Wang, Shengquan; Farjam, Tahmoores; Charalambous, Themistoklis

A Priority-Based Distributed Channel Access Mechanism for Control over CAN-like Networks

Published in:
2021 European Control Conference, ECC 2021

DOI:
10.23919/ECC54610.2021.9655068

Published: 01/01/2022

Document Version
Peer-reviewed accepted author manuscript, also known as Final accepted manuscript or Post-print

Please cite the original version:
Wang, S., Farjam, T., & Charalambous, T. (2022). A Priority-Based Distributed Channel Access Mechanism for Control over CAN-like Networks. In *2021 European Control Conference, ECC 2021* (pp. 176-182). TU Delft Open. https://doi.org/10.23919/ECC54610.2021.9655068

A Priority-Based Distributed Channel Access Mechanism for Control over CAN-like Networks

Shengquan Wang, Tahmoores Farjam, and Themistoklis Charalambous

***Abstract*— In this work, we study the distributed channel access problem for multiple subsystems equipped with smart sensors sharing a capacity-limited communication network to perform their control tasks. We propose a fully distributed scheme to use the limited communication resources efficiently with respect to a quadratic cost which reflects the control performance. More specifically, we adopt a priority assignment scheme which consists of two layers: *(i)* a dynamic priority that corresponds to the time-varying criticality of transmission for each subsystem, and *(ii)* a pre-specified static priority which ensures that channel access is collision-free. We first demonstrate how the dynamic priorities can be manipulated to allocate the resources with respect to the chosen cost. Next, we propose a synchronization method which enables fully distribute implementation over controller area network (CAN) hardware. We validate the compatibility of our method with the mature hardware technology of CAN by hardware-in-the-loop simulations. Finally, we demonstrate the efficacy of our proposed scheme and evaluate its performance in large-scale networks via simulation.**

***Index Terms*— Networked control systems, distributed channel access, cost of information loss, controller area network.**

I. Introduction

For more than a decade, there has been a growing trend toward adopting the concept of networked control systems (NCSs) in several applications (see, for example, [1]–[3] and references therein), where the spatially distributed sensors, controllers and actuators exchange information over a shared network, thus encompassing the tight coupling between communication and control. Although adopting NCSs instead of conventional solutions offers many benefits, such as flexible architectures and lower cost, orchestrating the communication of different components over the shared network can be challenging and even prohibitive in some scenarios.

Ample research has been dedicated to the development of *control-aware* scheduling mechanisms which often require a central unit in the network to collect data from all the involved subsystems and coordinate channel access; see, for example, [4]–[7] and references therein. However, a centralized approach results in large communication overhead and in many practical scenarios such an approach is infeasible due to the (possibly varying) network size and topology, or even problem constraints. Such a scenario necessitates implementation of distributed channel access mechanisms. A popular method is to adopt static schedules, such as round robin, where subsystems transmit according to a predefined sequence thus eliminating the need for information exchange between subsystems. Despite their simplicity and easy implementation, static schedules are inadequate for NCSs since they disregard the dynamics of the involved subsystems which determines how critical their need for communication is over time. To address this, contention-based dynamic scheduling methods have been proposed which allocate the resources based on the time-varying transmission priorities.

Try-once-discard (TOD) is one of the most well-known channel access schemes proposed for deterministic systems where the deviation of the state from its nominal value is chosen as the priority measure for sensors' data transmission [8]. The application of TOD has been also extended to networks with packet drops [9]–[11]. In [12], this idea was considered for fully observable stochastic systems and channel access was granted by combining the error-based deterministic priorities with probabilistic transmission decisions to decrease the collision rate. In case of partial observations, the information available at the sensor was utilized to derive the value of information (VoI) and use it as the priority measure in [13], [14]. In similar settings, using the information available at the estimator was shown to lead a priority metric, referred to by cost of information loss (CoIL), which relies only on the statistical properties of the observations [15]. By utilizing the concept of CoIL and relying on the fact that no measurements are needed for updating the evolution of the state estimate and the statistics on the receiver side, a timer-based mechanism was proposed for perfect communication channels [16] and for imperfect (wireless) communication channels [17]–[19]. In [16], the timers at each slot are set such that they are inversely proportional to the CoIL, while in [17]–[19] the channel conditions are also incorporated in the timers.

By assuming infinite accuracy and negligible duration of flag packets, given that the sensors transmit raw measurements, timers can provide collision-free channel access even in homogeneous networks. However, most sensors nowadays have limited processing power [20] which can be leveraged for better control performance as discussed in [21], [22]. Even with infinite accuracy and proper initialization, packet collisions are inevitable for homogeneous networks with smart sensors, since the resulting CoIL, and thus the timer values, can be exactly the same for multiple subsystems.

S. Wang, T. Farjam, and T. Charalambous are with the Department of Electrical Engineering and Automation, School of Electrical Engineering, Aalto University, Espoo, Finland. E-mails: {name.surname@aalto.fi}.

This work was supported by the Academy of Finland under Grant 320043. The work of T. Charalambous was supported by the Academy of Finland under Grant 317726.

In this work, we consider the LQG scenario in which multiple subsystems with decoupled linear dynamics and observations communicate over a time-slotted network and the aim is to efficiently share the limited resources for minimizing a quadratic cost. Additionally, we consider the case where subsystems are equipped with smart sensors which pre-process the measurements before transmission as it improves the estimation accuracy at the receiver side. In this regard, we utilize CoIL as a priority measure (by computing the timer values as proposed in [16]) and propose a distributed channel access method based on bit-wise arbitration. Similar to CAN protocol, during the contention phase (preceding data transmission in each frame), subsystems transmit their identifiers bit-by-bit and the subsystem with the dominant (to be explained later) identifier transmits its data on the network. However, unlike the standard CAN protocol, the identifiers in our work are assigned dynamically as in TOD [8] but such that channel access is given to the subsystem with the highest cost (e.g., CoIL). Our main contributions can be summarized as follows.

- Inspired by [8], we adopt a *hybrid priority* assignment scheme which consists of two segments: a dynamic and a static. The dynamic part is based on a mapping of the timer values, which ensures that higher priorities are given to subsystems with smaller timers, while the static segment guarantees collision-free channel access.
- We demonstrate how the *hybrid priorities* can be realized in CAN networks by manipulating the message identifiers and propose a synchronization method to enable fully distributed implementation.
- We provide experimental evidence of the efficacy of our proposed scheme over CAN bus by using hardware-in-the-loop simulations. The results indicate that our method provides collision-free channel access to subsystems with the highest CoILs. More importantly, our proposed synchronization method enables implementing the proposed scheme over CAN bus in a fully distributed manner.

The remainder of the paper is organized as follows. Section II provides the system model and necessary preliminaries. In Section III, we present our proposed priority assignment scheme and discuss how the dynamic priorities are determined. Section IV is dedicated to hardware implementation and how the synchronization issues over CAN-bus are resolved. In Section V we demonstrate the performance of our method for an experimental setup and finally draw conclusions and discuss future directions in Section VI.

Notation: Vectors and matrices are denoted by lowercase and uppercase letters, respectively. A random vector x from a multivariate Gaussian distribution with mean vector μ and covariance matrix X is denoted by $x \sim \mathcal{N}(\mu, X)$. $\mathbb{S}_+^n$ is the set of n by n positive semi-definite matrices. The transpose matrix of matrix A is denoted with A^T, its inverse with A^{-1}, and $\rho(A)$ denotes the spectral radius of A. $f^n(\cdot)$ is the n-fold composition of $f(\cdot)$ with the convention that $f^0(X) = X$, and $g \circ f(\cdot) \triangleq g(f(\cdot))$. $\mathbb{E}\{\cdot\}$ represents the expectation of its argument.

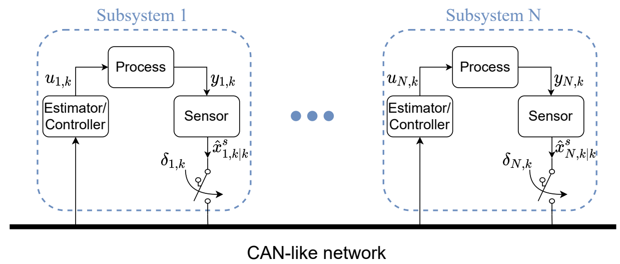

Fig. 1. The considered NCS in which N subsystems compete at each time step k for accessing the channel in order to transmit their most recent state estimate over the CAN-like network.

II. Problem Formulation and Preliminaries

The schematic diagram of the considered NCS is depicted in Fig.1 where N subsystems communicate over a shared network. This section provides the necessary preliminaries and modeling of the various components of each subsystem.

A. Dynamics

The dynamics of each subsystem i is described by the following linear time-invariant (LTI) process in discrete time:

$$x_{i,k+1} = A_i x_{i,k} + B_i u_{i,k} + w_{i,k}, \tag{1}$$

where $x_{i,k} \in \mathbb{R}^{n_i}$ and $u_{i,k} \in \mathbb{R}^{m_i}$ are the local states and control inputs at time step k, respectively. A_i and B_i are the system and input matrices of appropriate dimensions and we consider all subsystems are unstable, i.e., $\rho(A_i) > 1$ for all i. The dynamics are affected by the random disturbance $w_{i,k} \sim \mathcal{N}(0, W_i)$.

B. Sensing

The sensors measurements are described as

$$y_{i,k} = C_i x_{i,k} + v_{i,k}, \tag{2}$$

where $y_{i,k} \in \mathbb{R}^{p_i}$ is the noisy output of i at k measured by the local sensor, C_i is the measurement matrix and $v_{i,k} \sim \mathcal{N}(0, V_i)$ is the measurement noise which is assumed to be mutually independent from the plant disturbance. We consider the scenario in which the local sensors are smart, i.e., they have the computational capability for pre-processing their measurement data before transmission. This allows them to compute the *a posteriori* state estimate at each time k, denoted by $\hat{x}^s_{k|k}$, through the following recursion

$$\hat{x}^s_{i,k|k-1} = A_i \hat{x}^s_{i,k-1|k-1} + B_i u_{i,k-1}, \tag{3a}$$
$$P^s_{i,k|k-1} = h_i(P^s_{i,k-1|k-1}), \tag{3b}$$
$$K_{i,k} = P^s_{i,k|k-1} C_i^T (C_i P^s_{i,k|k-1} C_i^T + V_i)^{-1}, \tag{3c}$$
$$\hat{x}^s_{i,k|k} = \hat{x}^s_{i,k|k-1} + K_{i,k}(y_{i,k} - C_i \hat{x}^s_{i,k|k-1}), \tag{3d}$$
$$P^s_{i,k|k} = g_i \circ h_i(P^s_{i,k-1|k-1}), \tag{3e}$$

which corresponds to running a local Kalman filter at the sensor side. The functions h, $g : \mathbb{S}_+^n \to \mathbb{S}_+^n$ for computation of the *a priori* and *a posteriori* error covariances in (3b) and (3e), respectively, are defined as

$$h_i(X) \triangleq A_i X A_i^T + W_i, \tag{4}$$
$$g_i(X) \triangleq X - XC_i^T(C_i X C_i^T + V_i)^{-1} C_i X. \tag{5}$$

We assume that the pair (A_i, C_i) is observable and $(A_i, W_i^{1/2})$ is controllable which guarantees that $g_i \circ h_i(X) = X$ has unique positive semi-definite solution which we denote by $\overline{P}_i$. Since $P^s_{i,k|k}$ converges to $\overline{P}_i$ exponentially fast, we safely assume that $P^s_{i,k|k} = \overline{P}_i$ for all k [23].

Due to the network constraints, only one subsystem can transmit its data over the bus. This is captured by the following constraint

$$\sum_{i=1}^{N} \delta_{i,k} \leq 1, \qquad \forall k, \tag{6}$$

where $\delta_{i,k}$ represents whether i transmits at k such that

$$\delta_{i,k} = \begin{cases} 1, & \text{sensor } i \text{ transmits } \hat{x}^s_{i,k|k} \text{ at } k, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Note that if $\delta_{i,k} = 1$ then the corresponding data packet is guaranteed to be received successfully at its destination.

C. Control and estimation

We consider the LQG scenario and adopt a controller such that it minimizes the linear quadratic cost given by

$$J_\infty = \lim_{K\to\infty} \frac{1}{K} \mathbb{E}\left\{ \sum_{k=0}^{K-1} \sum_{i=1}^{N} \left(x_{i,k}^T Q_i x_{i,k} + u_{i,k}^T R_i u_{i,k} \right) \right\}, \tag{8}$$

where Q_i and R_i are positive semi-definite and positive definite weighting matrices of appropriate dimensions. The *certainty equivalent controller* is optimal in this setting and the inputs are given by

$$u_{i,k} = L_i \hat{x}_{k|k}, \tag{9}$$

where L_i is the stabilizing feedback matrix of appropriate dimensions. Assuming that the pairs (A_i, B_i) and $(A_i, Q_i^{1/2})$ are controllable and observable, respectively, L_i exists and is unique (see [24, Ch. 8]). Moreover, $\hat{x}_{i,k|k}$ is the state estimate which is computed by the estimator. Let $t_{i,k} \triangleq k - \max\{\kappa \leq k : \delta_{i,\kappa} = 1\}$ denote the time elapsed since the most recent transmission of i. Then the state estimate and the associated error covariance at the estimator side are given by

$$\hat{x}_{i,k|k} = (A_i + B_i L_i)^{t_{i,k}} \hat{x}^s_{i,k-t_{i,k}|k-t_{i,k}}, \tag{10}$$
$$P_{i,k|k} = h_i^{t_{i,k}}(\overline{P}_i), \tag{11}$$

where $h_i(X)$ is defined in (4) and we assume $h_i^0(X) \triangleq X$ for compact representation. Although the inputs are applied regardless of availability of a data packet, $t_{i,k}$ depends on the $\delta_{i,k}$ which influences the estimated state (10) and thus the input in (9).

D. Cost of information loss (CoIL)

We first reintroduce CoIL and then discuss how it can be calculated for the setup considered in this work. Let $J_{i,k}$ be the one-step cost of subsystem i at time k, i.e., $J_{i,k} = x_{i,k}^T Q_i x_{i,k} + u_{i,k}^T R_i u_{i,k}$ (see (8)). Our aim is to solve the following optimization problem to determine the decision variables $\delta_{i,k}$.

Problem 1.

$$\min_{\delta_{1,k},\ldots,\delta_{N,k}} \quad \sum_{i=1}^{N} \mathbb{E}\{J_{i,k} | \delta_{i,k}\},$$
$$\textit{subject to} \quad \sum_{i=1}^{N} \delta_{i,k} \leq 1,$$

where the constraint ensures collision-free transmission. This problem can be solved in a distributed manner by adopting the concept of CoIL, which as the name suggests, indicates how much the cost increases when a subsystem does not receive a data packet, i.e.,

$$\mathrm{CoIL}_{i,k} = \mathbb{E}\{J_{i,k} | \delta_{i,k} = 0\} - \mathbb{E}\{J_{i,k} | \delta_{i,k} = 1\}. \tag{12}$$

Similar to the approach of [15], by considering which components of the cost are affected by the channel access decision we can obtain

$$\begin{aligned} \mathrm{CoIL}_{i,k} &= \mathrm{tr}\left(\Gamma_i \left[\mathbb{E}\{P_{i,k|k} | \delta_{i,k} = 0\} - \mathbb{E}\{P_{i,k|k} | \delta_{i,k} = 1\}\right]\right) \\ &= \mathrm{tr}\left(\Gamma_i \left[h_i^{t_{i,k-1}+1}(\overline{P}_i) - \overline{P}_i\right]\right), \end{aligned} \tag{13}$$

where $\Gamma_i = L_i^T(B_i^T \Pi_i B_i + R_i) L_i$ and Π_i is the unique solution of the following discrete-time algebraic Riccati equation (DARE)

$$\Pi_i = A_i^T \Pi_i A_i + Q_i - L_i^T(B_i^T \Pi_i B_i + R_i) L_i. \tag{14}$$

In essence, CoIL can be interpreted as how much the error covariance at the estimator side decreases if it receives the sensor's data packet.

Consequently, the optimal channel access decisions at k for solving Problem 1 is given by $\arg\max_i \mathrm{CoIL}_{i,k}$. Therefore, $\mathrm{CoIL}_{i,k}$ corresponds to transmission priority of i at k and since each sensor can compute this value without information exchange as per (13), it can be used for distributed implementation.

E. The CAN bus

The CAN bus is a well-known priority based protocol used in several industries and is the most prevalent in automotive industries for connecting various components. The CAN protocol uses a non-destructive mechanism with binary count down for contention resolution. Bit values 1 and 0 are encoded as recessive and dominant, respectively, and the dominant signal always prevails upon recessive [25]. Before transmitting their data, subsystems transmit their fixed-length identifiers, i.e., priority, on the bus bit-by-bit and listen on the bus. If a subsystem sends a recessive bit but receives a dominant one, it means that other subsystems with higher priorities are trying to access the network. Therefore, this subsystem backs off while the dominant subsystems continue this procedure. When all the identifiers are distinct, a single subsystem will be the only winner at the end of the contention period and it proceeds to transmit its data without any collision.

III. Priority assignment and contention resolution

In this section, we first discuss how the timer values can be utilized for contention resolution in CAN-like network. Then, to overcome the shortcomings of adopting timers, we present our hybrid priority scheme and demonstrate how its implementation in CAN-like networks can solve Problem 1 in a distributed manner.

In essence, the timer-based mechanism grants channel access to the subsystem with the smallest timer value which is given by

$$\tau_{i,k} = \frac{\lambda}{\mathrm{CoIL}_{i,k}}, \tag{15}$$

where the positive constant λ is identical for all subsystems. Despite the significant reduction of communication overhead, adopting this mechanism in NCSs considered here can result in packet collisions, as aforementioned. To overcomes this, we propose a hybrid priority scheme which requires a fixed contention period and the contention can be resolved by bit-wise arbitration which facilitates its implementation on CAN-like networks.

Let $\hat{m}_{i,k}$ denote the identifier of subsystem i, at time step k, which represents its priority. Note that unlike the original CAN protocol, this value is time-varying. To minimize the one-step cost, the timer value given in (15) should be mapped to identifier $\hat{m}_{i,k}$ such that the subsystem with the smallest timer, or, equivalently, the highest CoIL, transmits its data. Based on the contention arbitration mechanism, the subsystem with the smallest timer is assigned with the smallest identifier $\hat{m}_{i,k}$ representing higher priority in CAN. Let $f(\cdot)$ be a continuous, nonnegative, and monotonically nondecreasing function; then, the identifier can be determined by $\hat{m}_{i,k} = [f(\tau_{i,k})]$ where $[\cdot]$ denotes the function round to the nearest integer. Assuming that the identifier consists of n contention bits, it is constrained between 0 and $\hat{m}_{\max}$, where $\hat{m}_{\max} = 2^n - 1$ denotes the identifier's upper bound and thus the identifier assignment could be refined as

$$\hat{m}_{i,k} = \begin{cases} 0 & \text{if } \hat{m}_{i,k} \leq 0, \\ [f(\tau_{i,k})] & \text{if } \hat{m}_{i,k} \leq \hat{m}_{\max} \\ \hat{m}_{\max} & \text{otherwise.} \end{cases}, \tag{16}$$

Although the subsystem with the smallest timer value has also the highest priority to transmit in this scenario, collisions are possible since multiple subsystems can have identical identifiers. Inspired by [8], we propose the hybrid priority scheme to ensure contention-free transmission. In the hybrid priority scheme, an identifier is comprised of two parts: a) *a dynamic identifier,* which is determined by CoIL as per (15) and (16) and occupies the most significant bits, b) *a static identifier*,[1] which is predefined with a unique number and occupies the least significant bits. Herein, $\hat{m}_{i,k}$ represents the dynamic identifier.

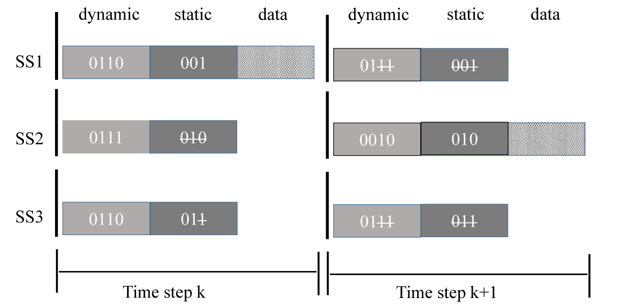

Fig. 2. Three subsystems competing for channel access using bit-wise arbitration with 0 being the dominant bit. At time step k, SS2 and SS3 lose the dynamic and static priority contention, respectively, and SS1 wins the contention. At time step $k+1$, SS2 has the dominant dynamic identifier and wins the contention.

When multiple subsystems attempt to access the network (almost) simultaneously, first the dynamic identifier are used to resolve the contention. Then, if multiple subsystems have the highest priority dynamic identifier, arbitration proceeds by comparing the static identifiers. Since the static identifiers are distinct, collision-free data transmission is guaranteed even if multiple subsystems have the same dynamic identifier. Fig. 2 depicts three subsystems competing for data transmission based on the hybrid priority scheme. At time step k, subsystem SS2 fails the dynamic priority contention due to its lower priority level i.e., higher timer value, and switches to listening mode. However, the other two subsystems continue to transmit their static identifiers which determines SS1 as the winner for data transmission due to the lower static identifier. As the next frame begins, subsystems re-compute their dynamic identifiers with respect to their current timers while the static identifiers remain unchanged. For the illustrative scenario in Fig. 2, at time $k+1$ SS2 has the smallest timer thus the smallest dynamic identifier and, hence, it transmits its data packet.

Proposition 1. *Implementing the proposed hybrid priority scheme guarantees mean-square stability of the NCS.*

Proof. Since the actuation link is perfect, mean-square stability of the NCS can be established by verifying whether $\mathbb{E}\{P_{i,k|k}\} < \infty$ for all i [26]. Similar to the arguments provided in [27, Proposition 1], since $\rho(A_i) > 1$ we can conclude that the off-duty cycle of each sensor is bounded which concludes the proof. In addition, the evolution of the error covariance at the estimators can be modeled by a finite Markov chain with a binary transition probability matrix and thus channel access converges to be periodic. □

IV. Hardware in the loop implementation

To realize the hybrid priority scheme for contention resolution on the hardware, the key factor is to guarantee

[1]Note that distributed coordination techniques can be used to obtain the static identifiers in a distributed fashion.

that all subsystems transmit (almost) simultaneously. To achieve this, one of the subsystems transmits a distinct `Sync Message` periodically as a trigger to synchronize the subsystems to get into contention on the network; see Fig. 3. The `Sync Message` has a unique identifier with the highest possible priority which distinguishes it from other hybrid identifiers.

Fig.3 demonstrates how subsystems are synchronized periodically by SS1, which broadcasts `Sync Message` according to its internal clock to synchronize all subsystems. After the reception of this message, all subsystems, including SS1 itself, compute their hybrid identifiers and contend for data transmission accordingly.

The hardware-in-the-loop implementation requires subsystems with the capacity to communicate on CAN network and also collects data from sensors. In addition, they need to compute the identifier $\hat{m}_{i,k}$ based on the timer as shown in Fig 4. STM32 Arm development board is a suitable candidate for our purpose as many open resources and standard libraries are available and also it is embedded with bxCAN controller. To validate the message on the network, CANoe and its CAN interface have also been introduced to collect messages. The micro-controller is STM32F103 Cortex-M3 with maximum CPU speed of 72 MHz, and the CAN transceiver is TJA1050 which could support up to 1 Mbps. Free real-time operating system is implemented considering it is free and easily portable. To simplify the realization, the dynamics of each subsystem is simulated in software and the necessary data is transmitted to the development board. There are several main tasks in the kernel, and the schedule is as follows.

- First, a queue is created before tasks start, which is always empty until a new token is written in by the task.
- SS1 starts to broadcast the `Sync Message` to synchronize all the subsystems connected to the shared medium.
- After the reception of `Sync Message`, one token is written into the queue waited by `CAN_trans_task` which is always in blocked state until the queue is not empty.
- Subsequently, `CAN_trans_task` exits the blocked state and computes the identifier $\hat{m}_{i,k}$ based on the timer value. Then the data packet is transmitted and the token is removed by this task. And the queue becomes empty and the task enters blocked state again.
- Another task will be utilized to check the status of transmission. If a message with different identifier is received, it means that the contention was lost and all the pending messages in bxCAN will be aborted and would not re-transmit again at current time step.
- The subsystem uploads the transmission status to the computer and downloads the dynamic identifier for the next time step.

Subsystem SS1 has the identical task set except it transmits the data packet directly without waiting for `Sync Message`. Thanks to the real time operating system, SS1 could still get into contention with others as they have the same task routine.

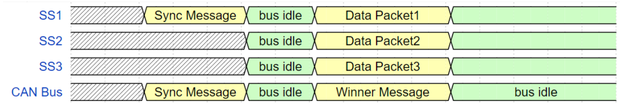

Fig. 3. Three subsystems are synchronized by SS1 and contend to transmit on the shared medium.

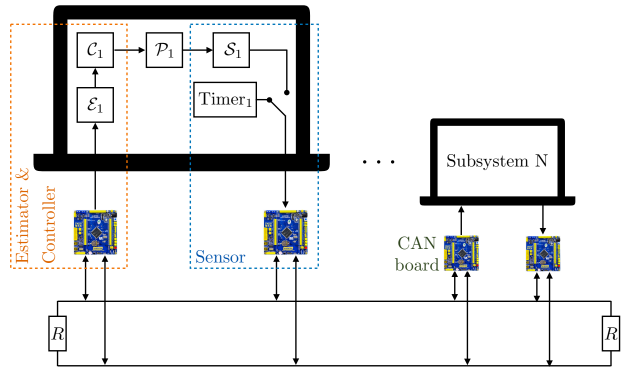

Fig. 4. Hardware in the loop diagram, in which the CAN board is used the MCU to compute identifiers and the CAN transceivers to communicate in the network.

V. Experimental Results

To validate the performance of the hybrid priority scheme, we first consider a NCS with three subsystems as depicted in Fig. 4. In our hardware-in-the-loop setup, one real CAN board is in charge of broadcasting the synchronization message on the bus. After transmission of this message, all subsystems, including the synchronizer, compete for channel access as described in Section III. The hardware interact over CAN bus and determine the channel access decision $\delta_{i,k}$, while the effect of the outcome on dynamics, control, estimation and sensing is simulated. the typical standard 11-bit long identifier is implemented, where the 8 most significant bits are used for the dynamic identifier and the static identifier occupies the three least significant bits. Consequently, the lowest priority identifier which is also the upper bound is $\hat{m}_{\max} = 2^8 - 1 = 255$ while the maximum number of subsystems which can share the network in a collision-free manner is $2^3 = 8$. The speed of CAN network is set as 100 kbps, and the length of cable is 8 m.

The LTI stochastic process of each subsystem is modeled by considering the following parameters

$$A_1 = \begin{bmatrix} 4.3 & 0 \\ 0 & 1.8 \end{bmatrix}, A_2 = \begin{bmatrix} 1.5 & 0.8 \\ 0 & 2.3 \end{bmatrix}, A_3 = \begin{bmatrix} 1.2 & 0.8 \\ 0 & 1.8 \end{bmatrix},$$

and $B = C = I_{2\times 2}$ where $I_{n\times n}$ denotes the n by n identity matrix. In addition, the covariance matrices of model disturbances and measurement noises are given by

$$W_i = \begin{bmatrix} 0.03 & 0 \\ 0 & 0.01 \end{bmatrix}, \quad V_i = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.05 \end{bmatrix},$$

and the weighting matrices of the quadratic cost are chosen as $Q = I_{2\times 2}$ and $R = 0.01 I_{2\times 2}$. The constant value of the timers in our setup is $\lambda = 20$ which yields $\tau_{i,k} = \frac{20}{\mathrm{CoIL}_{i,k}}$.

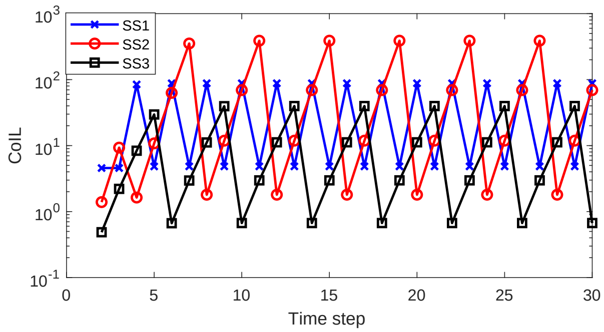

Fig. 5. The CoIL value evolution of each subsystem over time step given by the experimental setup.

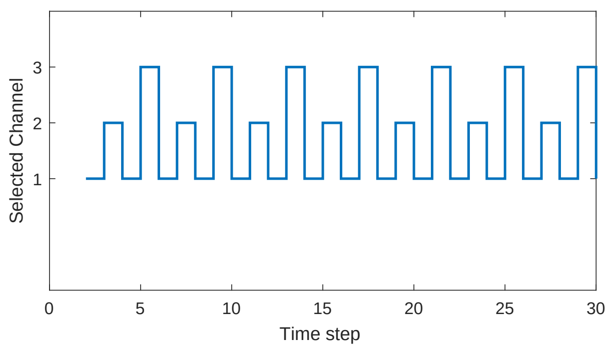

Fig. 6. The subsystem winning the contention for data transmission at each time step based on the identifiers assigned with respect to Fig. 5.

To assign the dynamic identifiers based on the timer values, we simply choose $f(\tau_{i,k}) = \tau_{i,k}$ in (16).

Fig. 5 demonstrates the evolution of CoIL for each subsystem running on real CAN network over 30 time steps and Fig. 6 exhibits which subsystem is being granted the channel access to transmit data, respectively. The subsystem with highest CoIL and thus the smallest timer value wins the first segment of contention resolution due to its higher priority dynamic identifier. The hybrid priority scheme is able to guarantee that transmission is collision-free despite the possibility of multiple subsystems having the same dynamic identifier. For instance, at the 6th time step, SS1 and SS2 have the same dynamic identifier as their timer values are close enough such that they are mapped to the same dynamic identifier. However, the static priority of SS1 is higher meaning that it wins the contention and transmits the data packet while SS2 backs off.

To evaluate the performance of our method in larger networks, we compare its performance with round robin, where subsystems transmit in a predefined sequence, and a centralized version of the timer-based mechanism for collision-free channel access, where a central unit grants channel access to the subsystem with the smallest timer value. The results depicted in Fig. 7 are obtained by considering that the dynamics of half of the subsystems are modeled by A_{I} and the rest by A_{II} while the other parameters are as

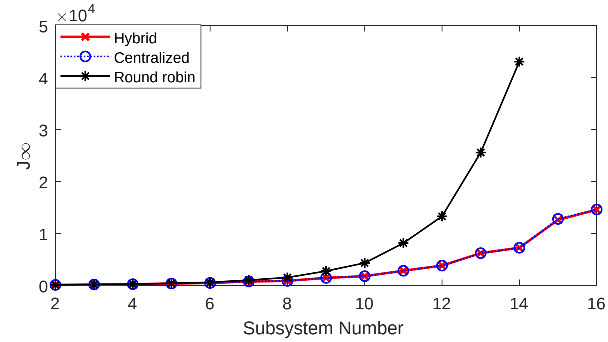

Fig. 7. The performance comparison between round robin, centralized module and hybrid priority scheme for various network sizes.

aforementioned.

$$A_{\mathrm{I}} = \begin{bmatrix} 1.4 & 0.1 \\ 0 & 1.1 \end{bmatrix}, A_{\mathrm{II}} = \begin{bmatrix} 1.1 & 0 \\ 0 & 1.1 \end{bmatrix}.$$

Due to the exponential growth of CoIL, as the number of subsystems that compete for single available channel increases, a more refined mapping of the timers is needed. Otherwise, the all timers could be small enough such that channel access is granted solely based on the static identifiers. Thus, we use the extended CAN identifier which has 29 contention bits, 16 of which represent the dynamic identifier giving $\hat{m}_{\max} = 2^{16} - 1 = 65535$ and 5 bits accommodate the static identifiers. Moreover, we use $\lambda = 1000$ for evaluating the timers. Fig. 7 demonstrates how the choice of channel access scheme and the number of involved subsystems affect the performance in terms of the cost (8). As expected, the distributed channel access provided by the hybrid scheme performs exactly as the centralized method, irrespective of the size of the network. Furthermore, despite the small loss of performance in small networks, round robin results in significantly higher cost as the size of the network grows.

VI. Conclusions and Future Directions

In this paper, we proposed a distributed channel access method for multiple subsystems equipped with smart sensors that share a limited capacity network. Based on the technique proposed for minimizing the one-step LQG cost in the timer-based mechanism, we proposed a hybrid priority scheme which also prioritizes subsystems based on timer values. Our proposed scheme is also capable of minimizing the one-step LQG cost in a distributed manner by granting channel access to the subsystem with the smallest timer value. In addition, our method guarantees collision-free channel access in all scenarios and the fixed-length bit-wise contention resolution facilitates its implementation on the mature hardware technology of CAN-like networks. Then, we investigated the hardware implementation on CAN bus and discussed how the synchronization issues can be addressed in order to ensure that all subsystems contend for channel access simultaneously without requiring a separate coordinator unit. Finally, we validated our method on an experimental setup consisting of three subsystems and compared the efficacy of our method to centralized schemes for larger networks.

Part of the ongoing research is to synchronize the subsystems without requiring a separate synchronization frame and take into account how the length of wires and the number of subsystems affect the synchronization. Other interesting directions include extending this method to multihop networks and imperfect wireless networks.

References

[1] 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 Journal of Automatica Sinica*, vol. 7, no. 1, pp. 1–17, Jan. 2020.

[2] P. Park, S. C. Ergen, C. Fischione, C. Lu, and K. H. Johansson, "Wireless network design for control systems: A survey," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 978–1013, 2018.

[3] R. A. Gupta and M.-Y. Chow, "Networked control system: Overview and research trends," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 7, pp. 2527–2535, Jul. 2010.

[4] L. Zhang, H. Gao, and O. Kaynak, "Network-induced constraints in networked control systems - a survey," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 403–416, 2013.

[5] M. Zanon, T. Charalambous, H. Wymeersch, and P. Falcone, "Optimal scheduling of downlink communication for a multi-agent system with a central observation post," *IEEE Control Systems Letters*, vol. 2, no. 1, pp. 37–42, Jan. 2018.

[6] K. Gatsis, A. Ribeiro, and G. J. Pappas, "Random access design for wireless control systems," *Automatica*, vol. 91, pp. 1–9, May 2018.

[7] Y. Ma, J. Guo, Y. Wang, A. Chakrabarty, H. Ahn, P. Orlik, and C. Lu, "Optimal dynamic scheduling of wireless networked control systems," in *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*, ser. ICCPS '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 77–86.

[8] G. C. Walsh and H. Ye, "Scheduling of networked control systems," *IEEE Control Systems*, vol. 21, no. 1, pp. 57–65, Feb. 2001.

[9] M. Tabbara and D. Nesic, "Input–output stability of networked control systems with stochastic protocols and channels," *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1160–1175, Jun. 2008.

[10] D. Christmann, R. Gotzhein, S. Siegmund, and F. Wirth, "Realization of try-once-discard in wireless multihop networks," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 17–26, Feb. 2014.

[11] L. H. Duc, D. Christmann, R. Gotzhein, S. Siegmund, and F. Wirth, "The stability of try-once-discard for stochastic communication channels: Theory and validation," in *IEEE Conference on Decision and Control (CDC)*. IEEE, Dec. 2015.

[12] M. H. Mamduhi, M. Vilgelm, W. Kellerer, and S. Hirche, "Prioritized contention resolution for random access networked control systems," in *IEEE Conference on Decision and Control (CDC)*, Dec. 2017.

[13] A. Molin, C. Ramesh, H. Esen, and K. H. Johansson, "Innovations-based priority assignment for control over CAN-like networks," in *IEEE Conference on Decision and Control (CDC)*, Dec. 2015.

[14] A. Molin, H. Esen, and K. H. Johansson, "Scheduling networked state estimators based on Value of Information," *Automatica*, vol. 110, p. 108578, Dec. 2019.

[15] T. Charalambous, A. Ozcelikkale, M. Zanon, P. Falcone, and H. Wymeersch, "On the resource allocation problem in wireless networked control systems," in *IEEE Conference on Decision and Control (CDC)*, Dec. 2017.

[16] T. Farjam, T. Charalambous, and H. Wymeersch, "A timer-based distributed channel access mechanism in networked control systems," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 5, pp. 652–656, May 2018.

[17] ——, "Timer-based distributed channel access for control over unknown unreliable time-varying communication channels," in *European Control Conference (ECC)*, Jun. 2019.

[18] ——, "Timer-based distributed channel access in networked control systems over known and unknown Gilbert-Elliott channels," in *European Control Conference (ECC)*, Jun. 2019.

[19] T. Farjam, H. Wymeersch, and T. Charalambous, "Distributed channel access for control over unknown memoryless communication channels," arXiv:2103.06048 [eess.SY], Mar. 2021.

[20] A. Flammini and E. Sisinni, "Wireless sensor networking in the internet of things and cloud computing era," *Procedia Engineering*, vol. 87, pp. 672–679, 2014.

[21] V. Gupta, B. Hassibi, and R. M. Murray, "Optimal LQG control across packet-dropping links," *Systems & Control Letters*, vol. 56, no. 6, pp. 439–446, Jun. 2007.

[22] L. Schenato, "Optimal estimation in networked control systems subject to random delay and packet drop," *IEEE Transactions on Automatic Control*, vol. 53, no. 5, pp. 1311–1317, Jun. 2008.

[23] J. B. M. Brian D. O. Anderson, *Optimal Filtering*. Dover Publications, 2012.

[24] K. J. Astrom, *Introduction to Stochastic Control Theory*. Dover Publications Inc., 2006.

[25] M. D. Natale, H. Zeng, P. Giusto, and A. Ghosal, *Understanding and Using the Controller Area Network Communication Protocol*. Springer-Verlag New York, 2012.

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

[27] D. Han, J. Wu, H. Zhang, and L. Shi, "Optimal sensor scheduling for multiple linear dynamical systems," *Automatica*, vol. 75, pp. 260–270, Jan. 2017.