Distributed control plane for safe cooperative vehicular cyber physical systems

Foukalas, Fotis; Pop, Paul

Published in:
IET Cyber-Physical Systems: Theory and Applications

Link to article, DOI:
10.1049/iet-cps.2019.0034

Publication date:
2019

Document Version
Publisher's PDF, also known as Version of record

IET Cyber-Physical Systems: Theory & Applications

Research Article

IET Journals
The Institution of Engineering and Technology

Distributed control plane for safe cooperative vehicular cyber physical systems

ISSN 2398-3396
Received on 6th May 2019
Revised 14th September 2019
Accepted on 9th October 2019
doi: 10.1049/iet-cps.2019.0034
www.ietdl.org

Fotis Foukalas[1] ✉, *Paul Pop*[1]

[1]*Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kongens Lyngby 2880, Denmark*
✉ *E-mail: fotisf@dtu.dk*

Abstract: Cooperative vehicular cyber physical systems build a group of entities that can accomplish a cooperative task using the distributed control approaches. To this end, such a cooperative task can be coordinated and managed by a distributed control plane that will be able to encapsulate all the required functionality in a layered architecture providing the required interoperability. Here, the authors propose such a distributed control plane that consists of the cooperative awareness layer, the communication layer and the distributed control layer. Wireless communications play an important role for the mobility provision, taking into account different constraints in order to provide high reliability and low latency. A simulation environment is considered with a leader–follower control format, where the reliability is evaluated. Further, a distributed safety monitoring approach is devised, given a control diagram and mapping of the events to the different components. The event monitoring relies on the self-triggered approach, where a use case is evaluated to highlight the impact of the input and output delays to the model predictive control component of the overall distributed control diagram including the calculation of the number of triggered events.

1 Introduction

Cyber physical systems (CPSs) are computer-based machines that integrate different digital components, such as computer architecture, software technologies and networking protocols. There are numerous examples of CPSs available already in the market and some future applications considered nowadays. For example, the unmanned aerial vehicles, mobile vehicle robots and autonomous cars are considered to be CPS applications that we can call vehicular CPS (vCPS) [1]. Although many CPS applications can be found in the literature, cooperative solutions that the individual CPSs are able to accomplish a common cooperative task is not well specified yet. Such cooperation is doable using the wireless communications, providing a high dependability level [2]. A few works focused on the cooperation of CPS applications is discussed below.

In [3], the authors provide an end-to-end network connectivity solution for autonomous teams of robots. The main design consideration is a controller, which guarantees the network connectivity through robust wireless communications among the mobile robots that aim to accomplish a particular assigned task. In [4], the authors proposed a cooperative adaptive cruise control solution required for vehicle platooning. The goal is to provide cooperative manoeuvres using wireless communications among the cars, keeping always their string stability. In [5], the authors provide a decentralised formation control solution for unmanned aerial vehicle (UAV) based on the formation stability conjecture, which is a key component to accomplish their cooperative task. In [6], the authors proposed a distributed controller application to retain the synchronous time-varying formation control for robots. The sampled data with communication delays are transmitted among the robots to accomplish their cooperative task. In [7], a more theoretical foundation of a cooperative control approach is provided for the time-varying formation control. Finally, in [8], a practical and experimental demonstration is provided for cooperative team of robots. Most of the works described above focus either on the networking level or the control functionality.

In this paper, we provide an integrated solution, where a distributed control plane (DCP) encapsulates the context-awareness messages, wireless communications and distributed control. Such a design approach is essential for new integrated CPS solutions, where cyber and physical components are integrated at all levels with the safety provision [9]. The robust system operation is essential to guarantee the reliability from communication and control point of view [10]. To this end, at the bottom of the proposed architecture, a distributed control layer is developed, which controls the cooperative tasks, e.g. manoeuvres in a leader–follower use case. The distributed control is able to translate the messages at the application layer into events, dealing with both the reliability at the communication layer and the safety at the control layer. To be more specific, we first develop a leader/follower distributed control approach for vCPS. Next, the reliability and latency of the communications layer that transmit messages from leader to follower is considered. Finally, the safety monitoring is studied by providing a safety event analysis and distributed event monitoring employing a self-triggered control approach. Example use cases for mobile robots are simulated in order to highlight the impact of the reliability and safety constraints into the leader/follower vCPS application scenario. Our design follows the system level design principles for layer-based and component-based model designs [9]. Future work is considered to evaluate the proposed plane assuming a large amount of vCPS and different types of hazardous events.

It is evident from the above that there is no such an integrated solution in the literature. Nevertheless, we would like to present a few more related works that we have found interesting in the context of an integrated solution and its components. In [1], many works related to context-awareness and communications among the vehicles are described and compared. However, none of them provide a full protocol stack solution through a control plane that can combine the required functionalities. On the other hand, they were not designed for vCPS applications. In [11], the authors provide a distributed solution taking into account the communication delays without paying attention to the context-awareness layer. Further, the works focused on the context-awareness among the vehicles or CPSs, such as [12, 13], do not deal with the distributed control level. The authors in [14] proposed communication strategies for cooperative tasks in vehicle platooning, providing a good use case for vCPSs. The authors in [15] proposed a non-cooperative solution, which is also not considered to be a safety event monitoring for a more integrated solution in vCPS applications. Finally, the authors in [16] deal with the optimal communication design in distributed control for CPS application in the smart grid. In [17], a multi-layered context-aware

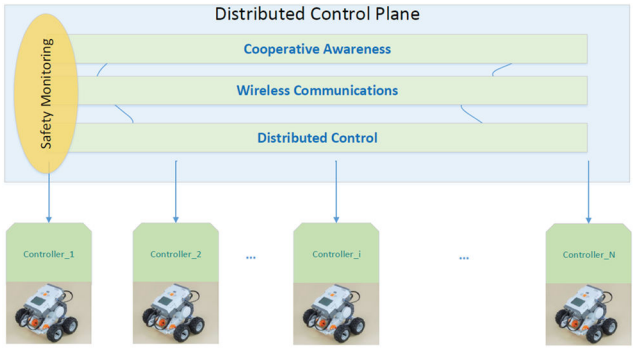

Fig. 1 *DCP reference architecture*

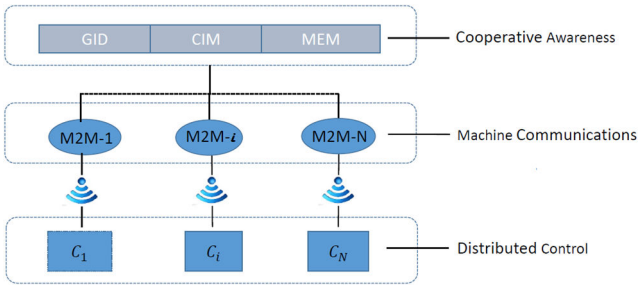

Fig. 2 *DCP functional architecture*

architecture is also introduced, where the vehicular social networks and context-aware vehicular security are integrated for a dynamic parking service application scenario. Nevertheless, our work includes all layers offering a more complete solution and focusing on vCPS using both simulation and experiments to proof the concept of the proposed DCP. We believe that such a plane could be a part of future standardisation activities for vCPS [18].

The rest of this paper is organised as follows. Section 2 provides the functional architecture of the DCP. Section 3 presents the details about the distributed control for vCPS. Section 4 discusses the ultra-reliable low latency distributed solution. Section 5 concludes this work.

2 Distributed control plane

A vCPS is considered in our study, where all vCPSs communicate through wireless communications and machine-to-machine (M2M) type of communications. The M2M communication layer transmits messages that are facilitating cooperative awareness and distributed control. This is considered to be a complex modelling and thus, we propose a DCP using a layered approach, which aims to provide the following functionality to support the cooperation among the vCPS [3]:

i. Providing the point-to-point connectivity information in order to ensure the end-to-end network integrity.
ii. Mapping of vCPS information into particular local control tasks for supporting the cooperative global task.
iii. Actuating the cooperation by controlling the cooperative task locally and globally.

Such a layered architecture also guarantees the interoperability (i.e. different types of communication protocols and distributed control) required for successful implementations.

The proposed DCP integrates three different layers of functionalities (Fig. 1). On top of the protocol stack, a cooperative awareness layer is situated that is responsible to transmit and receive the messages to each vCPS. Such a layer implements, for example, CAM (cooperative awareness message) application protocol that conveys the useful information related to the cooperative task. The CAM application protocol is considered to be on top of a communication protocol that is an M2M protocol, providing a single-hop communication (similar to vehicle-to-vehicle (V2V) communications) [19]. At the bottom of the architecture, a distribute control layer is situated, which is responsible for mapping the CAM information exchanged among the vCPSs to a particular control functionality. The distributed control functionality is considered to be an aperiodic wireless control application [20]. The specification of the cooperative task is required and the corresponding distributed control protocol that consists of the local actuators and the global ones is discussed later in this paper.

Different implementation strategies could be adopted based on the proposed networking and control architecture according to the application scenarios and use cases. This is actually the main goal of the proposed DCP, to accommodate many applications and different use cases. This is considered as a kind of reference model to design and develop the required functionality from any party in the future. To some extent, this recalls somehow an open reference model to guarantee the interoperability among the different industrial vendors. Although a DCP could be conceptualized in different ways, we attempt below to provide as much as possible a generic one that will be turned out an efficient solution with our practical implementation for vCPSs below.

The functional architecture of the DCP is depicted in Fig. 1, which consists of the following elements in detail (Fig. 2):

- At the CAM application protocol, the group identifier, the context information message and the manoeuvre event message are defined in order to support the cooperative vehicular task.
- At the communication protocol, an M2M protocol is implemented per ith, $\forall i \in [1, N]$, vCPS over wireless links. We assume that there are small base stations providing the required network resources.
- Distributed controller C_i, $\forall i \in [1, N]$, vCPSs are developed to provide the local and global tasks.
- Safety monitoring and control is considered to be the layer to retain the safe cooperative tasks.

The rest of this paper focuses on the design of the distributed control for a leader/follower use case, the reliable low-latency communications and the distributed safety monitoring.

3 Distributed control for vCPS

To design a distributed control system for vCPS, the following are required:

- to provide distributed control techniques for a linear control system,
- to provide a communication control channel to send critical control messages.

To this end, we first describe below the distributed control and next, the communication control channel details.

We assume an adaptive cruise control (ACC) system in order to model a leader–follower formation control for our vCPS use case. The leader–follower formation control for mobile robots using the model predictive control has been recently proposed in [21]. An ACC system uses its two modes: (a) speed control mode and (b) space control mode, where the first regulates the vehicle speed at a driver-defined setting and the second to avoid a collision with the leader vehicle. Space control can be implemented based on constant spacing or on constant time gap. Moreover, the space control should be implemented with a particular car-following policy. For testing the controller behaviour when the driver chooses to change the gap setting, only two vehicles were used, one of them acts as the leading vehicle and the other one runs the ACC controller. The vehicle dynamics are considered according to the following open-looped cruise control transfer function in a Laplace transform:

$$H(s) = \frac{1}{s(0.5s + 1)}, \tag{1}$$

where s approximates the dynamics of the throttle body and vehicle inertia. The vehicle dynamics block has connection with the model

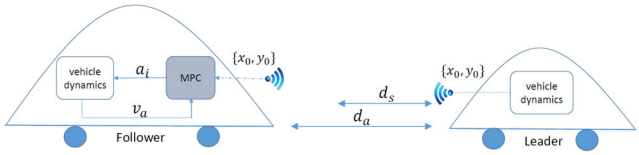

Fig. 3 *ACC enabled by MPC*

1: Initializs $k = 1$
procedure:
2: **while** Non-convergence **do**
3: Calculate current state: $x(k \mid k)$
4: Calculate control sequence: $u(\cdot \mid k)$
5: Substitute control input: $u(k) = u(k \mid k)$
6: Update iteration: $k = k + 1$
7: **end while**
end procedure

Fig. 4 *Algorithm 1: Linear MPC algorithm*

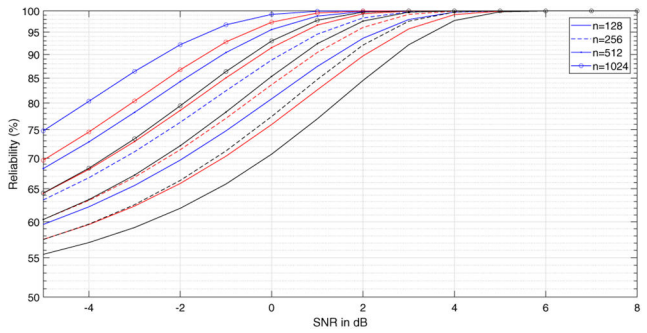

Fig. 5 *Reliability versus SNR in dB (blue line $l = n/3$, red line $l = n/2$ and black line $l = 2n/3$)*

predictive control (MPC) system, where the former gives the actual velocity v_a as an input to the latter. The MPC provides its own acceleration value a_i to the vehicle dynamics. Finally, the vehicle dynamics of the leader sends out through wireless communication the pair of velocity and position, x_0 and y_0. Fig. 3 depicts the specified safe distance d_s and the actual distance d_a that the overall ACC system must retain by interchanging between speed and headway modes of the control.

Also, the ACC consists of a linear MPC of the form

$$x(k+1) = Ax(k) + Bu(k) \tag{2}$$

$$y(k) = Cx(k), \tag{3}$$

where $x(k) \in \boldsymbol{R}^n$, $u(k) \in \boldsymbol{R}^m$, $y(k) \in \boldsymbol{R}^p$ denote the state, control input and measured output at the sampling instant k, respectively. It is a standing assumption that the system is both controllable and observable. Besides the dynamics, the system is saturated, and we conceptually write this control constraint as follows:

$$u \in \mathscr{U}, \quad u_{\min} \le u(k) \le u_{\max}, \tag{4}$$

where $\mathscr{U}$ is the control constraint polytope.

A linear MPC is an optimisation-based control law, and the performance measure is almost always a quadratic cost. Defining the positive definite matrices $\boldsymbol{H} = \boldsymbol{H}^{\mathrm{T}} \succ 0$ and performance weights $\boldsymbol{R} = \boldsymbol{R}^{\mathrm{T}} \succ 0$, the underlying goal is to find the optimal control input that minimises the infinite horizon performance measure, or cost as follows:

$$J(k) = \sum_{j=k}^{\infty} x^{\mathrm{T}}(j|k)\boldsymbol{H}x(j|k) + u^{\mathrm{T}}(j|k)\boldsymbol{R}u(j|k)\,. \tag{5}$$

In the unconstrained case, the solution to this problem is given by the linear quadratic controller. In the constrained case, however, there does not exist any analytic solution. Instead, the idea in MPC is to define a prediction horizon Z and approximate the problem with a finite horizon cost. Following the required analysis, one will conclude the following optimisation problem:

$$\min_{u} \sum_{j=k}^{k+Z-1} x^{\mathrm{T}}(j|k) + u^{\mathrm{T}}(j|k)\boldsymbol{R}u(j|k) \tag{6}$$

$$\text{s.t.} \quad u(k+j|k) \in \mathscr{U} \tag{7}$$

$$x(k+j|k) = Ax(k+j-1|k) + Bu(k+j-1|k)\,. \tag{8}$$

Under this premise, the MPC controller is implemented in Fig. 4.

The aforementioned solution is considered as a quadratic program (QP) [22]. Advances on solving QP problems for MPC applications can be found in the literature such as in [23]. However, we use the MPC model implemented from the MPC Toolbox in Matlab in order to implement the MPC in our system model [24]. Our overall MPC-based ACC implementation is similar to the one found in [25].

4 Ultra-reliable low-latency wireless communications

4.1 Design requirements

Wireless communications in vCPS is still an open challenge. A vCPS communications can be assumed as an *ad hoc* network application, where the conventional mechanisms cannot be used [26]. Cooperation with reliability and low latency is more important than having higher data rates. An interesting solution could be considered using wireless communications with short packets like an internet-of-thing application. Such a design should provide a type of communication protocol with short packets. Key design factors of such a protocol are the number of information bits l and the number of the overall packet sizes n, where n - l is considered to be the number of control bits. The rate approximation for a particular packet size n and information bits l for a specific packet error probability ϵ is given as follows [27]:

$$\boldsymbol{R}(n, \epsilon) \simeq C - \sqrt{\frac{V}{n}}Q^{-1}(\epsilon) + \frac{1}{2n}\log n, \tag{9}$$

where C and V are the capacity and dispersion of an additive white gaussian noise (AWGN) channel, as given in (7) and (8) in [27]. Moreover, $Q^{-1}(\cdot)$ denotes the inverse of the Gaussian Q function. The packet error probability can be given from the following formula:

$$\epsilon(l, n) \simeq Q\left(\frac{nC - l + (\log n)/2}{\sqrt{nV}}\right). \tag{10}$$

Using the analysis above, we are going to derive the reliability $1 - \epsilon(l, n)$ and spectral efficiency S_e results for different l/n values. The l/n values could vary from 1/6 for low data rates to 2/3 for higher data rates according to the performance analysis of the WAVE control channels [28] (wireless access in vehicular environment (WAVE) protocol is considered for vehicular type of communications). Fig. 5 depicts the reliability in % versus SNR (signal-to-noise ratio) values in dB for different number of n values and ratios l/n. It is observed that a higher number of packet lengths n gives a higher reliability. This is due to the short packet design requirement as pointed out in [27]. The lower number of information bits, which means higher number of control bits, will result in an additional higher reliability.

Fig. 6 depicts the spectral efficiency (spectral efficiency is calculated as follows: $S_e = l/nR$) achieved for different n packet size values and ratio l/n. It is observed that the higher the information bits l, the higher the achievable spectral efficiency. A high packet size results also in higher spectral efficiency, as expected.

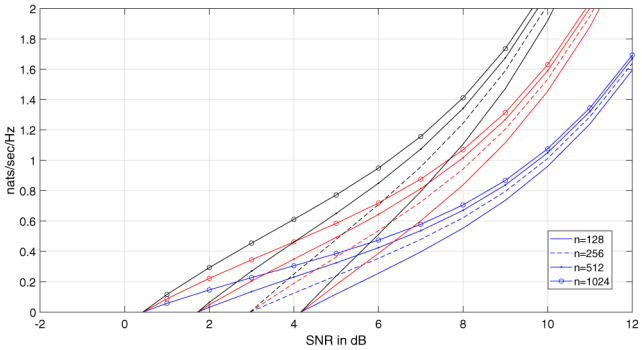

Fig. 6 *Spectral efficiency versus SNR in dB (blue line $l = n/3$, red line $l = n/2$ and black line $l = 2n/3$)*

1: $\forall i \in N$ vCPS, initialise.
2: Frame T with time slots $\tau_i = T/N, \forall i \in N$.
3: Matrix $\boldsymbol{H}[N, M]$, where n is the number of vCPS and M the number of l/n values.
procedure:
4: **while** $D_{\text{tot}} < D_0$ **do**
5: *Calculate* SNR $\tau_i, \forall i \in N$
6: *Activate* HM: assign τ_i for $per_i \leq per_0$
7: **if** $\Sigma(n_i) > N_0$ **then**
8: *Adapt* the frame length
9: **else if** $\Sigma(n_i) \leq N_0$ **then**
10: *Activate* HM: assign max $Se_i, \forall i \in N, \forall j \in M$
11: **end if**
12: **end while**
end procedure

Fig. 7 *Algorithm 2: HARA algorithm*

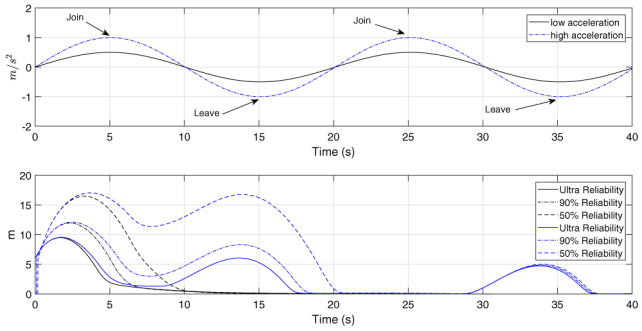

Fig. 8 *Acceleration partner in m/s^2 and the gap in meters (m) over time for different reliability constraints in the case of space gap (solid lines) and leader speed (dashed lines) outdated information*

Therefore, we aim to design a distributed solution, i.e. a communication protocol that can allocate the resources, i.e. channels, to each vCPS with high reliability and low latency. Ideas from both [27, 29] will be taken into account in order to conclude to our solution described below.

4.2 Distributed solution

We would like to design a distributed solution that can provide decisions about the overall frame structure in an adaptive fashion. In particular, the frame can have different sizes per vCPS use case instant retaining the overall delay at a specified level providing the reliability in parallel too. To this end, we formulate the problem below, where the spectral efficiency maximisation is considered as the objective function subject to the overall frame size N to not exist N_0, the overall delay D_{tot} to not exist D_0 (delay is considered to be the latency requirement that is equal to the overall adaptive frame T format) and the per_i be per vCPS, to be always below per_0 to retain the required reliability. The final problem formulation is as follows:

$$\begin{aligned} \max_{l_i, n_i} \quad & S_{e_i}(l_i, n_i) \\ \text{s.t.} \quad & \Sigma_i n_i \leq N_0 \\ & D_{\text{tot}} < D_0 \\ & \text{per}_i \leq \text{per}_0 . \end{aligned} \tag{11}$$

In order to solve such a problem, a heuristic algorithm that combines the Hungarian method is devised. Such heuristic algorithms are considered to be practical to many wireless communication use cases such as device to device [30]. The proposed algorithm is mainly devised to reduce the high computational complexity guaranteeing, however, the reliability and latency constraints. The assignment problem is a linear program, where in our case the number of sources (channels) equals the number of designations (vCPS), i.e. number of N. The algorithm is given in Fig. 7 (see Algorithm 2), where first the frame T is equally divided into time slots $\tau_i = T/N, \forall i \in N$, vCPS. Next, checking out for the total latency requirement $D_{\text{tot}} < D_0$ is guaranteed. In the sequel, the SNR values γ_i are calculated per vCPS and the hungarian method (HM) is activated to assign new τ_i for $\text{per}_i < \text{per}_0$, $\forall i \in N$, vCPS. At the second level, the HM is activated to keep the l/n values over a set M (the set M is the number of l/n values that are considered for each application scenario), which maximises the spectral efficiency by selecting the l - n value over the n packet size. Finally, the complexity of the proposed algorithm is equal to $O[(\max(N, M)^3)]$, where the complexity is getting lower and equal to $O[(\max(N)^3)]$ when $M = N$ [30]. The proposed algorithm is named heuristic adaptive resource allocation (HARA),which can be implemented both in centralised and decentralised fashions. However, further discussion on such implementations is out of the scope of this work that we are going to present in our future work within the SafeCOP project [2].

Regarding the leader–follower use case, we assume that the speed of a vCPS could be retained within 15–30 km/h, which is considered as a regular speed for mobile robot applications. Thus, our simulation results are carried out with a value of 5–10 m/s. With such a speed specification, the safe distance between two vCPS can be kept below 10 m (safe distance is the difference of the actual distance from the specified one). To this end, the following mode switchings are provided to the system:

- Speed control mode in the case of maintaining the target speed.
- Gap control mode in the case of maintaining the target space gap.

The mode switching is enabled by MPC and controlled by the input information x_0, y_0 sent out by the leader to the follower through the wireless communication. Such control information can be transmitted through wireless communications, e.g. a V2V service. We assume that the update time k of the MPC presented in Section 4.1 is the slot time τ by which a vCPS receives the message payload.

Under the assumptions above, the requirement is to keep the reliability and the latency high and low, respectively, for a particular payload message size. To this end, we provide simulation results obtained using our simulation setup. Fig. 8 depicts the safe gap in metres between the leader and the follower for different reliability constraints. The upper diagram depicts the acceleration partner in m/s^2, where the *Join/Leave* use cases take place. The latency is considered as $T = 50$ ms, which is an adequate time for processing five vCPS at $\tau_i = 5$ ms. This value should be extended to higher latency requirements in order to allocate more vCPS. In the case of ultra-reliability, i.e. 99.99%, the gap is below 10 m even in the case of high acceleration. However, the lower reliability makes the situation worst, exceeding the threshold of 10 m even in the case of a moderate reliability equal to 90%. We have also plotted results for different acceleration patterns, denoted as low and high, where the higher acceleration is possible to be managed over the time. We also highlight the points that *Join/Leave* manoeuvres can be carried out over time, where in low acceleration

Table 1 System safety constraints and requirements

Threats (T)	Safety constraints (events)	Safety requirements (monitoring)
$T1$	command drop	the command should not be dropped at the controller locally
	command delay	the command should not be delayed locally
$T2$	feebdack drop	the feedback should not be dropped
	feebdack delay	the feedback should be delayed
$T3$	communication drop	the communication should be error free
	communication delay	the communication channel should be with zero delay

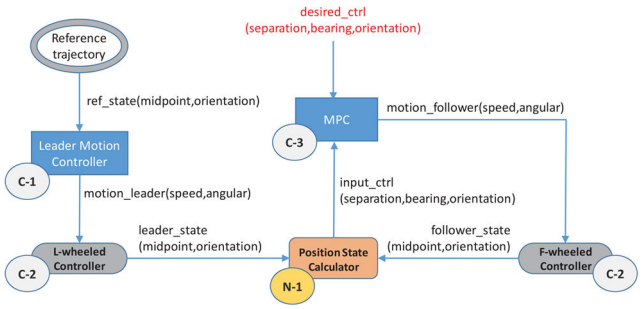

Fig. 9 *Control layer diagram: components and commands*

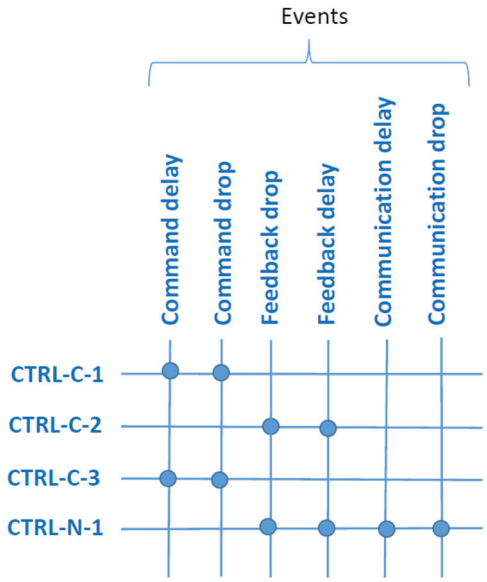

Fig. 10 *Mapping between the controller parts and the events*

the things are much doable to retain the ultra-reliability even at the beginning. Reliability objectives, given the spectral efficiency, also can be figured out by Figs. 5 and 6. The results in Fig. 8 depict the behaviour of the control system under certain reliability conditions from ultra (99.99%) to low (50%). It is also observed that over time, the gap is getting zero in the case of high acceleration while the *Leave* and before the next deceleration while the *Join*.

5 Distributed safety event monitoring for vCPS

The objective of the safety monitoring found in a cross-layer fashion on the DCP reference architecture (Fig. 1) is the active real-time safety monitoring in a distributed manner across the different vCPS maintaining the overall system safety. The design and specification of safety constraints according to particular safety requirements is essentially known as safety analysis. The safety analysis for our use case is described in the text below. Next, the distributed monitoring implementation is provided too. Our approach is similar to the safety monitoring framework for autonomous systems presented in [31]. However, we rely on the system-theoretic accident model and processes (STAMP) analysis found in [32, 33] and not in the hazard analysis such as hazard and operability study (HAZOP).

5.1 Safety event analysis

The safety analysis consists of the behaviour model specification such as safety constraints and the associated events and parameters. Such security constraints are specified in relation to the distributed control system. Those constraints are essential for the distributed safety monitoring that guarantees the non violation of the run-time safety of the system. Keeping the event monitoring concept in mind [33], we define the following safety constraints and requirements found in Table 1. More specifically, we classify three major threats related to the command, feedback and communication aspect of the overall distributed control (the distributed control has been defined already above in Section 3, assuming a leader/follower vCPS use case). It is recognised that the $T1$ thread is related to the command drop and delay, the $T2$ thread is related to the feedback drop and delay, and finally the $T3$ thread is related to the communication drop and delay. The corresponding requirements are also described in the table, such as explaining the need of actual event to be monitored from the distributed event monitoring system. For example, the event that the command had been dropped or delayed within the controller per vCPS needs to be monitored identically for the feedback. On the other hand, the communication among the distributed controller should be error free and without delays too.

In a STAMP analysis, a generic control layer diagram is required consisting of the main components of the overall distributed control system and commands that pass by the different components. The control layer diagram with its own components and commands is depicted in Fig. 9. More particularly, the component layer diagram relies on the MPC distributed control for leader/follower mobile robots discussed in Section 3. To this end, we can see the following main components: the leader motion controller, the wheeled controller either for the leader or follower, the position state calculator and the MPC by itself. The diagram shows the starting point of the considered vCPS use case that is the reference trajectory that the leader motion controller uses through the *ref_state* command, where the latter sends information to the wheeled controller regarding the speed and angular. The leader state is passed through the *leader_state* command to the position state calculator that acts like input to the MPC eventually specified by separation, bearing and orientation type of information. The *desired_ctrl* command is leading the MPC such as the follower can follow the leader properly. The three different types of controllers and the one node of the overall system are denoted as $C-1, C-2, C-3$ and $N-1$, respectively. It is obvious that the $N-1$ node is affected by the communications reliability, as discussed above.

Fig. 10 depicts now the mapping between the different controllers and the nodes as presented above with the safety events mentioned at the beginning of the safety analysis. Such a mapping is considered to be the final step of the safety analysis, where the events monitoring must lead to particular control actions. This is essential to devise the distributed event monitoring that follows in the next section. Briefly discussing the mapping diagram, we can identify the connection of the command drop and delay to the leader motion controller and the MPC. The feedback type of events is related to the wheeled type of controller and the communication with the position state calculator that plays the role of the control node. Finally, the feedback is also linked to the control node as well. The proposed mapping will be clarified in more detail with the distributed event monitoring that follows below. Notably, our focus below is not on the communications reliability and thus, on

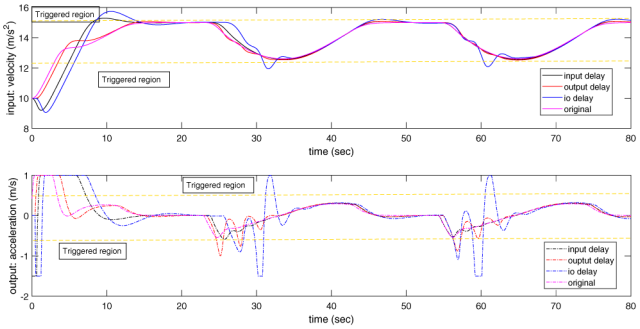

Fig. 11 *Effect of input and output delays to the MPC component*

Initialise time $t_{\max}$ upper bound, time t_j^{k+1}, $k = 0$
procedure:
while $t < t_{\max}$ **do**
  for $j = 1, .., N$ agents
  if $j = h$ // which means agent j is triggered **then**
    update $k_j = k_j + 1$ at t_j
    check condition $t_j^{k+1} \leq t_j^k + \xi_j$
  else if leader agent $t_{k_j+1}^{j-1}$ updating control **then**
    check condition again
  end if
  Calculate current state: $x(k_j \mid k_j)$
  Calculate control sequence: $u(\cdot \mid k_j)$
  Substitute control input: $u(k_j) = u(k_j \mid k_j)$
  Update iteration: $k_j = k_j + 1$
  Event concatenation to packet n_j, l_j
  end for
end while
end procedure

Fig. 12 *Algorithm 3: self-triggered distributed event monitoring algorithm*

the $N - 1$ node of the mapping diagram since we deal now with the events related to the control part of our use case.

5.2 Distributed event monitoring

This section deals with the distributed event monitoring solution. Such a solution monitors the safety in a cross-layer fashion by triggering events that take place from the communications layer to the control layer, as explained above. In [34], the authors proposed a cross-layer communication protocol though that is benchmarked in a security vehicle monitoring use case. Our approach below is considered to be a cross-layer approach among the different layers of the DCP discussed above. Hence, both communication and control layers are considered for safety event monitoring of the vCPS.

In particular, we opt to use a sort of risk-aware MAC protocol that is a single channel protocol and has been proposed for a simple highway with one lane in which all the vehicles are moving in the same direction [29]. In this case, the vehicle segment is divided into two segments: a contention-based segment (e.g. carrier-sensing multiple access), responsible for transmitting–warning messages in emergency situations, and a contention-free segment (e.g. time division multiple access) and used for delivering beacon messages. About monitoring, there are two strategies to follow: either aperiodic or periodic as discussed in [20, 35], respectively. The aperiodic could be considered self-triggered control and the periodic event-triggered control according to [20]. However, according to the safety monitoring framework we would like to assume that there are both event and self-triggered safety messages in our vCPS use case. Therefore, our distributed monitoring solution will be designed as a hybrid solution, which is discussed in detail below.

Our case is considered to be distributed, where each agent updates its own control input at event times it decided based on the self-triggered event within the particular controller or node, as depicted in Fig. 11. Notably, we assume that each vCPS is provided with an agent for such a distributed event monitoring solution. In the self-triggered setup, the next time t_j^{k+1} at which the control law is updated is predetermined at the previous event time t_j^k (k is the sampling instant of the MPC). This does not require any state measurement between the control updates. Advanced machine learning could also make the triggered prediction more accurate but this is out of the scope of this work. The self-triggered algorithm is summarised in Algorithm 3 (see Fig. 12), which we explain in the following text. Initialise time $t_{\max}$ upper bound, that is the limit of event monitoring after initial beaconing synchronisation. The algorithm is distributed in the sense that all $j = 1, \ldots, N$ agents are being checked through the system operation. Given that an event is triggered at agent j, the iteration k_j is updated at time t_j and the condition of not exceeding the update value ξ_j is checked. The algorithm is running assuming the follower and the leader follows, i.e. starting from the last one, it ends up to the first one in a platooning type of the use case. Thus, the neighbour agent, i.e. leader, is also being updated and checked. Finally, the control input $u(k_j)$ is updated and the event triggered is concatenated to the overall packet.

We now focus on the self-triggered scheme to demonstrate the triggering situations at the MPC depicted in the control layer diagram (Fig. 10). We consider, in particular, the input and output of the MPC on the host (follower) car and depict the input and control progress over time calculating the triggering number of events from the safety point of view. It should be noted that in the self-triggered algorithm, continuous monitoring of measurement errors is not needed any more and thus, the communication load can be reduced. This is the main advantage of the self-triggered scheme. However, the number of triggering times determined by the self-triggered scheme will be, in general, more than that of the event-triggered scheme [36]. The goal is to monitor the control of MPC updates through the distributed event monitoring, as explained in details above. To demonstrate the triggering situations of the MPC, we further present a figure describing the control inputs i.e. both input and output for the MPC with the self-triggered control schemes applied. Notably, the use case example below is for a pair of leader–follower with two agents, where the events are considered to be the MPC of the follower. For further investigation on large distributed agents embedded to vCPS, a future work should be considered.

Fig. 11 depicts the control input and output of the MPC component over time. Table 2 lists the number of triggering times that can be found in Fig. 11. It depicts the input that is actually the velocity in m/s^2 and the output that is the acceleration in m/s. The input is actually coming from the *position state calculator* to the *MPC* and the output is going back to the position state through the *F-wheeled controller*. In order to calculate the triggered events, we concentrate on the output that highlights the impact from the safety point of view. We assume a delay of 10 ms, where the MPC operates in a sampling period of 0.1 s. Fig. 11 depicts the considered triggered regions, i.e. the regions where we assume the output is not considered to be close to the original one, i.e. without delays. Notably, such an example is also related to the command delay and drop events of the mapping layer in Fig. 11. The commands are carried out at the *CTRL-C-3* that is actually the MPC module. It is observed that the input delay at the MPC does not affect significantly the MPC output since it does not cross the triggered event region. A delay, though exists, can be compensated with some estimation techniques. Events are triggered significantly in the case of output delay and input output.

Table 2 Number of triggered acceleration events

Input/output	Input	Output	Original
30	0	10	0

The self-triggered algorithm is about the distributed safety monitoring without continuous state monitoring of the control system. As a result, the communication load among the distributed control system is getting lower. To this end, Table 2 shows the reduction on the communication overhead using the self-triggered rather than an event-triggered solution. Table 2 lists the number of triggered acceleration events at the MPC. The input/output case

number is 30 events, where the output numbers 10. As mentioned above the input delay does not affect the system in terms of triggered events.

6 Conclusions

In this work, we introduced a DCP for cooperative vCPS that consists of three layers such as cooperative awareness, M2M communications and distributed control. Next, we discussed about the distributed control using the model predictive control for such a mobile robot application. Afterwards, the communication protocol specification is considered taking into account the short packet design requirements. A problem formulation is defined that could guarantee the ultra-reliable distributed control satisfying the latency constraint and the rate maximisation. Our distributed solution relies on the Hungarian method for low-complexity implementations. A use case with low and high acceleration of Join/Leave events is simulated, which shows the impact of reliable short packet transmission. Finally, we introduce the safety events for full cooperative awareness among different vCPS. The safety framework is specified defining the control layer diagram and its mapping to events. A distributed event monitoring algorithm is deployed to notify the MPC of the follower. Simulation results highlight the number of triggered events to keep the leader/follower structure safe under certain conditions. Future work is considered to be the evaluation of the proposed plane under a large amount of vCPS with different types of hazardous events encapsulated to the cooperation awareness layer. Another potential consideration in collaboration with the industry is related to standards for emerging connected CPSs.

7 Acknowledgments

The research leading to these results has been performed in the SafeCOP project that received funding from the ECSEL Joint Undertaking under grant agreement no. 692529 and from the National funding.

8 References

[1] Jia, D., Lu, K., Wang, J., *et al.*: 'A survey on platoon-based vehicular cyber-physical systems', *IEEE Commun. Surv. Tutor.*, 2016, **18**, (1), pp. 263–284

[2] Pop, P., Scholle, D., Hansson, H., *et al.*: 'The SafeCOP ECSEL project: safe cooperating cyber-physical systems using wireless communication'. 2016 Euromicro Conf. on Digital System Design (DSD), Limassol, Cyprus, September 2016

[3] Fink, J., Ribeiro, A., Kumar, V.: 'Robust control for mobility and wireless communication in cyber–physical systems with application to robot teams', *Proc. IEEE*, 2012, **100**, (1), pp. 164–178

[4] Milanes, V., Shladover, S.E., Spring, J., *et al.*: 'Cooperative adaptive cruise control in real traffic situations', *IEEE Trans. Intell. Transp. Syst.*, 2014, **15**, (1), pp. 296–305

[5] Yang, A., Naeem, W., Fei, M.: 'Decentralised formation control and stability analysis for multi-vehicle cooperative manoeuvre', *IEEE/CAA J. Autom. Sin.*, 2014, **1**, (1), pp. 92–100

[6] Liu, Z., Chen, W., Lu, J., *et al.*: 'Formation control of mobile robots using distributed controller with sampled-data and communication delays', *IEEE Trans. Control Syst. Technol.*, 2106, **24**, (6), pp. 2125–2132

[7] Briñón-Arranz, L., Seuret, A., Canudas-de-Wit, C.: 'Cooperative control design for time-varying formations of multi-agent systems', *IEEE Trans. Autom. Control*, 2014, **59**, (8), pp. 2283–2288

[8] Hausman, K., Muller, J., Hariharan, A., *et al.*: '*Cooperative control for target tracking with onboard sensing, experimental robotics*' (Springer Publ., New York, USA, 2015), pp. 879–892

[9] Khaitan, S.K., McCalley, J.D.: 'Design techniques and applications of cyberphysical systems: a survey', *IEEE Syst. J.*, 2015, **9**, (2), pp. 350–365

[10] Hu, F., Lu, Y., Vasilakos, A.V., *et al.*: 'Robust cyber–physical systems: concept, models, and implementation', *Future Gener. Comput. Syst.*, 2016, **56**, pp. 449–475

[11] di Bernardo, M., Salvi, A., Santini, S.: 'Distributed consensus strategy for platooning of vehicles in the presence of time-varying heterogeneous communication delays', *IEEE Trans. Intell. Transp. Syst.*, 2015, **16**, (1), pp. 102–112

[12] Amoozadeha, M., Dengb, H., Chuaha, C.-N., *et al.*: 'Platoon management with cooperative adaptive cruise control enabled by VANET', *Veh. Commun.*, 2015, **2**, (2), pp. 110–123

[13] Segata, M., Bloessl, B., Joerer, S., *et al.*: 'Supporting platooning maneuvers through IVC: an initial protocol analysis for the JOIN maneuver'. 2014 11th Annual Conf. on Wireless On-demand Network Systems and Services (WONS), Obergurgl, Austria, April 2014

[14] Segata, M., Bloessl, B., Joerer, S., *et al.*: 'Toward communication strategies for platooning: simulative and experimental evaluation', *IEEE Trans. Veh. Technol.*, 2015, **64**, (12), pp. 5411–5423

[15] Shen, B., Zhou, X., Kim, M.: 'Mixed scheduling with heterogeneous delay constraints in cyber-physical systems', *Future Gener. Comput. Syst.*, 2016, **61**, pp. 108–117

[16] Korukonda, M.P., Mishra, S.R., Shukla, A., *et al.*: 'Handling multi-parametric variations in distributed control of cyber-physical energy systems through optimal communication design', *IET Cyber-Phys. Syst., Theory Appl.*, 2017, **2**, (2), pp. 90–100

[17] Wan, J., Zhang, D., Zhao, S., *et al.*: 'Context-aware vehicular cyber-physical systems with cloud support: architecture, challenges, and solutions', *IEEE Commun. Mag.*, 2014, **52**, (8), pp. 106–113

[18] Trappey, A.M.J., Trappey, C.V., Govindarajan, U.H., *et al.*: 'A review of technology standards and patent portfolios for enabling cyber-physical systems in advanced manufacturing', *Optim. Emerging Wirel. Netw. Ind. 4.0 IEEE Access*, 2016, **4**, pp. 7356–7382

[19] Gazis, V.: 'A survey of standards for machine-to-machine and the internet of things', *IEEE Commun. Surv. Tutor.*, 2017, **19**, (1), pp. 482–511

[20] Araujo, J., Mazo, M., Anta, A., *et al.*: 'System architectures, protocols and algorithms for aperiodic wireless control systems', *IEEE Trans. Ind. Inf.*, 2014, **10**, (1), pp. 175–184

[21] Xiao, H., Li, Z., Chen, C.L.P.: 'Formation control of leader follower mobile robots systems using model predictive control based on neural-dynamic optimization', *IEEE Trans. Ind. Electron.*, 2016, **63**, (9), pp. 5752–5762

[22] Liu, A., Bai, L.: 'Distributed model predictive control for wide area measurement power systems under malicious attacks', *IET Cyber-Phys. Syst., Theory Appl.*, 2018, **3**, (3), pp. 111–118

[23] Cimini, G., Bemporad, A.: 'Exact complexity certification of active-set methods for quadratic programming', *IEEE Trans. Autom. Control*, 2017, **PP**, (99), pp. 1–1

[24] Model Predictive Control Toolbox, The MathWorks, Inc., 2017. Available at https://se.mathworks.com/help/mpc/

[25] Bageshwar, V.L., Garrard, W.L., Rajamani, R.: 'Model predictive control of transitional maneuvers for adaptive cruise control vehicles', *IEEE Trans. Veh. Technol.*, 2004, **53**, (5), pp. 1573–1585

[26] Kim, S.-L., Burgard, W., Kim, D.: 'Wireless communications in networked robotics', *IEEE Wirel. Commun.*, 2009, **16**, (1), pp. 4–5

[27] Durisi, G., Koch, T., Popovski, P.: 'Toward massive, ultrareliable, and low-latency wireless communication with short packets', *IEEE Proc.*, 2016, **104**, (9), pp. 1711–1726

[28] Lee, J.-M., Woo, M.-S., Min, S.-G.: 'Performance analysis of WAVE control channels for public safety services in VANETs', *Int. J. Comput. Commun. Eng.*, 2013, **2**, (5), pp. 563–570

[29] Hadded, M., Muhlethaler, P., Laouiti, A., *et al.*: 'TDMA-based MAC protocols for vehicular *ad hoc* networks: a survey, qualitative analysis, and open research issues', *IEEE Commun. Surv. Tutor.*, 2015, **17**, (4), pp. 2461–2492

[30] Gu, J., Bae, S.J., Hasan, S.F., *et al.*: 'Heuristic algorithm for proportional fair scheduling in D2D-cellular systems', *IEEE Trans. Wirel. Commun.*, 2016, **15**, (1), pp. 769–780

[31] Machin, M., Guiochet, J., Waeselynck, H., *et al.*: 'SMOF: a safety monitoring framework for autonomous systems', *IEEE Trans. Syst. Man Cybern., Syst.*, 2018, **48**, (5), pp. 702–715

[32] Friedberg, I., MxLaughlin, K., Smith, P., *et al.*: 'STPA-SafeSec: safety and security analysis for cyber physical systems', *J. Inf. Secur. Appl.*, 2017, **34**, part 2, pp. 183–196

[33] Nourian, A., Madnick, S.: 'A systems theoretic approach to the security threats in cyber physical systems applied to stuxnet', *IEEE Trans. Dependable Secur. Comput.*, 2018, **15**, (1), pp. 2–15

[34] Liu, J., Wan, J., Wang, Q., *et al.*: 'A time-recordable cross-layer communication protocol for the positioning of vehicular cyber-physical systems', *Future Gener. Comput. Syst.*, 2016, **56**, pp. 438–448

[35] Lyu, F., Zhu, H., Zhou, H., *et al.*: 'SS-MAC: a novel time slot-sharing MAC for safety messages broadcasting in VANETs', *IEEE Trans. Veh. Technol.*, 2018, **67**, (4), pp. 3586–3597

[36] Hu, W., Liu, L., Feng, G.: 'Consensus of linear of multi-agent systems by distributed event-triggered strategy', *IEEE Trans. Cybern.*, 2016, **46**, (1), pp. 148–157