimization $\min \mathbf{E}[J_{\mathrm{MPC-QoS}}(k)|x_k]$ into an LQ programming problem. The explicit computation of the conditional expectations leads to a particular mixed integer LQ problem, where the integer variables are now of the 0–1 type.

Using for $\hat{X}(k)$, the same matrix notation as before, and the following ones for the terms $\hat{u}_{k+i|k}\sigma_{k+i}^L$, $\hat{a}_{k+i|k}(-\sigma_{k+i}^L + \sigma_{k+i}^H)$, and $\delta_{k+i|k}$

$$\hat{U}_\sigma(k) := \mathsf{colvec}\left\{\hat{u}_{k+i|k}\sigma_{k+i}^L\big|_{i=0}^{N_c}\right\}$$

$$\Delta(k) := \mathsf{colvec}\left\{\delta_{k+i|k}\big|_{i=0}^{N_c}\right\}$$

$$\hat{A}_\sigma(k) := \mathsf{colvec}\left\{\hat{a}_{k+i|k}(-\sigma_{k+i}^L + \sigma_{k+i}^H)\big|_{i=0}^{N_c}\right\}$$

the matrix notation of the state equation is

$$\hat{X}(k) = \mathcal{A}x_k + \mathcal{B}\hat{U}_\sigma(k) + \mathcal{B}\hat{A}_\sigma(k) \tag{15}$$

whereas the index $J_{\mathrm{MPC-QoS}}$ becomes

$$J_{\mathrm{MPC-QoS}} = \hat{X}(k)^T \mathcal{Q}\hat{X}(k) + \hat{U}^T(k)\mathcal{Q}\hat{U}(k) + \\ + \Delta^T(k)\mathcal{W}\Delta(k) \tag{16}$$

with $\mathcal{W} = \mathrm{diag}\{W_i\}_{i=0}^{N_c}$. Since the index weights the binary variables through the matrix $\mathcal{W}$, there is no need to also weight the auxiliary vector $\hat{A}_\sigma(k)$. The constraints for the problem are (6) and $\Delta(k) \in \{0,1\}^{N_c}$. Moreover, since the inequalities (11)–(14) have to hold $\forall k$, they can be put in matrix form as

$$\mathcal{E}_u\hat{U}(k) + \mathcal{E}_a\hat{A}(k) + \mathcal{E}_\delta\Delta(k) \leq \mathcal{E} \tag{17}$$

where $\hat{A}(k)$ is equal to $\hat{A}_\sigma(k)$ without the contribution of the Bernoulli random variable, and for opportune matrices $\mathcal{E}_u, \mathcal{E}_a, \mathcal{E}_\delta$, and $\mathcal{E}$.

Since all the terms $\hat{x}_{k+i|k}, \hat{u}_{k+i|k}, \hat{a}_{k+i|k}$ in $\mathbf{E}[J_{\mathrm{MPC-QoS}}|x_k]$ are measurable, w.r.t., x_k by construction and by using well-known results about the Bernoulli random variable and the conditional expectation, [31], to minimize the stochastic index is equivalent to solve the deterministic optimization problem

$$\min_{V(k)} V^T(k)\mathcal{H}V(k) + \mathcal{P}_k V(k) + \mathcal{K}_k$$

where the unknown vector is $V(k) := \left[\hat{U}(k)\ \hat{A}(k)\ \Delta(k)\right]^T$. The matrices $\mathcal{H}$, $\mathcal{P}_k$, and $\mathcal{K}_k$ are function of $\mathcal{A}, \mathcal{B}, \mathcal{Q}, \mathcal{R}$ and $\bar{\sigma}^L, \bar{\sigma}^H$. The current state x_k is in $\mathcal{P}_k$, whereas $\mathcal{K}_k$ collects all the contributions independent of $V(k)$. The analytical expressions of these matrices can be found in [16].

Since the constraints (6) and (17) can be written as $\mathcal{C}V(k) \leq \mathcal{D}$, the MIQP related to the MPC-QoS problem is

$$\begin{aligned} V^\star(k) = \ & \arg\min_{V(k)} V^T(k)\mathcal{H}V(k) + \mathcal{P}_k V(k) \\ & \text{s. to } \mathcal{C}V(k) \leq \mathcal{D} \\ & \Delta(k) \in \{0,1\}^{N_u}. \end{aligned} \tag{18}$$

By collecting these results, we proved the following.

Theorem 7: The stochastic MPC-QoS Problem 5 is equivalent to the deterministic MIQP problem in (18).

In [30], this kind of controller is called mixed integer predictive controller. The matrix $\mathcal{H}$ is positive definite because $\mathcal{R}$ and $\mathcal{Q}$ are positive definite and thanks to the following result.

Proposition 8: Let $\bar{\sigma}^L$ and $\bar{\sigma}^H$ be real numbers such that $0 < \bar{\sigma}^L < \bar{\sigma}^H < 1$. If the weighting matrix Q is positive definite and the Assumption 3 holds, then the matrices $\mathcal{M}^{\sigma uu}$ and $\mathcal{M}^{\sigma dd}$ are positive definite.

Proof: We refer the reader to [16] for the demonstration based on the Rayleigh–Rits theorem [32]. □

When the optimal solution $V^\star(k) = [\hat{U}^\star(k) \quad \hat{A}^\star(k) \quad \Delta^\star(k)]^T$ of the problem (18) is available, the optimal control and the optimal transmission strategy at time k are

$$\begin{aligned} u^\star(k) &= \begin{bmatrix} I & 0 & \cdots & 0 \end{bmatrix} \hat{U}^\star(k) \\ \delta^\star(k) &= \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \Delta^\star(k). \end{aligned}$$

V. Simulation Results

The proposed control architecture has been validated in MATLAB/Simulink by simulating the block diagram in Fig. 6 with the control architecture of Fig. 4. The electromechanical parameters of the PMSM are: static phase resistance $R = 0.13\ \Omega$, armature inductance $L = 0.67$ mH, torque constant $k_\tau = 0.4135$ Nm/A, voltage constant $k_V = 0.477$ V/(rad/s) $= 50$ V/kRPM, number of pair couples $2p = 3$, moment of inertia $J_{\mathrm{rotor}} = 0.000175\,\mathrm{kg{\cdot}m^2}$, damping $B_{\mathrm{rotor}} = 1.66 \times 10^{-6}$ Nm/(rad/s); whereas the mechanical parameters of motor and vehicle [i.e., the constants in (2)] are $J_m = 0.13\,\mathrm{kg{\cdot}m^2}$ and $B_m = 0.013$ Nm/(rad/s). We set also the torque limit $u_k \in [-11.68, 11.68]$, which corresponds to $I_k^{q,\mathrm{ref}} \in [-30, 30]$ A.

The constants in Fig. 4 are $k_1 = 0.154$(km/h)/(rad/s) (which depends on the gear ratio, $N = 7$, and the wheel diameter, $D = 0.6$ m) and $k_2 = 1/k_\tau$.

According to the block diagram in Fig. 4, to improve steady-state performance (i.e., constant velocity), the integral of the velocity has been added to the state variable in (3). In such a way, the MPC-QoS controller $C(z)$ behaves as a PI controller in velocity where the gains are selected at run time according to the minimization problem (11). The matrix

$$Q = \begin{bmatrix} Q_p & 0 \\ 0 & Q_v \end{bmatrix}$$

is chosen diagonal for simplicity, whereas R is set to one because it is just a scale factor. The control and prediction horizons have the same value, i.e., $N_c = N_p = 8$; the output command is constrained to belong to the interval $[-11.68, 11.68]$ Nm. The sample time is $T_s = 0.01$ s.

In all simulations, we use the same reference velocity signal and disturbance (equivalent torque at the motor rotor) to fairly compare the standard deviation of the tracking error

$$\tilde{\omega}(k) = \omega^{\mathrm{ref}}(k) - \omega(k). \tag{19}$$

We will start by showing the behavior of the controlled motor with the following parameters:

$$\begin{aligned} & Q_p = 0.1, Q_v = 2 \\ & \bar{\sigma}^H = \bar{\mu}^H = 0.9, \bar{\sigma}^L = \bar{\mu}^L = 0.5 \end{aligned}$$

and varying W, which controls the use of high-priority packets in the optimization process.

1) In Fig. 7, W is so high that no packets are sent using the H priority. This case is the lower bound in terms of performance and corresponds to the stochastic MPC without DiffServ.
2) In Figs. 8 and 9, $W = 50$ and $W = 100$, respectively. When W decreases, the number of packets sent with H priority increases thus improving control performance, i.e., the standard deviation of $\tilde{\omega}(\cdot)$ decreases.

It is worth highlighting that in all cases the same MPC-QoS controller is adopted. Moreover, the policy selected at the controller side $\delta^\star$ is used also for sending the measurement at the motor side; taking into account the communication delay and the packet lost probability, we set $\varrho^\star(k) = \delta^\star(k - t_{C2P})$ when

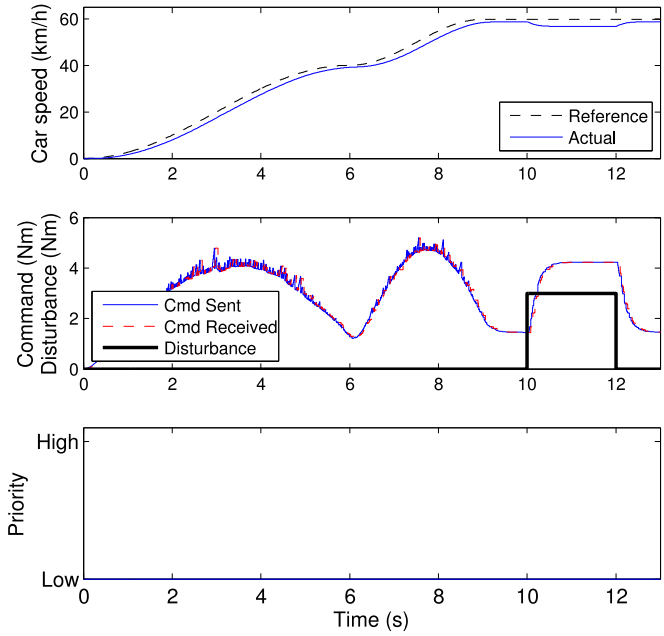


Fig. 7. $Q_p = 0.1$, $Q_v = 2$, $R = 1$, $W = 100\,000$, $\sigma_H = 0.9$, $\sigma_L = 0.5$, %H = 0, %L = 100, std. tracking error=0.8327, total pck loss tate [%] = 50.

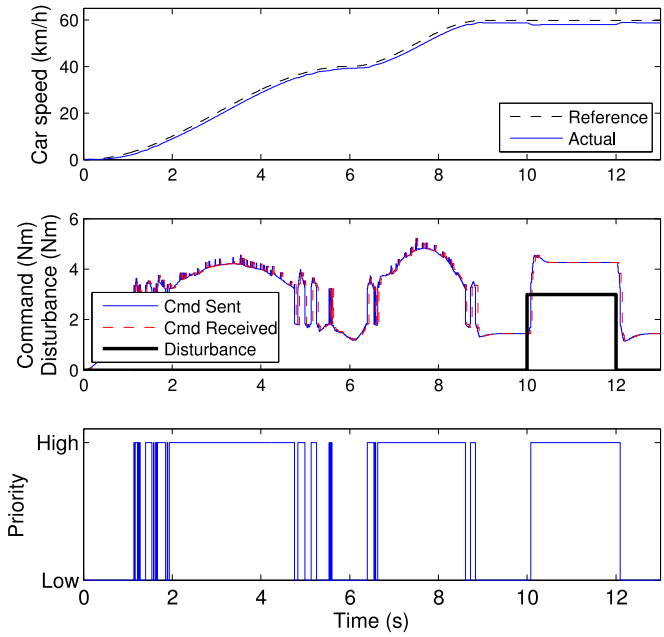


Fig. 8. $Q_p = 0.1$, $Q_v = 2$, $R = 1$, $W = 50$, $\sigma_H = 0.9$, $\sigma_L = 0.5$, %H = 60.42, %L = 39.58, std. tracking error = 0.3920, total pck loss rate [%] = 25.83.

the packet containing $\delta^\star$ arrives or $\varrho^\star(k) = \varrho^\star(k-1)$ if the packet gets lost.

The first plot in each figure shows the reference and actual car speed. The second plot shows the disturbance. The third plot shows the sent and received commands: the command, if received, is delayed by t_{C2P}, otherwise the current controller of the PMSM holds the last command (i.e., current $I_k^{q,\mathrm{ref}}$ according

Fig. 9. $Q_p = 0.1$, $Q_v = 2$, $R = 1$, $W = 100$, $\sigma_H = 0.9$, $\sigma_L = 0.5$, %H = 35.74, %L = 64.26, std. tracking error = 0.4347, total pck loss rate [%] = 35.70.

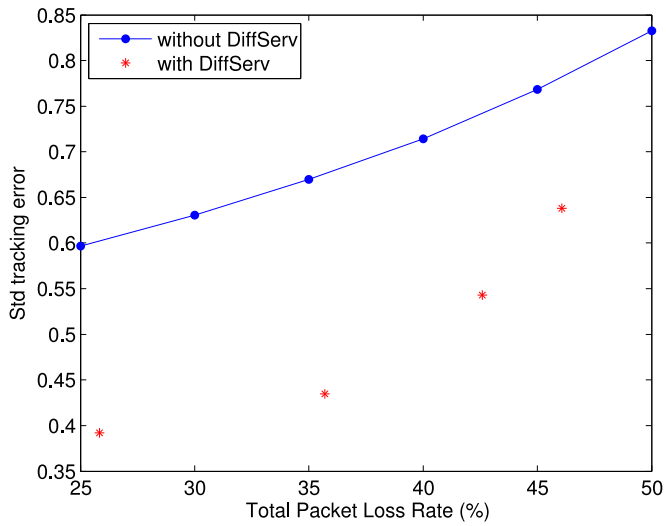


Fig. 10. Comparison of tracking error. P1: W = 200, %H = 9.84, %L = 90.16; P2: W = 150, %H = 18.52, %L = 81.48; P3: W = 100, %H = 35.74, %L = 64.26; P4: W = 50, %H = 60.42, %L = 39.58.

to the block diagram in Fig. 4). The fourth plot reports the use of low and high priority classes.

Fig. 10 compares the standard deviation of tracking error as a function of network packet loss rate with/without DiffServ (i.e., use of traffic priority in the MPC optimization). For each value of total packet loss rate corresponding to the points P1,...,P4, the DiffServ approach always reduces the tracking error with respect to unprioritized transmission. It is worth noting that for a given network load, the total packet loss rate is independent of the use of DiffServ but DiffServ leads to sacrifice the least

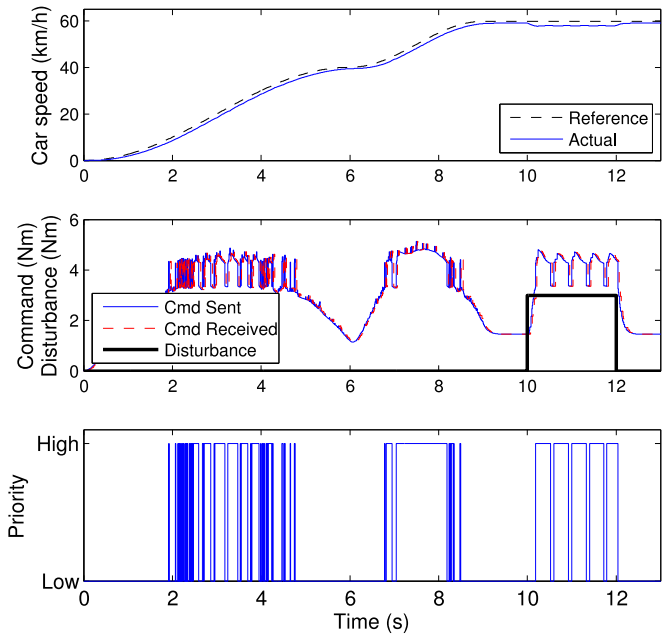


Fig. 11. $Q_p = 0.1$, $Q_v = 2$, $R = 1$, $W = 50$, $\sigma_H = 0.9$, $\sigma_L = 0.7$, $\%H = 33.90$, $\%L = 66.10$, std. tracking error $= 0.4746$, total pck loss rate [%] $= 23.22$.

important packets to preserve the most important ones thus maximizing control performance. This test also shows that DiffServ can be combined to any other pure control strategy for a further improvement.

Another simulation was performed to show what happens when the reliability provided by the low-priority class is close to the one of the high-priority class (i.e., $\bar{\sigma}^L$ is close to $\bar{\sigma}^H$). Comparing Fig. 8 ($\bar{\sigma}_L = 0.5$) and Fig. 11 (with $\bar{\sigma}_L = 0.7$), we observe that the percentage of H decreases, from 60.42% to 33.90% (∼50 percent) since the controller avoids to waste high-priority packets if they are useless (the standard deviation of the tracking error slightly increases from 0.3920 to 0.4746.). This attitude is particularly valuable because of a kind of memory effect of the network according to which if high-priority packets are economized at a given time, then their reliability is further increased in the following. Another advantage of this sensitivity to the difference between $\bar{\sigma}^L$ and $\bar{\sigma}^H$ is related to the fact that such parameters are not constant in actual scenarios but rather they depend on the instantaneous network load. This suggests a future extension of MPC-QoS in which such parameters are promptly updated from network statistics to adapt the control strategy to network condition.

VI. Conclusions

In this paper, a novel MPC for electric drives has been proposed. The original contribution is the assignment of priority to each packet based on its content, on the corresponding effect on its delivery, and on the state of the channel. All these aspects are jointly addressed by an extended MPC approach, which minimizes the tracking error and the use of high-priority packets. Therefore, not only the performance of a specific control loop are maximized, as done by all previous literature, but also the overall behavior of the control flows sharing the same network. The complete statistical behavior of messages is considered (i.e., delay distribution rather than just the worst-case response time); by using a smart buffering technique, delay variation is transformed into constant and known delay (easily addressable in control design) with the introduction of a loss probability, which is modeled in the proposed MPC framework. Simulation results show that the communication channel is used in a smarter way, i.e., priorities allow to sacrifice the least important packets to preserve the most important ones thus maximizing control performance.

References

[1] S. Williamson, A. Rathore, and F. Musavi, "Industrial electronics for electric transportation: Current state-of-the-art and future challenges," *IEEE Trans. Ind. Electron.*, vol. 62, no. 5, pp. 3021–3032, May. 2015.

[2] M. Rahmani, K. Tappayuthpijarn, B. Krebs, E. Steinbach, and R. Bogenberger, "Traffic shaping for resource-efficient in-vehicle communication," *IEEE Trans. Ind. Informat.*, vol. 5, no. 4, pp. 414–428, Nov. 2009.

[3] S. Kim, E. Lee, M. Choi, H. Jeong, and S. Seo, "Design optimization of vehicle control networks," *IEEE Trans. Veh. Technol.*, vol. 60, no. 7, pp. 3002–3016, Sep. 2011.

[4] M. Hu, J. Luo, Y. Wang, M. Lukasiewycz, and Z. Zeng, "Holistic scheduling of real-time applications in time-triggered in-vehicle networks," *IEEE Trans. Ind. Informat.*, vol. 10, no. 3, pp. 1817–1828, Aug. 2014.

[5] I. Jurado, P. Millán, D. Quevedo, and F. Rubio, "Stochastic MPC with applications to process control," *Int. J. Control*, vol. 88, no. 4, pp. 792–800, 2015.

[6] G. Goodwin, H. Haimovich, D. Quevedo, and J. Welsh, "A moving horizon approach to networked control system design," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1427–1445, Sep. 2004.

[7] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. Sastry, "Foundations of control and estimation over lossy networks," *Proc. IEEE*, vol. 95, no. 1, pp. 163–187, Jan. 2007.

[8] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry, "Kalman filtering with intermittent observations," *IEEE Trans. Autom. Control*, vol. 49, no. 9, pp. 1453–1464, Sep. 2004.

[9] C. F. Caruntu, M. Lazar, R. H. Gielen, P. van den Bosch, and S. D. Cairano, "Lyapunov based predictive control of vehicle drivetrains over CAN," *Control Eng. Practice*, vol. 21, no. 12, pp. 1884–1898, 2013.

[10] K. Nichols, V. Jacobson, and L. Zhang, "A Two-bit differentiated services architecture for the internet," Internet Requests for Comments, RFC 2638, Jul. 1999.

[11] J. De Martin and D. Quaglia, "Distortion-based packet marking for MPEG video transmission over DiffServ networks," in *Proc. IEEE Int. Conf. Multimedia Expo.*, 2001, pp. 399–402.

[12] R. Muradore, D. Quaglia, and P. Fiorini, "Adaptive LQ control over differentiated service lossy networks," in *Proc. World Congr. Int. Federation Autom. Control*, 2011, pp. 13245–13250.

[13] T. Goggia, A. Sorniotti, L. De Novellis, A. Ferrara, P. Gruber, J. Theunissen, D. Steenbeke, B. Knauder, and J. Zehetner, "Integral sliding mode for the torque-vectoring control of fully electric vehicles: Theoretical design and experimental assessment," *IEEE Trans. Veh. Technol.*, vol. 64, no. 5, pp. 1701–1715, May. 2015.

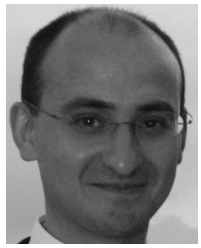
[14] Z. Shuai, H. Zhang, J. Wang, J. Li, and M. Ouyang, "Combined AFS and DYC control of four-wheel-independent-drive electric vehicles over CAN network with time-varying delays," *IEEE Trans. Veh. Technol.*, vol. 63, no. 2, pp. 591–602, Feb. 2014.

[15] Z. Shuai, H. Zhang, J. Wang, J. Li, and M. Ouyang, "Lateral motion control for four-wheel-independent-drive electric vehicles using optimal torque allocation and dynamic message priority scheduling," *Control Eng. Practice*, vol. 24, pp. 55–66, 2014.

[16] R. Muradore, D. Quaglia, and P. Fiorini, "Predictive control of networked control systems over differentiated services lossy networks," in *Proc. IEEE/ACM Conf. Design, Autom. Test Eur.*, Mar. 2012, pp. 1245–1250.

[17] S. Tuohy, M. Glavin, C. Hughes, E. Jones, M. Trivedi, and L. Kilmartin, "Intra-vehicle networks: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 534–545, Apr. 2015.

[18] International Organization for Standardization, "Road vehicles—Interchange of digital information—Controller area network (CAN) for high speed communication," International Organization for Standardization,Geneva, Switzerland, Tech. Rep. ISO 11898:1993(E), 1994.
[19] T. Herpel, K.-S. Hielscher, U. Klehmet, and R. German, "Stochastic and deterministic performance evaluation of automotive CAN communication," *Comput. Netw.*, vol. 53, no. 8, pp. 1171–1185, 2009.
[20] L. Repele, R. Muradore, D. Quaglia, and P. Fiorini, "Improving performance of networked control systems by using adaptive buffering," *IEEE Trans. Ind. Electron.*, vol. 61, no. 9, pp. 4847–4856, Sep. 2014.
[21] T. D. Gillespie, *Fundamentals of Vehicle Dynamics*. Warrendale, PA, USA: SAE International, 1992.
[22] J. Maciejowski, *Predictive Control: With Constraints*. London, U.K.: Pearson, 2002.
[23] A. Bemporad, M. Morari, V. Dua, and E. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002.
[24] A. Bemporad, "Predictive controller with artificial Lyapunov function for linear systems with input/state constraints," *Automatica*, vol. 34, no. 10, pp. 1255–1260, 1998.
[25] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, pp. 789–814, 2000.
[26] J. Rawlings and K. Muske, "The stability of constrained receding horizon control," *IEEE Trans. Autom. Control*, vol. 38, no. 10, pp. 1512–1516, Oct. 1993.
[27] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems. 2014*. [Online]. Available: http://www.mpc.berkeley.edu/mpc-course-material
[28] K. Zuberi and K. Shin, "Scheduling messages on controller area network for real-time CIM applications," *IEEE Trans. Robot. Autom.*, vol. 13, no. 2, pp. 310–316, Apr. 1997.
[29] S. Floyd and V. Jacobson, "Random early detection gateways for congestion avoidance," *IEEE/ACM Trans. Netw.*, vol. 1, no. 4, pp. 397–413, Aug. 1993.
[30] A. Bemporad and M. Morari, "Control of systems integrating logic, dynamics, and constraints," *Automatica*, vol. 35, pp. 407–428, 1999.
[31] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*. New York, NY, USA: McGraw-Hill, 1991.
[32] R. Horn and C. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 2005.



Davide Quaglia (S'00–M'03) received the Ph.D. degree in computer engineering from the Politecnico di Torino, Turin, Italy, in 2003.

He is currently an Assistant Professor in the Computer Science Department, University of Verona, Verona, Italy, where he currently teaches the courses "HW Architectures for Bioinformatics" and "Design of Networked Embedded Systems.'. He is also a cofounder and an active Project Leader of EDALab s.r.l., a spin-off company of the University of Verona. He is the author/coauthor of about 50 papers. His current research interests include networked embedded systems, networked control systems, and cyber-physical systems.

Dr. Quaglia is the Chair of the Special Session on Cyber-Physical Systems of the Euromicro DSD Conference and a Member of Euromicro DSD as well as the ECSI FDL Program Committees.

Riccardo Muradore (S'99–M'04) received the Laurea degree in information engineering in 1999 and the Ph.D. degree in electronic and information engineering in 2003 from the University of Padova, Padova, Italy.

He was a Postdoctoral Fellow with the Department of Chemical Engineering, University of Padova, from 2003 to 2005. He then spent three years at the European Southern Observatory, Munich, Germany, as a Control Engineer, working on adaptive optics systems. In 2008, he joined the ALTAIR Robotics Laboratory, University of Verona, Verona, Italy, where, since 2013, he has been an Assistant Professor. His research interests include robust control, robotics, teleoperation, networked control systems, and adaptive optics.