Bulk data transfer in Controller Area Networks for Industry 4.0

Giovanni Zanuso

*Division of Electric Power and Energy Systems
KTH Royal Institute of Technology
Stockholm, Sweden
zanuso@kth.se*

Viktoria Fodor

*Division of Network and Systems Engineering
KTH Royal Institute of Technology
Stockholm, Sweden
vfodor@kth.se*

Abstract—Industrial Controller Area Networks (CANs) are traditionally responsible for transmitting status and control information from and to local controllers. However, with the advances of electronics, end-nodes are capable of performing more complex operations, such as monitoring the state of the device and optimizing its operation. As a consequence, CANs should be used for the transmission of heterogeneous message types. In this work we evaluate how CANs can share their transmission resources and collect also bulk data, in addition to serving the existing controller traffic. We derive the constraints bulk data transmission needs to fulfill to ensure the robust and timely transmission of controller traffic, and propose scheduling solutions that ensure fast data delivery. Our results demonstrate that bulk data transmission is possible, unless the networks is already highly loaded, and fast delivery can be achieved even with simple transceiver hardware.

Index Terms—Industrial networks, controller area networks, heterogeneous traffic, bulk data transfer

I. INTRODUCTION

The Industry 4.0 and Industrial Internet visions [1], [2] build on the developments in the areas of electronics, communication and computing. With the help of increased connectivity between the devices of an industrial plant, the aim is to achieve higher level of automation, and with this higher efficiency and flexibility in the industrial processes.

Devices today are connected to central controllers on the so called plant and direct process control levels, and send and receive highly time sensitive state information and control commands respectively. These end devices today are governed by processing units, such as microprocessors and microcontrollers. The processing units have increasing storage and computing capabilities, and consequently, in addition to receiving control information and actuating accordingly, they could perform more complex measurements for status monitoring, and could also participate on the optimization of the plant performances on longer timescales. Monitoring and process optimization would however require the collection of additional larger amount of data from the devices.

The main question we ask in this work is whether this data transmission would be possible in the existing communication

infrastructures, which were designed with the very specific process control application in mind. For this purpose, we consider the traditional CAN buses, which are widely used both in the process and in the automotive industry [3].

We utilize the mathematical models available to derive response times when the CAN bus is used to collect small, periodically generated messages [4]. We extend these models to derive the worst case delays of bulk data transmissions, considering two different transceiver hardware architectures.

The contributions of the paper are as follows:

- Considering CAN nodes with a limited buffer space, we provide a model for schedulability analysis and an algorithmic solution to schedule bulk data transmissions with minimized transmission delays.
- Considering buffered bulk data, we provide the analytic model to calculate the transmission delay.
- We perform a numerical evaluation, and show that with careful scheduling bulk data can be transmitted with low delay even when the buffer space is limited.

The rest of the paper is organized as follows. In Section II we summarize the related literature. In Section III the basics of CANs are introduced and in Section IV we present the system model. Section V contains the main contribution of the paper with the response time and schedulability analysis. A numerical evaluation of the different scheduling solutions is then presented in Section VI. Section VII concludes the work.

II. RELATED WORK

CANs were first analyzed extensively in the nineties, often focusing on the automotive industry. This period is characterized by a sequence of publications from Tindell and his co-authors [5]–[7], proposing an analytic model of the worst case message latencies and related schedulability analysis for CANs transmitting periodic messages with hard real-time constraints. These first models were proved to be faulty by Davis et al. in [4], where they proposed modifications. These models are used today, with contributions towards considering the effects of faults [8], providing algorithms for the dynamic extension of the address space [9], and proposing schedulability analysis in the presence of FIFO transceivers [10], or with priority queues [11].

This work has been supported in part by Vinnova "Smartare elektroniksystem" strategic innovation program and the Swedish Electromobility Center (SEC).

In this work we address the capabilities of CAN buses when the transmission resources are shared by the real-time, periodic control messages, and aperiodic bulk data transfer without transmission deadline limitations. Our problem formulation is motivated by condition monitoring of electric drives, which are widely employed in both process industry (e.g. for pumps and fans) and in automotive (e.g. electric vehicles). Classical condition monitoring methods are based on motor current signature analysis [12], while recently various diagnostic solutions based on high-frequency current sampling (or oversampling) have been proposed [13], [14]. The effectiveness of condition monitoring methods depends on a comparison between current measurements and an initial reference measurement performed when the electric drive was supposedly healthy [12]. Moreover, diagnostics methods may need to take into account the whole history of measurements. Therefore, large amounts of data need to be acquired and processed, which in the first place requires the collection of the measurement data from the electric drives connected to the CAN bus.

III. CONTROLLER AREA NETWORKS

The CAN is a serial data bus that implements a virtual priority queue for medium access [3]. The key of the medium access control is priority based arbitration. The CAN physical layer supports two states termed dominant ('0') and recessive ('1'). If two or more CAN controllers are transmitting at the same time and at least one of them transmits a '0', then the value on the bus will be a '0'. This feature is used to implement priority based access. The priority of a message is represented by its address, and addresses are assigned in a way that if two or more addresses are transmitted on the bus at the same time, the address of highest priority does not change, while in the lower priority addresses some 1-s will turn to 0 due to interference. The transmitters sending lower priority message notice the change by sensing the communication bus, and stop transmission. Since nodes are allowed to start transmission only if they sense the medium idle, the access control realizes non-preemptive priority access. In other words, a low priority message that is already under transmission will not be interrupted when a high priority message appears at one of the controllers.

Priority based arbitration ensures very robust access control. The downside is that all nodes are required to see the same bits at the same time, which limits the length of the bus or the transmission speed. Accordingly, the maximum data rate of a CAN bus varies from 1 Mbit/s for a network up to 40 m in length or to 125 kbit/s for a 500 m long network.

Traditionally, CANs transmit periodic control messages with hard delivery deadlines, and the main question of CAN dimensioning is the schedulability of the messages. This usually means the fulfillment of two conditions: whether the message is transmitted within the delivery deadline, and whether the message is transmitted before a new message is generated. This second condition comes from hardware constraints, as CAN nodes usually have small buffer capacity, but it also

Fig. 1. A CAN bus with nodes managing multiple transmission buffers for transmitting control messages and a dedicated buffer for bulk data transfer.

reflects that, in typical control applications, a given type of information or control message becomes outdated when a new one is generated.

IV. SYSTEM MODEL

A typical CAN is shown on Figure 1. The network contains the serial bus and the connected controllers (nodes), where each controller may contain multiple message buffers. CAN solutions may differ in the details of the realizations, considering the memory space and the service order among the local buffers [10]. In this work we consider that each node contains at least one buffer for periodic control messages and another one for eventual bulk data transfer. We assume that each node respect the priorities of its queued messages, and consequently, at any given time when arbitration starts, the highest priority message queued in the network is served.

We consider a CAN transmitting periodic control messages of N priority classes. Note that it may reflect a network with N nodes, or a smaller network with several buffers at each node. The bus has a transmission speed of v . Given a generic control message with a priority m and size L_m , its transmission time on the bus is $C_m = \frac{L_m}{v}$. The priority m messages are transmitted periodically with a time period of T_m and have a hard delivery deadline $D_m \leq T_m$.

The worst-case response time R_m of a message is defined as the longest possible time from the initiating event to the message being received by the nodes that require it. A control message is said to be schedulable if and only if $R_m \leq D_m$.

In addition, we consider that occasionally nodes transmit a bulk data of size L_b , for condition monitoring or diagnostic purposes, which differently than control messages do not have strict deadlines. Still, we are interested in the delivery time of the bulk data, depending on the service policy and the network parameters. Because it exceeds the maximum payload of a single CAN packet, the bulk data is transmitted as a sequence of N_b packets, each with transmission time C_b .

Specifically, we consider two approaches to transmit the bulk data:

- Periodic transmission: In the case of a simple hardware, the state of the transmission buffer is not known by the higher protocol layers at the node. The packets of the bulk data are placed in the transmission buffer periodically,

TABLE I
NOTATION USED IN THE ANALYTIC MODEL

N	Number of priority classes
m	Priority class $m = 1 \dots N$
v	Transmission rate
L_m	Control message size
C_m	Control message transmission time
T_m	Control message transmission interval
D_m	Control message deadline
R_m	Control message worst-case response time
L_b	Bulk data size
N_b	Number of bulk data packets
C_b	Bulk data packet transmission time
T_b	Bulk data transmission interval
p_b	Bulk data priority level
E_b	Bulk data worst-case transmission time
B_m	Time when channel blocked by low priority message
J_m	Jitter between packet generation and placement in the buffer
τ_{bit}	Transmission time of a bit
t_m	Maximum length busy period on priority level m
Q_m	Number of priority m messages in a busy period
$w_m(q)$	Worst-case waiting time of message q of priority m
$R_m(q)$	Worst-case response time of message q of priority m
ρ	CAN control message network load

with a transmission period T_b and are transmitted with priority p_b . The values of T_b and p_b have to be selected such that all control traffic is still schedulable, and the bulk data packet is transmitted from the buffer before a new packet is placed there, that is, $R_b < T_b$.

- Buffered transmission: With a more intelligent system, a new packet can be placed in the transmission buffer as soon as it is not full. Since on a higher level it looks like the entire data is stored at the transmitter, we call this case as buffered data transmission. Note, that in this case all the capacity of the bus can be utilized when bulk data is available, even when the bulk data is transmitted with lowest priority $p_b = N + 1$.

V. RESPONSE TIME AND SCHEDULABILITY ANALYSIS

A. Periodic Control Messages

In this section we briefly summarize the response time model presented in [4]. The notation used here and in the rest of the paper is summarized in Table I.

The idea of the modeling comes from early works on priority scheduling [15]. The model derives R_m , the worst case, or maximum, response time of messages with priority m in four steps.

Step 1 determines t_m , the maximum length of a busy period seen by a priority m queue. The busy period is a continuous time interval, where the medium is occupied with messages of priority m or higher. The worst case busy period emerges, when a message m and all higher priority messages are placed in the transmission buffers at the same time, while a lower priority message just started to be transmitted and occupies the bus. Then, t_m becomes

$$t_m = B_m + \sum_{\forall k \in hp(m)} \left\lceil \frac{t_m + J_k + \tau_{bit}}{T_k} \right\rceil C_k, \quad (1)$$

where $hp(m)$ is the set of messages with priority higher than or equal to m .

The model parameters are $B_m = \max\{C_k\}, \forall k \in lp(m)$, which is the maximum time the channel can be blocked by a message of priority lower than m , J_m , an initial delay (called jitter) between the generation of a packet and its placement in the transmission buffer, and τ_{bit} , the transmission time of a bit.¹

In Step 2, Q_m , the number of instances of priority m messages transmitted within a busy period, is obtained as

$$Q_m = \left\lceil \frac{t_m + J_m}{T_m} \right\rceil, \quad (2)$$

and in Step 3, $w_m(q)$, the maximum time from the beginning of a busy period until the transmission of the q -th instance of the message m within the busy period, with $q = 1 \dots Q_m$, is derived as

$$w_m(q) = B_m + (q-1)C_m + \sum_{\forall k \in hp(m)} \left\lceil \frac{w_m(q) + J_k + \tau_{bit}}{T_k} \right\rceil C_k, \quad (3)$$

where $hp(m)$ is the set of messages with priority higher than m .

Finally, in Step 4, $R_m(q)$, the maximum response time of the q -th instance of the message m and the overall maximum R_m are calculated as

$$R_m(q) = w_m(q) + C_m + J_m - (q-1)T_m, \quad (4)$$

and

$$R_m = \max\{R_m(q)\}, \text{ with } q = 1, \dots, Q_m. \quad (5)$$

These equations can be derived by starting from the highest priority messages and considering that each low priority message needs to wait until the delivery of all higher priority messages that were generated during this waiting time. The detailed derivation is given in e.g., [4]- [7].

Note that the expressions for t_m and w_m are not explicit, however they can be calculated iteratively, since the right hand side is a non-decreasing function of these variables.

Some observations will be useful for the further discussions. First, message transmissions are affected by lower priority messages only through the blocking time B_m , but not through the transmission period of lower priority messages, or through the transmission period of messages with priority m .

Second, a busy period of priority m is always ended by the transmission of a message of priority m . As a consequence, if there is only one instance of message m transmitted in a busy period, then the response time $R_m(1)$ is equal to the busy period length t_m plus the eventual jitter J_m .

Once the response time R_m is calculated for each priority level, the schedulability can be evaluated by checking whether $R_m \leq D_m, \forall m$.

¹Note that in contrast to [4], the expression (1) includes τ_{bit} . According to our numerical experiments, this is needed, similarly to e.g. (3).

B. Bulk Data Transmission

Bulk data is transmitted intermixed with the periodic control messages. We consider the case when only one of the nodes has bulk data to transmit. This case is relevant, since bulk data often contains monitoring traffic generated with very low frequency.

1) *Periodic Transmission of Bulk Data:* First we consider the case when packets of the bulk data are placed in the transmission buffer of the node periodically, with a period T_b large enough to ensure that the previous packet is already transmitted, that is $T_b \geq R_b$, where R_b is the maximum response time of a single bulk data packet, according to (5). For fast bulk data transmissions, a small T_b value is beneficial. Clearly, the priority selected to transmit the bulk data affects which T_b values lead to schedulable control and bulk data traffic.

If the bulk data traffic is transmitted with low priority (e.g., $p_b = N + 1$), large T_b values may be enforced, since at least one instance of each higher priority control message needs to be accommodated in this period. This may lead to low network utilization and unnecessarily high T_b values when there are many priority classes and each transmits with low frequency.

Conversely, utilizing a high priority for the bulk data, T_b is constrained by the schedulability requirement of the delay sensitive control messages.

Considering these constraints, we propose to find the optimal priority level p_b^* and the corresponding minimum transmission period T_b^* numerically. This requires us to define an appropriate search interval for T_b^* .

Theorem 1: Consider the case where the bulk data is transmitted with the lowest priority, that is $p_b = N + 1$, and the control messages, having priorities $m = 1, 2, \dots, N$, are schedulable. Then, the minimum transmission period T_b^* can be bounded by T_b^L and T_b^H as follows

$$T_b^* \geq T_b^L = w_{N+1}(1) + C_b + J_b, \quad (6)$$

where $w_{N+1}(1)$ is given by (3), and

$$T_b^* \leq T_b^H = R_{N+1}^L, \quad (7)$$

with R_{N+1}^L is given by (5), assuming a transmission period $T_b = T_b^L$.

Proof: Due to (4) and (5), (6) has to hold. Since the waiting time of the first bulk data packet transmitted in a busy period depends only on the known transmission period of the control messages, $w_{N+1}(1)$ can be calculated by (3), and thus T_b^L can be calculated as well.

To derive (7) and T_b^H , we can first conclude that $R_{N+1}^L \geq T_b^L$ according to (4) and (5). Then, we need to prove that a transmission period $T_b = R_{N+1}^L$ surely ensures that the bulk data traffic is schedulable, that is, $R_b \leq T_b$.

First, a general observation is needed. Let us compare $R_m(q)$ for two scenarios. In both scenarios the messages are transmitted as message q within a busy period, but with different T_m values, whereas all the other network parameters are the same. In the two scenarios there will be different

Q_m values due to (2), and thus (5) will be obtained over a different interval of q values. Nevertheless, since the waiting time does not depend on T_m , $w_m(q)$ will be the same for the two scenarios. Then, according to (4), the message with the longer T_m will have a lower response time $R_m(q)$.

Consequently, a bulk data transmission with transmission period $T_b = R_{N+1}^L > T_b^L$ will experience a response time $R_b < R_{N+1}^L$, which ensures $R_b \leq T_b$, the schedulability of the bulk data transmission. ■

Corollary 1: Since the response time R_b is decreasing under increasing T_b value, the shortest possible transmission period such that $R_b \leq T_b$ can be found by binary search in an appropriate $[T_b^L, T_b^H]$ interval.

Based on Theorem 1 and Corollary 1, we suggest Algorithm 1 to search for the optimal priority level p_b^* and the resulting shortest possible transmission period T_b^* , so that not only the packets of the bulk data, but also the ones of the control messages having priorities $m = 1, 2, \dots, p_b^* - 1, p_b^* + 1, \dots, N + 1$, are schedulable.

First, the priority p_b of the bulk data messages is set to the lowest value $N + 1$, and the schedulability of all single control message is evaluated, performing the four steps described in Section V-A for $m = 1, 2, \dots, N$. In this way it is possible to check whether the introduction of the bulk data transmission with lowest priority makes the control messages unschedulable, regardless of the not yet calculated period T_b . Specifically, without the bulk data transfer, the lowest priority control traffic does not experience blocked channel due to lower priority transmission, that is $B_N = 0$. This is changed when the bulk data transfer is introduced, giving $B_N = C_b$, which in turn can force the control packets to arrive late. For higher priority messages the same can happen if $C_b > C_m$.

Then, the algorithm performs a binary search according to Theorem 1 and Corollary 1. It is worth noting that each binary search step is conditioned by the schedulability of both the control and the bulk data messages.

After that, the binary search is repeated with the bulk data transmission set to increasing priorities. At each priority level, the interval for the binary search is found by means of the worst-case response time of the first bulk data packet in a busy period, as the lower limit T_b^L , and the previously found shortest transmission period as upper limit T_b^H . The algorithm stops if $p_b = 1$.

Once p_b^* and T_b^* are selected, the worst-case transmission time E_b of the entire bulk message containing N_b packets can be calculated by considering that $N_{bp} = \lceil N_b / Q_b \rceil$ consecutive busy periods are present during the transmission, where Q_b is obtained with (1) and (2). Specifically, a number Q_b of packets are transmitted within each of the first $N_{bp} - 1$ complete busy periods while $N_r = N_b - (N_{bp} - 1)Q_b$ packets are transmitted in the last busy period. Thus,

$$E_b = (N_{bp} - 1) Q_b T_b + w_b(N_r) + C_b + J_b, \quad (8)$$

where the first term gives the generation time of the first bulk data packet in the last busy period, which is the same

Algorithm 1 Periodic transmission of bulk data. Outputs: bulk data priority p_b^* and transmission period T_b^* .

```

1:  $p_b = N + 1$ ;
2: Recalculate  $B_m$  for  $m = 1 \dots N + 1$ ;
3: Evaluate schedulability for  $m = 1 \dots N$ ;
4: If not schedulable, then goto End;
5: Calculate  $T_b^L, T_b^H$ , based on Theorem 1;
6: Find schedulable  $T_{b,N+1}$  via binary search in  $[T_b^L, T_b^H]$ ;
7: Set  $p_b^* = N + 1, T_b^* = T_{b,N+1}$ ;
8: For  $\{ p_b = N, p_b > 0, p_b-- \}$  do
9:   Change the priority of message  $m = p_b$  to  $p_b + 1$ ;
10:  Find schedulable  $T_{b,p_b}$  via binary search in
     $[w_{p_b}(1) + C_b + J_b, T_b^*]$ ;
11:  If  $T_{b,p_b}$  found, then update  $p_b^* = p_b, T_b^* = T_{b,p_b}$ ;
12: end For
13: End

```

as the start time of the last busy period. The term $w_b(N_r)$ is the time from the start of the last busy period until the start of the transmission of the last bulk data packet. Finally, C_b is the transmission time of this last packet and J_b is the eventual jitter.

2) *Buffered Transmission with Low Priority*: Buffered transmission allows a continuous stream of bulk data packets when the bus is idle, and a shortest possible transmission delay of the bulk data. Since priority in this case does not play any role, $p_b = N + 1$ is appropriate. The schedulability of higher priority traffic is still affected through the value of C_b as in the case of periodic transmission, and therefore it needs to be evaluated.

Since all idle time is filled in by the bulk data packets, the entire bulk data transmission forms one busy period. The length of the busy period can be calculated iteratively through the implicit form

$$t_b = B_b + \sum_{\forall k \in hp(b)} \left\lceil \frac{t_b + J_k + \tau_{bit}}{T_k} \right\rceil C_k + N_b C_b. \quad (9)$$

The form of (9) is similar to that of (1), with the difference that the bulk traffic itself occupies a time interval of $N_b C_b$.

The length of the busy period gives directly the worst-case delay E_b of the bulk data transfer, that is

$$E_b = t_b + J_b. \quad (10)$$

VI. NUMERICAL RESULTS

As an example, we consider a CAN bus with N nodes. Node m , with $m = 1 \dots N$, transmits control messages with priority m . The message size C_m and a transmission interval T_m is the same for all nodes. In addition, one of the nodes transmit bulk data.

For the numerical example we consider the use case of electric drive condition monitoring [14], with $N = 6$ or 10 nodes, $N_b = 20$ packets of bulk data to transmit, and an equal transmission time for both the control messages and the bulk data, $C_m = C_b = 0.2$ ms.

Fig. 2. Bulk data worst case transmission time under increasing control traffic load, for the lowest priority (LP), optimal priority (OP) and buffered transmission (BUF) cases.

Fig. 3. Bulk data worst case transmission time under increasing number of nodes transmitting control messages, but under constant load, for the lowest priority (LP), optimal priority (OP) and buffered transmission (BUF) cases ($\rho = 0.7$).

The transmission interval of the control messages is the same for all the nodes and is tuned such that a given load ρ is achieved on the CAN bus. Specifically, since each node transmits one message in a transmission interval,

$$\rho = \frac{NC_m}{T_m}. \quad (11)$$

A conservative delivery deadline is set as $D_m = T_m - 2C_m$.

Figure 2 compares E_b , the maximum time required to transmit the entire bulk data, under three transmission methods: i) periodic transmission of bulk data packets, where the bulk data is transmitted with the lowest priority $p_b = N + 1$, ii) periodic transmission of bulk data packets, but using the optimal priority level p_b^* and the corresponding transmission interval T_b^* , calculated according to Algorithm 1, and finally, iii) buffered bulk data transmission, using $p_b = N + 1$.

We evaluate E_b under increasing control traffic load. As we see, in the considered interval, the load does not affect the delay E_b (and thus the transmission interval T_b) if the bulk data is transmitted with lowest priority (LP). Instead, it is determined by N , the number of transmitting nodes. More nodes mean eventually larger bursts of control traffic, and thus necessarily longer transmission interval for the bulk traffic.

On the other end, the delay of buffered transmission (BUF) is determined by the load, and does not depend on the transmission pattern itself, which is affected by N . Since it utilizes nearly all available capacity, buffered transmission ensures the best result among the three considered solutions. If buffered transmission is not possible, transmitting bulk data with the optimal priority p_b^* gives good performance, when the network load is low. For the specific example, for load levels of up to $\rho = 0.4$ the bulk data was transmitted with highest priority $p_b^* = 1$. The requirement of deterministic transmission intervals still has some negative effect at very low load - since the back to back transmission of bulk data messages is still not possible. It also leads high E_b at high load, where the control traffic becomes very sensitive to additional delay, and thus the bulk data can not use high priority and low transmission intervals.

Figure 3 shows the bulk data transmission time as the number of nodes (and priority levels) N in the network is increased. The load is kept constant, which according to (11) implies that T_m increases linearly with N . Specifically, we consider a control message load of $\rho = 0.7$.

We see that buffered transmission (BUF) keeps more or less the same delay, since the time not occupied by the transmission of control messages is used for bulk data transmission. We have also expected that the delay increases linearly when bulk data is transmitted periodically, and with fixed low priority $p_b = N + 1$ (LP), since in this case the bulk data transmission interval needs to accommodate increasing number of higher priority messages.

Finally, we see that optimizing the priority level of bulk data transmission (OP) can lead to significant gains. In this specific example, the optimal priority level is strongly affected by our choice of delivery deadline. Since we apply $D_m = T_m - 2C_m$, the delivery deadline is very strict if T_m is small, which happens at small N values. In turn this means that the bulk data traffic has to use low priority levels - and thus E_b changes together with that of the low priority transmission. Instead, at high number of nodes, D_m becomes comparable to T_m , and bulk data has a chance to use high priority levels ($p_b^* = 4$ in this case), without hurting the control traffic, reaching a performance close to the one of buffered transmissions.

Although these results are far from general, they show the efficiency of optimizing the priority level of the bulk data.

VII. CONCLUSIONS

In this paper we investigated how traditional CAN can accommodate bulk data transfer in addition to the usual transfer of periodic control messages. We provided an analytic

model of control message and bulk data transmission delays, which in turn allowed us to design an algorithm for optimizing the priority level of bulk data transmission.

We showed that the optimized selection of priority level and transmission period allows low transmission times, even when it is not possible to buffer the data. Buffered bulk data transmission in turn has the largest gain when the control traffic load is high, and bulk data needs to utilize irregular free slots, or when the response time limits of the control messages are very strict.

REFERENCES

- [1] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the internet of things and Industry 4.0," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 17–27, Mar. 2017.
- [2] A. W. Colombo, S. Karnouskos, O. Kaynak, Y. Shi, and S. Yin, "Industrial cyberphysical systems: A backbone of the fourth industrial revolution," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 6–16, Mar. 2017.
- [3] ISO, *Road vehicles – Controller area network (CAN) – Part 1: Data link layer and physical signalling*, ISO ISO 11898-1:2015, 2015.
- [4] R. I. Davis, A. Burns, R. J. Bril, and J. J. Lukkien, "Controller Area Network (CAN) schedulability analysis: Refuted, revisited and revised," *Real-Time Systems*, vol. 35, no. 3, pp. 239–272, Jan. 2007.
- [5] Tindell, Hansson, and Wellings, "Analysing real-time communications: controller area network (CAN)," in *1994 Proceedings Real-Time Systems Symposium*, Dec. 1994.
- [6] K. Tindell and A. Burns, "Guaranteeing Message Latencies on Control Area Network (CAN)," 1994.
- [7] K. Tindell, A. Burns, and A. J. Wellings, "Calculating controller area network (CAN) message response times," *Control Engineering Practice*, vol. 3, no. 8, pp. 1163–1169, 1995.
- [8] I. Broster, A. Burns, and G. Rodriguez-Navas, "Probabilistic analysis of CAN with faults," in *23rd IEEE Real-Time Systems Symposium*, 2002.
- [9] K. W. Schmidt, "Robust Priority Assignments for Extending Existing Controller Area Network Applications," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 578–585, Feb 2014.
- [10] R. I. Davis, S. Kollmann, V. Pollex, and F. Slomka, "Schedulability analysis for Controller Area Network (CAN) with FIFO queues, priority queues and gateways," *Real-Time Systems*, vol. 49, no. 1, pp. 73–116, Jan. 2013.
- [11] S. Mubeen, J. Mäki-Turja, and M. Sjödin, "Extending worst case response-time analysis for mixed messages in Controller Area Network with priority and FIFO queues," *IEEE Access*, vol. 2, pp. 365–380, 2014.
- [12] M. Riera-Guaspa, J. A. Antonino-Daviu, and G. Capolino, "Advances in electrical machine, power electronic, and drive condition monitoring and fault detection: State of the art," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 3, pp. 1746–1759, Mar. 2015.
- [13] P. Nussbaumer, M. A. Vogelsberger, and T. M. Wolbank, "Induction machine insulation health state monitoring based on online switching transient exploitation," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 3, pp. 1835–1845, March 2015.
- [14] G. Zanuso, V. Fodor, L. Peretti, and O. Wallmark, "Networked electric drives in the Industry 4.0," in *International Conference on Electrical Machines and Systems (ICEMS)*, Oct. 2018.
- [15] C. L. Liu and J. W. Layland, "Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment," *Journal of the ACM*, vol. 20, no. 1, pp. 46–61, Jan. 1973.