technische universität
dortmund

Optimal Task Phasing for End-To-End Latency in Harmonic and Semi-Harmonic Automotive Systems

Mario Günzel and Matthias Becker
TU Dortmund University, Department of Computer Science, Germany
KTH Royal Institute of Technology, Sweden

Citation: TBD

BibTeX:

```
@inproceedings{GuenzelBecker25RTAS,
  author={Mario Günzel and Matthias Becker},
  booktitle={{IEEE} Real-Time and Embedded Technology and Applications Symposium ({RTAS})},
  title={Optimal Task Phasing for End-To-End Latency in Harmonic and Semi-Harmonic Automotive Systems},
  year={2025},
}
```


computer
science 12

Optimal Task Phasing for End-To-End Latency in Harmonic and Semi-Harmonic Automotive Systems

Mario Günzel
TU Dortmund University, Germany
mario.guenzel@tu-dortmund.de

Matthias Becker
KTH Royal Institute of Technology, Sweden
mabecker@kth.se

***Abstract*—In the context of automotive systems, the end-to-end latency of a sequence of tasks (a so-called cause-effect chain) is a common metric to ensure correct timing behavior. To control the end-to-end latency, proper task configuration is crucial. While the literature considers the configuration of task periods, optimization of task phases to minimize the end-to-end latency is only sparsely discussed.**

In this work, we examine the configuration of task phases to optimize the end-to-end latency of a cause-effect chain that communicates under the Logical Execution Time (LET) paradigm. To that end, we develop a strategy for cause-effect chains with harmonic or semi-harmonic periods, which are very common in industrial applications. We prove that our strategy is optimal in the sense that it minimizes the end-to-end latency. Furthermore, our evaluation based on a real-world use-case and on synthetic automotive benchmarks shows that optimizing task phases can reduce end-to-end latencies significantly. Our approach takes at most $49\,\mu s$ to find the optimal phasing and compute the end-to-end latency for cause-effect chains with 50 tasks, reducing the end-to-end latency by 28% in median.

I. Introduction

To address the growing complexity of embedded systems, different functions are divided across several communicating tasks. Each sequence of tasks that is involved in performing a functionality is called a *cause-effect chain*. Cause-effect chains typically start at sensor tasks, followed by processing tasks and finally terminate at actuator tasks. For a correct functionality, data must be propagated through the cause-effect chain.

The timing behavior of a cause-effect chain is described by its end-to-end latency. That is, the maximum time from a cause to its corresponding effect must be guaranteed to be below a specific bound. In the literature, two metrics are usually distinguished when analyzing the end-to-end latency: Maximum Reaction Time (MRT) and Maximum Data Age (MDA). Multiple analysis approaches exist in the literature [3]–[5], [7], [8], [12]–[15], [17], [24], [25], [32], [34]. Recently, it has been shown that both MRT and MDA are equivalent [18].

End-to-end timing requirements on cause-effect chains are the primary type of timing requirement for a majority of industrial applications [2]. During the design, it must be ensured that the final system results in an end-to-end latency of each cause-effect chain that does not exceed its latency bound. In contrast to the scheduler decision and general architecture, the task configuration (i.e., the period and phase) can often be modified more easily. Therefore, the task configuration is the typical target for system optimization.

In the literature two communication mechanisms are predominantly studied: *implicit communication mechanism* and communication under the *Logical Execution Time (LET)* paradigm. LET has become increasingly interesting for industrial domains, such as the automotive domain [1], [19]. The reason is that LET, proposed as part of the GIOTTO language [20], [22] to abstract the communication from the timing behavior of tasks, leads to deterministic time and data flow through cause-effect chains that consist of LET tasks, and therefore simplifies the integration phase and provides robustness to changes of application functionality and extensions.

For cause-effect chains under the LET model, Paladino et al. [29] recently proposed an approach to optimize task priorities to minimize end-to-end latency. While often the assumption is that all tasks are released synchronously, the configuration of task phases represents a target for optimization. So far, only Martinez et al [28] investigate the selection of task phases. Their proposed heuristic identifies all non-indentical combinations of task phases which are subsequently analyzed to find the setting that leads to the minimal end-to-end latency. While their optimization approaches allow task periods to be chosen freely, in many industrial applications, periods are chosen to be *semi-harmonic*, i.e., most periods are integer multiples of each other. For instance, in the automotive domain, task periods are often assigned from a fixed set of semi-harmonic periods, such as $\{1, 2, 5, 10, 20, 50, 100, 200, 1000\}$, as described by Kramer et al. [26].

In this work, we develop a strategy to select task phases for semi-harmonic systems, which is optimal in the sense that it minimizes the end-to-end latency of a cause-effect chain. To that end, we identify and examine two types of semi-harmonic systems which cover systems described in [26]: (i) In *max-harmonic* systems, the largest period is divisible by all other periods.[1] (ii) In $(2, k)$*-max-harmonic* systems, all periods divide one of the two largest periods, and the hyperperiod is 2 times the largest period and k times the second largest period. To find the optimal phasing, this work is the first that exploits *partitioned job chains* analytically after their initial exploration in [18]. Specifically, we provide the following **contributions**:

- In Section VI, we propose an optimal phasing, to minimize the end-to-end latency of *max-harmonic systems*.

[1]This also covers fully harmonic systems, where all periods are divisible by all smaller periods.

- In Section VII, we extend our results to $(2, k)$-*max-harmonic systems*, to make them applicable to cause-effect chains with *semi-harmonic automotive periods* such as those presented in [26].
- Section VIII, demonstrates the benefits of the proposed phasing using an automotive case study. Furthermore, quantitative evaluations are preformed using synthetic cause-effect chains with automotive periods as well as $(2, k)$-max-harmonic periods. For cause-effect chains with automotive periods and a length of 50, the optimal phasing is compute din $49\,\mu s$ and reduces the median latency by 28% compared to a synchronous release.

II. Related Work

Several communication paradigms are considered for cause-effect chains [36]. Depending on the assumed communication paradigm, different solutions have been proposed. For chains under implicit communication, Davare et al. [12] optimize the period configuration of tasks while ensuring that latency constraints are met. Schlatow et al. [34] consider a mix of periodic and sporadic tasks. Their approach optimizes the task phase, while the original deadlines remain unchanged. Klaus et al. [23] address the execution of multiple, interconnected cause-effect chains on multicore platforms. They leverage job-level dependencies [4] to meet the chains' latency constraints, while minimizing the resulting synchronization overheads. The approach by Sinha et al. [35] admits new cause-effect chains to a system by tuning task periods and the number of input samples that are consumed by each task instance such that latency constraints are met. The configuration of synchronous programs under end-to-end latency constraints is described by Bourke et al. [11]. Their approach is based on a language to express execution rates and rate transitions. An ILP then assigns tasks to scheduling slots and sorts them within each slot, taking different rate transition operators into account. This is done such that all chains' latency constraints are met.

For LET, implementations of the paradigm has been demonstrated for Real-Time Operating Systems [9] and POSIX-based systems [6]. And several works address optimizations for an efficient LET implementation [10], [30], [31]. Resmerita et al. [33] discuss how to convert legacy applications to the LET model. And Gemlau et al. [16] extend the LET paradigm to distributed systems with *System-Level LET (SL-LET)*. Both, LET and SL-LET are also part of the AUTOSAR standard [1]. The configuration of cause-effect chains under LET has been addressed in several works. Wang et al. [37] address the *flexible LET (fLET)* model. In contrast to the LET model, under fLET the LET interval can be shrunk, which is then used to improving the end-to-end latency. Maia et al. [27] consider scheduling information which is used to shorten LET intervals to reduce the end-to-end latency. Different communication paradigms for cause-effect chains, including LET, are compared in [36] and the authors propose a priority assignment that reduces end-to-end latency. Xu et al. [38] optimize the mapping of AUTOSAR runnables to LET tasks and tasks to cores to meet the chains' latency constraints, using reinforcement learning. Bini et al. [8] present an analysis for chains of LET tasks that is based on algebraic rings. They show how two communicating LET tasks can be modelled as a single LET task and optimize cause-effect chains by inserting additional copy tasks. Paladigno et al. [29] consider a generalized LET model where output data may be published at or after the tasks response time. For this setting constant latency chains are build which enables the exploration of the design space to assign task periods such that end-to-end latency is minimized. Most related to the problem we address in this paper is the work by Martinez et al. [28]. They develop an approach to compute the end-to-end latency of a cause-effect chain and propose a heuristic for the assignment of task phases. The heuristic is applicable to chains with arbitrary periods and determines the set of non-equivalent phase assignments which are then individually analyzed to determine the assignment that results in the minimal end-to-end latency. In contrast, our work presents a suggested task phasing for cause-effect chains with max-harmonic and $(2, k)$-max-harmonic periods, that is proven to be optimal. Hence no exploration of different configurations is required. For this configuration we further show how to efficiently compute the end-to-end latency.

Table I
Notation.

Symbol	Description
$\tau = (C_\tau, T_\tau, \phi_\tau) \in \mathbb{T}$	Tasks
$H(\mathbb{T}')$	Hyperperiod of a set $\mathbb{T}'$ of tasks
$\mathrm{re}(\tau(m))$, $\mathrm{we}(\tau(m))$	Read-event and write-event of job $\tau(m)$
E	A cause-effect chain with tasks $\mathbb{T}_E \subseteq \mathbb{T}$
W_i	Warm-up of task τ_i
$\mathrm{Lat}(E)$	End-to-end latency of cause-effect chain E
$\vec{c}_m(E)$, $\overleftarrow{c}_m(E)$	immediate forward and backward job chains of E
$pc_m^p = (\overleftarrow{c}_m^p, \vec{c}_{m+1}^p)$	m-th p-partitioned job chain

III. System Model

In embedded systems, several tasks are deployed that each fulfills a different dedicated purpose. The set of all tasks is denoted as $\mathbb{T}$. To allow the embedded system to fulfill complex assignments, tasks have to interact with each other. That is achieved by data exchange between tasks. The notation of this work is summarized in Table I.

Task Model: Each task $\tau \in \mathbb{T}$ can be described by a tuple $(C_\tau, T_\tau, \phi_\tau) \in \mathbb{R}^3$. More specifically, $C_\tau > 0$ is the worst-case execution time (WCET) of τ, $T_\tau > 0$ is the task period, and ϕ_τ is the task phase. The task recurrently releases jobs, denoted by $\tau(m),\ m \in \mathbb{N}{=}\{0, 1, 2, \ldots\}$ according to its description. That is, $\tau(0)$ is released at time ϕ_τ, and subsequent jobs are released every T_τ time units, i.e., $\tau(m)$ is released at time $\phi_\tau + m \cdot T_\tau$. Each job $\tau(m)$ executes for at most C_τ time units. We consider implicit-deadline tasks where the deadline of each job is at the release of the subsequent job, i.e., the deadline of $\tau(m)$ is at $\phi_\tau + (m + 1) \cdot T_\tau$. The hyperperiod of a set $\mathbb{T}' \subseteq \mathbb{T}$ of tasks is the least common multiple of the periods $H(\mathbb{T}') = \mathrm{LCM}(\{T_\tau \,|\, \tau \in \mathbb{T}'\})$.

Task sets are *harmonic* if all task periods are integer multiples of one another. That is, for all $\tau, \tau' \in \mathbb{T}$, we have

$$\frac{T_\tau}{T_{\tau'}} \in \mathbb{N} \qquad \text{or} \qquad \frac{T_{\tau'}}{T_\tau} \in \mathbb{N}. \tag{1}$$

Harmonic task sets have the benefit that their release pattern is highly repetitive. As a result, the data propagation through these tasks is easier to analyze and to control. For the results of this work, weaker forms of harmonic are already sufficient. Specifically, in Section VI, tasks of the cause-effect chain $\mathbb{T}_E \subseteq \mathbb{T}$ are *max-harmonic*:

Definition 1 (Max-Harmonic)**.** A task set $\mathbb{T}' \subseteq \mathbb{T}$ is called *max-harmonic* if all periods divide the largest period, i.e.,

$$\frac{\max_{\tau' \in \mathbb{T}'} T_{\tau'}}{T_\tau} \in \mathbb{N} \tag{2}$$

for all $\tau \in \mathbb{T}'$.

While max-harmonic tasks are essential ingredients in many industry systems, they are not sufficient to cover for example the automotive benchmark from [26]. For instance, the tasks may have periods 1, 2, and 5 which are not max-harmonic. Therefore, we extend our results to $(2,k)$-*max-harmonic* tasks:

Definition 2 ($(2,k)$-Max-Harmonic)**.** Given a set $\mathbb{T}' \subseteq \mathbb{T}$ with the two largest periods[2]:

$$T_{\max,1}(\mathbb{T}') := \max\{T_\tau \,|\, \tau \in \mathbb{T}'\} \tag{3}$$

$$T_{\max,2}(\mathbb{T}') := \max\{T_\tau \,|\, \tau \in \mathbb{T}', T_\tau \neq T_{\max,1}(\mathbb{T}')\} \tag{4}$$

The set $\mathbb{T}'$ is called $(2,k)$-*max-harmonic* if the following conditions hold:

- The two subsets $\{\tau \in \mathbb{T}' \,|\, T_\tau \neq T_{\max,1}(\mathbb{T}')\} \subset \mathbb{T}'$ and $\{\tau \in \mathbb{T}' \,|\, T_\tau \neq T_{\max,2}(\mathbb{T}')\} \subset \mathbb{T}'$ are max-harmonic.
- The hyperperiod $H(\mathbb{T}')$ of the given task set $\mathbb{T}'$ is $H(\mathbb{T}') = 2 \cdot T_{\max,1}(\mathbb{T}') = k \cdot T_{\max,2}(\mathbb{T}')$.

As an example, tasks with periods $\{1,2,5\}$ or $\{10,20,50\}$ are $(2,5)$-max-harmonic. Furthermore, task sets derived by the automotive benchmark [26] are always either max-harmonic or $(2,5)$-max-harmonic, and therefore studying these two types is sufficient to apply our results to such task sets.

Task communication: To transmit data between tasks, they need to communicate. We assume that the communication can be modeled via shared resources. That is, a task writes data to a shared resource (overwriting previous data), and the successor task reads the data from that shared resource. In this work, we focus on the communication model of the Logical Execution Time (LET) [22] mechanism, where each job reads data at its release and writes data at its deadline. More specifically, the read-event of $\tau(m)$ is at time

$$\mathrm{re}(\tau(m)) := \phi_\tau + m \cdot T_\tau, \tag{5}$$

and the write-event of $\tau(m)$ is at time

$$\mathrm{we}(\tau(m)) := \phi_\tau + (m+1) \cdot T_\tau. \tag{6}$$

[2]If there is only one period, the task set is max-harmonic. For a task set to be $(2,k)$-max-harmonic at least two different periods are required.

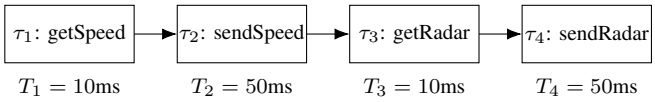


Figure 1. Example cause-effect chain $E = (\tau_1 \to \tau_2 \to \tau_3 \to \tau_4)$ of an Autonomous Emergency Braking System (AEBS), with annotated periods.

We assume that communication overheads are negligible, which is realized by existing LET implementations [6], [9].

Scheduling Model: The results of this work are versatile in the sense that we do not rely on a specific scheduling model. For example, typical Fixed-Priority (FP) schedulers like Rate-Monotonic (RM) and Deadline-Monotonic (DM), or Dynamic-Priority (DP) schedulers like Earliest-Deadline-First (EDF) can be applied. To ensure that task communication via LET can be realized, schedulability of the task set needs to be verified. This can be done using appropriate schedulability tests for the underlying scheduling model.

IV. Cause-Effect Chains and End-to-End Latencies

To achieve complex functionalities, usually several tasks have to cooperate. A sequence of cooperating tasks is called a cause-effect chain

$$E = (\tau_1 \to \cdots \to \tau_n) \tag{7}$$

with $n \in \mathbb{N}_{\geq 1}$. Specifically, data that is written by τ_i is read by τ_{i+1}, for $i = 1, \ldots, n-1$. The underlying functionality is achieved when data has been propagated through all n tasks. For this work, we denote by

$$\mathbb{T}_E := \{\tau_1, \ldots, \tau_n\} \subseteq \mathbb{T} \tag{8}$$

the tasks of $\mathbb{T}$ that are part of the cause-effect chain. Furthermore, we assume that $\tau_i \neq \tau_j$ for all $i \neq j$. Figure 1 shows a cause-effect chain that is part of the Autonomous Emergency Braking System (AEBS) described in [21]. Sensor values are sampled with a period of $10\,\mathrm{ms}$. Those values are subsequently processed by a filter task with a period of $50\,\mathrm{ms}$ that filters outliers and smooths the sensor noise, before the filtered values are written.

In the literature, two metrics for the end-to-end latency are usually considered:

1) Maximum Reaction Time: *How long does it take until an external activity is processed?*
2) Maximum Data Age: *How old is data used in an actuation?*

Recently, it was shown that both metrics are equivalent [18]. Therefore, we do not distinguish those metrics and use the term *end-to-end latency* ($\mathrm{Lat}(E)$) universally. Further, it was shown in [18] that *partitioned job chains* can be used to define the end-to-end latency. Partitioned job chains rely on immediate forward and immediate backward job chains [13], which are used to track data propagation and data origin through the system, respectively.

Definition 3 (Immediate forward job chain)**.** Let $m \in \mathbb{N}$ and $E = (\tau_1 \to \cdots \to \tau_n)$ be a cause-effect chain. The m-th *immediate forward job chain* of E is a sequence of jobs

$$\vec{c}_m(E) = (\tau_1(j_1) \to \cdots \to \tau_n(j_n)) \tag{9}$$

such that $j_1 = m$, and for all $i = 1, \ldots, n-1$ the job $\tau_{i+1}(j_{i+1})$ is the *earliest* job of task τ_{i+1} that reads data after it is written by $\tau_i(j_i)$, i.e., we have $j_{i+1} = \min\{\xi \in \mathbb{N} \,|\, \mathrm{we}(\tau_i(j_i)) \leq \mathrm{re}(\tau_{i+1}(\xi))\} = \min\left\{\xi \in \mathbb{N} \,\middle|\, \phi_{\tau_i} + (j_i + 1) \cdot T_{\tau_i} \leq \phi_{\tau_{i+1}} + \xi \cdot T_{\tau_{i+1}}\right\}$.

Definition 4 (Immediate backward job chain)**.** Let $m \in \mathbb{N}$ and $E = (\tau_1 \to \cdots \to \tau_n)$ be a cause-effect chain. The m-th *immediate backward job chain* of E is a sequence of jobs

$$\overleftarrow{c}_m(E) = (\tau_1(j_1) \to \cdots \to \tau_n(j_n)) \tag{10}$$

such that $j_n = m$, and for all $i = 2, \ldots, n$ the job $\tau_{i-1}(j_{i-1})$ is the *latest* job of τ_{i-1} that writes data before it is read by $\tau_i(j_i)$, i.e., $j_{i-1} = \max\{\xi \in \mathbb{N} \,|\, \mathrm{we}(\tau_{i-1}(\xi)) \leq \mathrm{re}(\tau_i(j_i))\} = \max\left\{\xi \in \mathbb{N} \,\middle|\, \phi_{\tau_{i-1}} + (\xi + 1) \cdot T_{\tau_{i-1}} \leq \phi_{\tau_i} + j_i \cdot T_{\tau_i}\right\}$.

We note that for some small m the immediate backward job chain may not always exist in case $\{\xi \in \mathbb{N} \,|\, \mathrm{we}(\tau_{i-1}(\xi)) \leq \mathrm{re}(\tau_i(j_i))\} = \emptyset$. Intuitively, this means that the job $\tau_i(j_i)$ has no data to process because task τ_{i-1} has not been executed yet. Such a scenario can only occur during the initial startup of the system. However, typical analyses are more interested in the system behavior during runtime when the system is already properly warmed up. We follow [18] to determine the warm-up phase of each job.

Definition 5 (Warm-up)**.** Consider the cause-effect chain $E = (\tau_1 \to \cdots \to \tau_n)$. Let $W \in \mathbb{N}$ be the smallest integer such that $\overleftarrow{c}_W(E) = (\tau_1(W_1) \to \cdots \to \tau_n(W_n))$ exists, i.e., $\overleftarrow{c}_W(E)$ is the *first* immediate backward job chain. We say that task τ_i is *warmed up* with respect to E at job $\tau_i(W_i)$.

Intuitively, a task is warmed up with the first job that generates data that propagates to the end of the cause-effect chain without being overwritten. As shown in [18], to defined the end-to-end latency we can choose any task τ_p and track both data origin and data progess starting from τ_p. This is formalized using partitioned job chains.

Definition 6 (Partitioned job chain.)**.** Let $m \in \mathbb{N}$, $E = (\tau_1 \to \cdots \to \tau_n)$, and $p \in \{1, \ldots, n\}$. The m-th p-partitioned job chain of E is a tuple

$$pc_m^p = (\overleftarrow{c}_m^p, \vec{c}_{m+1}^p) \tag{11}$$

of an immediate backward and an immediate forward job chain. More specifically, $\overleftarrow{c}_m^p$ is the m-th immediate backward job chain of $(\tau_1 \to \cdots \to \tau_p)$, i.e.,

$$\overleftarrow{c}_m^p := \overleftarrow{c}_m((\tau_1 \to \cdots \to \tau_p)) \tag{12}$$

and $\vec{c}_{m+1}^p$ is the $(m+1)$-th immediate forward job chain of $(\tau_p \to \cdots \to \tau_n)$, i.e.,

$$\vec{c}_{m+1}^p := \vec{c}_{m+1}((\tau_p \to \cdots \to \tau_n)). \tag{13}$$

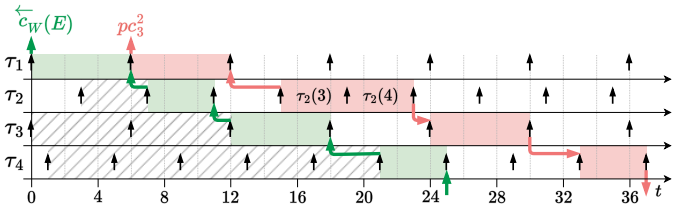


Figure 2. A cause-effect chain $E = (\tau_1 \to \tau_2 \to \tau_3 \to \tau_4)$ with LET communication semantics. Job releases are marked with black upward arrows. The partitioned job chain $pc_3^2 = (\overleftarrow{c}_3^2, \vec{c}_4^2)$ is marked in red. Red backward arrows mark $\overleftarrow{c}_3^2$ and red forward arrows mark $\vec{c}_4^2$. The first immediate backward job chain that exists, $\overleftarrow{c}_W(E)$, is shown in green, and the warm-up phase is indicated by jobs with hatched filling.

We note that the $pc_m^p = (\overleftarrow{c}_m^p, \vec{c}_{m+1}^p)$ exists if and only if its first entry $\overleftarrow{c}_m^p$ exists. Hence, the pc_m^p exists for all $m \geq W_p$. The length of $pc_m^p = (\overleftarrow{c}_m^p, \vec{c}_{m+1}^p)$ with $\overleftarrow{c}_m^p = (\tau_1(j_1) \to \cdots \to \tau_p(j_p))$ and $\vec{c}_{m+1}^p = (\tau_p(j_p') \to \cdots \to \tau_n(j_n'))$ is defined as

$$\ell(pc_m^p) := \mathrm{we}(\tau_n(j_n')) - \mathrm{re}(\tau_1(j_1)) \tag{14}$$

$$= (j_n' + 1) \cdot T_{\tau_n} + \phi_{\tau_n} - (j_1 \cdot T_{\tau_1} + \phi_{\tau_1}). \tag{15}$$

An example of a partitioned job chain of tasks with periods $T_{\tau_1} = T_{\tau_3} = 6$ and $T_{\tau_2} = T_{\tau_4} = 4$ is shown in Figure 2.

Definition 7 (End-to-end latency)**.** The end-to-end latency of a cause-effect chain $E = (\tau_1 \to \cdots \to \tau_n)$ is the maximal length of p-partitioned job chains after the warm-up for any $p \in \{1, \ldots, n\}$. That is,

$$\mathrm{Lat}(E) = \sup_{m \geq W_p} \ell(pc_m^p). \tag{16}$$

It is shown in [18] that the end-to-end latency is independent of the choice of p, i.e., any $p \in \{1, \ldots, n\}$ can be chosen.

Furthermore, for periodic tasks under LET, the read- and write-events repeat every hyperperiod of $\mathbb{T}_E$, i.e., every $H(\mathbb{T}_E) = \mathrm{LCM}(\{T_{\tau_1}, \ldots, T_{\tau_n}\})$. Therefore, the partitioned job chains repeat every hyperperiod and only finitely many partitioned job chains need to be considered for the computation of the end-to-end latency.

Lemma 8. *Let $\mu(p) := \frac{H(\mathbb{T}_E)}{T_{\tau_p}}$ be the number of periods of τ_p that fit into one hyperperiod of $\mathbb{T}_E$. We have*

$$\ell(pc_m^p) \geq \ell\left(pc_{m+\mu(p)}^p\right) \tag{17}$$

for all $m \geq W_p$. In particular,

$$\mathrm{Lat}(E) = \max\left(\ell\left(pc_{W_p}^p\right), \ldots, \ell\left(pc_{W_p+\mu(p)-1}^p\right)\right) \tag{18}$$

holds.

Proof. We first prove (17) and then derive (18) from that.

Proof of (17)**:** We consider partitioned job chains $pc_m^p = (\overleftarrow{c}_m^p, \vec{c}_{m+1}^p)$ and $pc_{m+\mu(p)}^p = (\overleftarrow{c}_{m+\mu(p)}^p, \vec{c}_{m+\mu(p)+1}^p)$ with $m \geq W_p$. Furthermore, we denote the jobs of $\overleftarrow{c}_m^p$ and $\vec{c}_{m+1}^p$ as:

$$\overleftarrow{c}_m^p = (\tau_1(j_1) \to \cdots \to \tau_p(j_p)) \tag{19}$$

$$\vec{c}_{m+1}^p = (\tau_p(j_p') \to \cdots \to \tau_n(j_n')) \tag{20}$$

We observe that by shifting $\overleftarrow{c}^p_m$ by $H(\mathbb{T}_E)$ time units, we obtain another job chain $jc = (\tau_1(j_1+\mu(1)) \to \cdots \to \tau_p(j_p+\mu(p)))$. Since jc has the same last job as $\overleftarrow{c}^p_{m+\mu(p)}$, and $\overleftarrow{c}^p_{m+\mu(p)}$ is immediate backward, we obtain

$$\ell(\overleftarrow{c}^p_{m+\mu(p)}) \leq \ell(jc) = \ell(\overleftarrow{c}^p_m). \tag{21}$$

Similarly, by shifting $\vec{c}^p_{m+1}$ by $H(\mathbb{T}_E)$ time units, we obtain another job chain $jc' = (\tau_p(j'_p + \mu(p)) \to \cdots \to \tau_n(j'_n + \mu(n)))$. Since jc' has the same first job as $\vec{c}^p_{m+\mu(p)+1}$, and $\vec{c}^p_{m+\mu(p)+1}$ is immediate forward, we obtain

$$\ell(\vec{c}^p_{m+\mu(p)+1}) \leq \ell(jc') = \ell(\vec{c}^p_{m+1}). \tag{22}$$

Since $\ell(pc^p_m) = \mathrm{we}(\tau_n(j'_n)) - \mathrm{re}(\tau_1(j_1))$, we have $\ell(pc^p_m) = \ell(\overleftarrow{c}^p_m) + \ell(\vec{c}^p_{m+1})$. Combining (21) and (22), we obtain $\ell(pc^p_m) = \ell(\overleftarrow{c}^p_m) + \ell(\vec{c}^p_{m+1}) \geq \ell(\overleftarrow{c}^p_{m+\mu(p)}) + \ell(\vec{c}^p_{m+\mu(p)+1}) = \ell(pc^p_{m+\mu(p)})$ This proves Equation (17).

Proof of (18)**:** This is a monotonicity argument. Especially, rewriting Equation (16) gives that $\mathrm{Lat}(E)$ equals

$$\sup_{k\in\mathbb{Z}_{\geq 0}} \max\left(\ell(pc^p_{W_p+k\cdot\mu(p)}), \ldots, \ell(pc^p_{W_p+\mu(p)-1+k\cdot\mu(p)})\right). \tag{23}$$

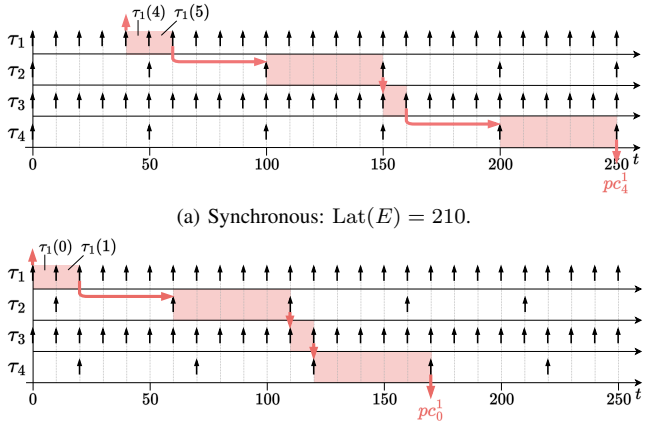
Due to the result formulated in Equation (17), the value of $\max\left(\ell(pc^p_{W_p+k\cdot\mu(p)}), \ldots, \ell(pc^p_{W_p+\mu(p)-1+k\cdot\mu(p)})\right)$ is monotonically decreasing with respect to k. Therefore, the maximal value is achieved with $k = 0$, i.e., Equation (18) holds. □

V. Motivating Example

In this section, we demonstrate the impact of task phasing on the end-to-end latency. To that end, we consider the cause-effect chain of the AEBS use-case [21] introduced in Section IV by Figure 1, i.e., $E = (\tau_1 \to \tau_2 \to \tau_3 \to \tau_4)$ with $T_{\tau_1} = 10, T_{\tau_2} = 50, T_{\tau_3} = 10, T_{\tau_4} = 50$. The typical approach of making the task set synchronous, i.e., $\phi_{\tau_1} = \phi_{\tau_2} = \phi_{\tau_3} = \phi_{\tau_4} = 0$ leads to the schedule depicted in Figure 3a.

The longest 1-partitioned job chain after the warm-up is pc^1_4, marked red. Therefore, the end-to-end latency is $\ell(pc^1_4) = 210$ for the synchronous case.

We observe in Figure 3a, that after the write-event of the first job of τ_1, the next read-event of task τ_2 is 50 time units later. Therefore, by moving the phase of τ_2 from 0 to 10, we ensure that the data is read by task τ_2 immediately after it was written by τ_1. Similarly, between task τ_3 and τ_4 a relative phasing of additional 10 time units is required that τ_4 reads the data immediately after it is written. Specifically, Figure 3b depicts the case with modified phases: $\phi_{\tau_1} = 0, \phi_{\tau_2} = 10, \phi_{\tau_3} = 0, \phi_{\tau_4} = 20$. The longest 1-partitioned job chain after the warm-up has a length of $\ell(pc^1_0) = 170$. Therefore, the end-to-end latency for the case with modified phases is $\mathrm{Lat}(E) = 170$. We conclude that a modification of the phases improves the end-to-end latency by 19%.



(a) Synchronous: $\mathrm{Lat}(E) = 210$.

(b) Optimized phasing: $\mathrm{Lat}(E) = 170$.

Figure 3. Motivating example of the AEBS use-case. Marked in red is the longest 1-partitioned job chain. Changing the phase of τ_2 from 0 to 10 and the phase of τ_4 from 0 to 20, improves the end-to-end latency by 19%.

VI. Optimal Task Phasing for Harmonic and Max-Harmonic Systems

As the motivating example demonstrates, there is a high potential of reducing the end-to-end latency by modifying the task phases. In this section, we consider a cause-effect chain $E = (\tau_1 \to \cdots \to \tau_n)$, and explore optimal task phasing that minimizes $\mathrm{Lat}(E)$ if the periods of tasks in the chain are harmonic. Actually, the results of this section work even for a larger class of task sets, in which the tasks $\mathbb{T}_E = \{\tau_1, \ldots, \tau_n\}$ of the cause-effect chain E are *max-harmonic* (cf. Definition 1). Naturally, if $\mathbb{T}$ is harmonic, then $\mathbb{T}$ and all its subsets $\mathbb{T}' \subseteq \mathbb{T}$, including $\mathbb{T}_E$, are max-harmonic.

Lemma 9. *Let τ_p be the task with the largest period of E, i.e., $T_{\tau_p} = \max_{i=1,\ldots,n} T_{\tau_i}$. If $\mathbb{T}_E$ is max-harmonic, then*

$$\mathrm{Lat}(E) = \ell(pc^p_{W_p}) \tag{24}$$

holds.

Proof. Since $\mathbb{T}_E$ is max-harmonic, we have a hyperperiod of $H(\mathbb{T}_E) = T_{\tau_p}$. Applying Lemma 8 proves the lemma. □

The previous lemma indicates that, to minimize the end-to-end latency, it is sufficient to choose phases which minimize the *first* partitioned job chain. We achieve that, by ensuring that there is no slack between write- and read-events of subsequent jobs of $pc^p_{W_p}$. In particular, we consider the following phasing.

Definition 10 (Suggested Phasing for Max-Harmonic)**.** For $E = (\tau_1 \to \cdots \to \tau_n)$, we define a task phasing

$$\phi^{MH}_{\tau_i} = \sum_{\xi=1}^{i-1} T_{\tau_\xi}, \tag{25}$$

for all $\tau_i \in \mathbb{T}_E$.

The phasing is illustrated in Figure 4. Indeed, we observe that the read- and write-event of subsequent jobs coincide for the first p-partitioned job chain, which minimizes the length of that partitioned job chain. Specifically, the end-to-end latency

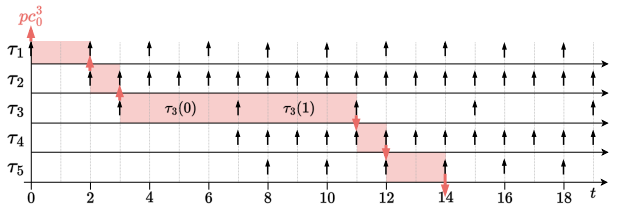


Figure 4. Phase enforcement. The first 3-partitioned job chain for $E = (\tau_1 \to \tau_2 \to \cdots \to \tau_5)$ is marked in red.

under the suggested task phasing for max-harmonic task sets is as follows.

Theorem 11 (End-to-end latency with suggested phasing)**.** *If $\mathbb{T}_E$ is max-harmonic and the task offsets are chosen as $\phi_{\tau_i} = \phi^{MH}_{\tau_i}$, then*

$$\mathrm{Lat}(E) = \sum_{i=1}^{n} T_{\tau_i} + \max_{i=1,\dots,n} T_{\tau_i}. \tag{26}$$

Proof. Let $p \in \{1,\dots,n\}$ such that τ_p is the task with the maximal period, i.e., $T_{\tau_p} = \max_{i=1,\dots,n} T_{\tau_i}$. We observe that by the chosen offsets, $\mathrm{we}(\tau_i(0)) = \mathrm{re}(\tau_{i+1}(0))$ for all $i = 1,\dots,n-1$. As a result, the immediate backward job chain $\overleftarrow{c}_0(E) = (\tau_1(0) \to \cdots \to \tau_n(0))$ exists, and $W_1 = \dots = W_n = 0$. By Lemma 9, the end-to-end latency is

$$\mathrm{Lat}(E) = \ell(pc^p_{W_p}) = \ell(pc^p_0). \tag{27}$$

In the following we investigate $pc^p_0 = (\overleftarrow{c}^p_0, \vec{c}^p_1)$, where the backward chain is denoted by $\overleftarrow{c}^p_0 = (\tau_1(j_1) \to \cdots \to \tau_p(j_p))$ and the forward chain is $\vec{c}^p_1 = (\tau_p(j'_p) \to \cdots \to \tau_n(j'_n))$.

Since $\overleftarrow{c}^p_0$ is an immediate backward job chain with last job $\tau_p(0)$ and $(\tau_1(W_1) \to \cdots \to \tau_p(W_p))$ is an immediate backward job chain with last job $\tau_p(W_p) = \tau_p(0)$ as well, they both coincide. Hence, for all $i = 1,\dots,p$, the index j_i must be the same as W_i which is 0.

Next we investigate the immediate forward job chain $\vec{c}^p_1 = (\tau_p(j'_p) \to \cdots \to \tau_n(j'_n))$. We show by induction that

$$j'_i = \frac{T_{\tau_p}}{T_{\tau_i}} \tag{28}$$

for all $i = p, p+1, \dots, n$.

Proof of Equation (28)**:**

Base case ($i = p$): By definition, $j'_i = 1 = \frac{T_{\tau_p}}{T_{\tau_i}}$. Hence, Equation (28) holds for $i = p$.

Induction step ($i \mapsto i+1$): By the induction hypothesis, $j'_i = \frac{T_{\tau_p}}{T_{\tau_i}}$. Therefore, $\mathrm{we}(\tau_i(j'_i)) = \phi_{\tau_i} + \left(\frac{T_{\tau_p}}{T_{\tau_i}} + 1\right) \cdot T_{\tau_i} = \phi_{\tau_i} + T_{\tau_p} + T_{\tau_i}$. Since $\phi_{\tau_i} = \phi^{MH}_{\tau_i} = \sum_{\xi=1}^{i-1} T_{\tau_\xi}$, we obtain $\mathrm{we}(\tau_i(j'_i)) = \sum_{\xi=1}^{i-1} T_{\tau_\xi} + T_{\tau_p} + T_{\tau_i} = \sum_{\xi=1}^{i} T_{\tau_\xi} + T_{\tau_p} = \sum_{\xi=1}^{i} T_{\tau_\xi} + \frac{T_{\tau_p}}{T_{\tau_{i+1}}} \cdot T_{\tau_{i+1}} = \phi_{\tau_{i+1}} + \frac{T_{\tau_p}}{T_{\tau_{i+1}}} \cdot T_{\tau_{i+1}}$ This is, by definition, the same as $\mathrm{re}\left(\tau_{i+1}\left(\frac{T_{\tau_p}}{T_{\tau_{i+1}}}\right)\right)$, i.e., we obtain $\mathrm{we}(\tau_i(j'_i)) = \mathrm{re}\left(\tau_{i+1}\left(\frac{T_{\tau_p}}{T_{\tau_{i+1}}}\right)\right)$. Since $\vec{c}^p_1$ is an immediate forward job chain, we know that j'_{i+1} is the *minimal* number in $\mathbb{N}$ with $\mathrm{we}(\tau_i(j'_i)) \le \mathrm{re}(\tau_{i+1}(j'_{i+1}))$. Hence, $j'_{i+1} = \frac{T_{\tau_p}}{T_{\tau_{i+1}}}$. Please note that $\frac{T_{\tau_p}}{T_{\tau_{i+1}}} \in \mathbb{N}$ since the task periods are max-harmonic. □ Equation (28)

We conclude that $\ell(pc^p_0) = \mathrm{we}(\tau_n(j'_n)) - \mathrm{re}(\tau_1(j_1)) = \mathrm{we}\left(\tau_n\left(\frac{T_{\tau_p}}{T_{\tau_n}}\right)\right) - \mathrm{re}(\tau_1(0)) = \phi_{\tau_n} + \left(\frac{T_{\tau_p}}{T_{\tau_n}} + 1\right) \cdot T_{\tau_n} - \phi_{\tau_1} = \sum_{\xi=1}^{n-1} T_{\tau_\xi} + T_{\tau_p} + T_{\tau_n} - 0 = \sum_{\xi=1}^{n} T_{\tau_\xi} + T_{\tau_p}$. Hence, the latency is $\mathrm{Lat}(E) = \ell(pc^p_0) = \sum_{i=1}^{n} T_{\tau_i} + T_{\tau_p} = \sum_{i=1}^{n} T_{\tau_i} + \max_{i=1,\dots,n} T_{\tau_i}$. □

In the following, we show that the suggested phasing is optimal. To that end, we formulate a general lower bound on the end-to-end latency.

Lemma 12 (Lower Bound)**.** *If $\mathbb{T}_E$ is max-harmonic, then the end-to-end latency of E under* any *task phasing is lower bounded by*

$$\mathrm{Lat}(E) \ge \sum_{i=1}^{n} T_{\tau_i} + \max_{i=1,\dots,n} T_{\tau_i}. \tag{29}$$

Proof. Let $p \in \{1,\dots,n\}$ such that τ_p is the task with the maximal period, i.e., $T_{\tau_p} = \max_{i=1,\dots,n} T_{\tau_i}$. Consider any p-partitioned job chain $pc^p_m = (\overleftarrow{c}^p_m, \vec{c}^p_{m+1})$. We denote the immediate backward job chain by $\overleftarrow{c}^p_m = (\tau_1(j_1) \to \cdots \to \tau_p(j_p))$ and the immediate forward job chain by $\vec{c}^p_{m+1} = (\tau_p(j'_p) \to \cdots \to \tau_n(j'_n))$. The length of pc^p_m can be expressed by:

$$\begin{aligned}
\ell(pc^p_m) &= \mathrm{we}(\tau_n(j'_n)) - \mathrm{re}(\tau_1(j_1)) && (30)\\
&= \mathrm{we}(\tau_n(j'_n)) - \mathrm{we}(\tau_p(j'_p)) && (31)\\
&\quad + \mathrm{we}(\tau_p(j'_p)) - \mathrm{re}(\tau_p(j_p)) && (32)\\
&\quad + \mathrm{re}(\tau_p(j_p)) - \mathrm{re}(\tau_1(j_1)) && (33)
\end{aligned}$$

In the following we investigate each summand (Equations (31), (32), (33)) individually.

Summand of Equation (31)**:** This summand can be written as the sum $(31) = \sum_{i=p+1}^{n} (\mathrm{we}(\tau_i(j'_i)) - \mathrm{we}(\tau_{i-1}(j'_{i-1})))$. We use the property $\mathrm{we}(\tau_{i-1}(j'_{i-1})) \le \mathrm{re}(\tau_i(j'_i))$ of immediate forward job chains to bound (31) from below by $(31) \ge \sum_{i=p+1}^{n} (\mathrm{we}(\tau_i(j'_i)) - \mathrm{re}(\tau_i(j'_i)))$. Since $\mathrm{we}(\tau_i(j'_i)) - \mathrm{re}(\tau_i(j'_i)) = T_{\tau_i}$, we obtain $(31) \ge \sum_{i=p+1}^{n} T_{\tau_i}$.

Summand of Equation (32)**:** By definition, $j_p = m$ and $j'_p = m+1$. Therefore, $(32) = (\phi_{\tau_p} + (m+1+1) \cdot T_{\tau_p}) - (\phi_{\tau_p} + m \cdot T_{\tau_p}) = 2 \cdot T_{\tau_p}$.

Summand of Equation (33)**:** This summand is $(33) = \sum_{i=1}^{p-1} (\mathrm{re}(\tau_{i+1}(j_{i+1})) - \mathrm{re}(\tau_i(j_i)))$. We use the property $\mathrm{re}(\tau_{i+1}(j_{i+1})) \ge \mathrm{we}(\tau_i(j_i))$ of immediate backward job chains to achieve $(33) \ge \sum_{i=1}^{p-1} (\mathrm{we}(\tau_i(j_i)) - \mathrm{re}(\tau_i(j_i))) = \sum_{i=1}^{p-1} T_{\tau_i}$.

We conclude $\ell(pc^p_m) = (31) + (32) + (33) \ge \sum_{i=1}^{n} T_{\tau_i} + T_{\tau_p} = \sum_{i=1}^{n} T_{\tau_i} + \max_{i=1,\dots,n} T_{\tau_i}$. Hence, $\mathrm{Lat}(E) = \sup_{m \ge W_p} \ell(pc^p_m) \ge \sum_{i=1}^{n} T_{\tau_i} + \max_{i=1,\dots,n} T_{\tau_i}$ under any task phasing. □

Since the lower bound coincides with the end-to-end latency for max-harmonic task sets with our suggested phasing, our phasing is optimal.

Theorem 13 (Optimality)**.** *If $\mathbb{T}_E$ is max-harmonic, then the task phasing presented in Equation* (25) *minimizes the end-to-end latency. In particular, the suggested phasing minimizes the end-to-end latency in harmonic task sets.*

Proof. Since the end-to-end latency under our proposed phasing (Theorem 11) coincides with the lower bound under any phasing (Lemma 12), our task phasing is optimal. □

VII. Optimal Task Phasing for $(2,k)$-Max-Harmonic Systems

In industrial applications, task periods are chosen such that the systems are *almost* harmonic (also called *semi-harmonic*). Especially in the automotive domain, it is common to select task periods such that

$$T_\tau \in T^{Aut} := \{1, 2, 5, 10, 20, 50, 100, 200, 1000\} \tag{34}$$

for all $\tau \in \mathbb{T}$, as specified by real-world automotive benchmarks [26]. While for cause-effect chains $E = (\tau_1 \to \cdots \to \tau_n)$ with $T_{\tau_i} \in T^{Aut}$ for all $i = 1, \ldots, n$, the tasks $\mathbb{T}_E$ are often max-harmonic, for which optimal phasing is determined in Section VI, the following two cases are not covered:

(i) The two largest periods of $\mathbb{T}_E$ are 2 and 5.
(ii) The two largest periods of $\mathbb{T}_E$ are 20 and 50.

In this section, we extend our results to $(2,k)$-*max-harmonic* task sets (cf. Definition 2) to cover (i) and (ii), and therefore allow the application of our results to more realistic cases. For the sake of readability, in this section we refer to the two largest periods of $\mathbb{T}_E$ as

$$T^E_{max,1} := T_{\max,1}(\mathbb{T}_E) \quad \text{and} \quad T^E_{max,2} := T_{\max,2}(\mathbb{T}_E). \tag{35}$$

While it is sufficient to only consider *one* partitioned job chain $pc^p_{W_p}$ to calculate the end-to-end latency if $\mathbb{T}_E$ is max-harmonic, as proven in Lemma 9, if $\mathbb{T}_E$ is $(2,k)$-max-harmonic, we need to consider *two* partitioned job chains instead, namely $pc^p_{W_p}$ and $pc^p_{W_p+1}$.

Lemma 14. *Let τ_p be a task with the largest period of E, i.e., $T_{\tau_p} = T^E_{max,1}$. If $\mathbb{T}_E$ is $(2,k)$-max-harmonic, then*

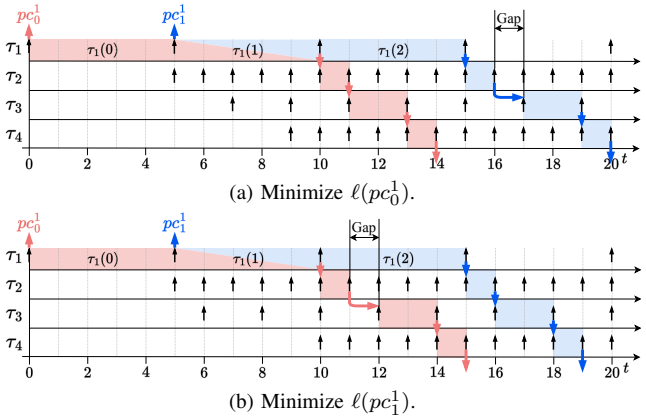
$$\mathrm{Lat}(E) = \max\left(\ell(pc^p_{W_p}), \ell(pc^p_{W_p+1})\right) \tag{36}$$

is the end-to-end latency.

Proof. By definition, if $\mathbb{T}_E$ is $(2,k)$-max-harmonic, then the hyperperiod $H(\mathbb{T}_E) = \mathrm{LCM}(\{T_{\tau_1}, \ldots, T_{\tau_n}\})$ of tasks of $\mathbb{T}_E$ is two times the largest period, i.e., $H(\mathbb{T}_E) = 2 \cdot T^E_{max,1}$. Applying Lemma 8 proves the result of this lemma. □

Building upon the results of the previous lemma, to minimize the end-to-end latency, we need to minimize the maximal length of both $pc^p_{W_p}$ and $pc^p_{W_p+1}$. However, while it is possible to achieve either $pc^p_{W_p} = \sum_{i=1}^n T_{\tau_i} + T^E_{max,1}$ or $pc^p_{W_p+1} = \sum_{i=1}^n T_{\tau_i} + T^E_{max,1}$, it is not possible to achieve both simultaneously, as demonstrated by the following example, depicted in Figure 5.

Example 15. We consider a cause-effect chain $E = (\tau_1 \to \tau_2 \to \tau_3 \to \tau_4)$ with four tasks. The tasks have periods $T_{\tau_1} = 5$, $T_{\tau_2} = 1$, $T_{\tau_3} = 2$ and $T_{\tau_4} = 1$. Therefore, $\mathbb{T}_E$ is $(2,k)$-max-harmonic. The task τ_1 has the largest period. Therefore, we can choose $p = 1$ in Lemma 14. The first immediate backward job chain is $\overleftarrow{c}_0(E) = (\tau_1(0) \to \tau_2(1) \to \tau_3(0) \to \tau_4(0))$ in Figure 5a and $\overleftarrow{c}_0(E) = (\tau_1(0) \to \tau_2(2) \to \tau_3(1) \to \tau_4(0))$ in Figure 5b. Therefore, in both cases $W_p = W_1 = 0$ (cf. Definition 5). By Lemma 14 the end-to-end latency is determined by the 1-partitioned job chains pc^1_0 and pc^1_1 marked in red and blue, respectively, in Figure 5. If we minimize $\ell(pc^1_0)$ by aligning write-events and read-events of subsequent jobs, we obtain a gap of one time unit in pc^1_1, as depicted in Figure 5a. Specifically, for pc^1_1, the write-event of task τ_2 at 16 and the read-event of τ_3 at 17 do not align. Conversely, if we minimize $\ell(pc^1_1)$, there is a gap of one time unit in pc^1_0, as depicted in Figure 5b.



(a) Minimize $\ell(pc^1_0)$.

(b) Minimize $\ell(pc^1_1)$.

Figure 5. Example of four tasks demonstrating that not both $\ell(pc^1_0)$ and $\ell(pc^1_1)$ can be minimized simultaneously. The partitioned job chain pc^1_0 is marked in red and pc^1_1 is marked in blue. The jobs belonging to both are half red and half blue. The resulting gaps are marked in the figure.

The preceding example shows that *gaps* must be introduced either for $pc^p_{W_p}$ or $pc^p_{W_p+1}$. Before discussing the reason for those gaps, we first formalize the definition of gaps and their impact on the length on $pc^p_{W_p}$ and $pc^p_{W_p+1}$.

Definition 16 (Gaps)**.** Let p be the lowest index with $T_{\tau_p} = T^E_{max,1}$. Given the first p-partitioned job chain after the warm-up $pc^p_{W_p} = (\overleftarrow{c}^p_{W_p}, \vec{c}^p_{W_p+1})$ with

$$\overleftarrow{c}^p_{W_p} = (\tau_1(j'_1) \to \cdots \to \tau_p(j'_p)) \tag{37}$$

$$\vec{c}^p_{W_p+1} = (\tau_p(j_p) \to \cdots \to \tau_n(j_n)), \tag{38}$$

then we denote the gaps between write-events and read-events for $pc^p_{W_p}$ as $\gamma_1, \ldots, \gamma_{n-1}$ with:

$$\gamma_i := \begin{cases} \mathrm{re}(\tau_{i+1}(j'_{i+1})) - \mathrm{we}(\tau_i(j'_i)) & \text{, if } i < p \\ \mathrm{re}(\tau_{i+1}(j_{i+1})) - \mathrm{we}(\tau_i(j_i)) & \text{, if } i \geq p \end{cases} \tag{39}$$

Furthermore, given the second p-partitioned job chain after the warm-up $pc^p_{W_p+1} = (\overleftarrow{c}^p_{W_p+1}, \vec{c}^p_{W_p+2})$ with

$$\overleftarrow{c}^p_{W_p+1} = (\tau_1(\tilde{j}'_1) \to \cdots \to \tau_p(\tilde{j}'_p)) \tag{40}$$

$$\vec{c}^p_{W_p+2} = (\tau_p(\tilde{j}_p) \to \cdots \to \tau_n(\tilde{j}_n)), \tag{41}$$

then we denote the gaps for $pc^p_{W_p+1}$ as $\tilde{\gamma}_1, \dots, \tilde{\gamma}_{n-1}$ with:

$$\tilde{\gamma}_i \coloneqq \begin{cases} \mathrm{re}(\tau_{i+1}(\tilde{j}'_{i+1})) - \mathrm{we}(\tau_i(\tilde{j}'_i)) & \text{, if } i < p \\ \mathrm{re}(\tau_{i+1}(\tilde{j}_{i+1})) - \mathrm{we}(\tau_i(\tilde{j}_i)) & \text{, if } i \geq p \end{cases} \tag{42}$$

In Example 15, the gaps are $\gamma_1 = \gamma_2 = \gamma_3 = \tilde{\gamma}_1 = \tilde{\gamma}_3 = 0$ and $\tilde{\gamma}_2 = 1$ for Figure 5a, and $\gamma_1 = \gamma_3 = \tilde{\gamma}_1 = \tilde{\gamma}_2 = \tilde{\gamma}_3 = 0$ and $\gamma_2 = 1$ for Figure 5b. Given the gaps of $pc^p_{W_p}$ and $pc^p_{W_p+1}$, then their lengths can be derived as follows.

Lemma 17. *The length of a p-partitioned job chain is the sum of $\sum_{i=1}^n T_{\tau_i} + T_{\tau_p}$ and its gaps. Specifically, with p, $pc^p_{W_p}$ and $pc^p_{W_p+1}$ as in Definition 16,*

$$\ell(pc^p_{W_p}) = \sum_{i=1}^{n} T_{\tau_i} + T_{\tau_p} + \sum_{i=1}^{n-1} \gamma_i \tag{43}$$

$$\ell(pc^p_{W_p+1}) = \sum_{i=1}^{n} T_{\tau_i} + T_{\tau_p} + \sum_{i=1}^{n-1} \tilde{\gamma}_i \tag{44}$$

holds.

Proof. This is achieved by distributing the whole interval from read-event of the first job to write-event of the last job of the partitioned job chain into the parts where data is processed by the chain, i.e., read by a task but not written yet, and where it is waiting to be processed, i.e., written by a task but not read by the subsequent task yet. Formally,

$$\ell(pc^p_{W_p}) = \mathrm{we}(\tau_n(j_n)) - \mathrm{re}(\tau_1(j'_1)) \tag{45}$$

$$= A + B \tag{46}$$

where A is defined as $\sum_{i=1}^{p}(\mathrm{we}(\tau_i(j'_i)) - \mathrm{re}(\tau_i(j'_i))) + \sum_{i=p}^{n}(\mathrm{we}(\tau_i(j_i)) - \mathrm{re}(\tau_i(j_i)))$ and B is defined as $\sum_{i=1}^{p-1}(\mathrm{re}(\tau_{i+1}(j'_{i+1})) - \mathrm{we}(\tau_i(j'_i))) + \sum_{i=p}^{n-1}(\mathrm{re}(\tau_{i+1}(j_{i+1})) - \mathrm{we}(\tau_i(j_i)))$. By definition, $A = \sum_{i=1}^{n} T_{\tau_i} + T_{\tau_p}$ and $B = \sum_{i=1}^{n-1} \gamma_i$. This proves Equation (43). The proof of Equation (44) is analogous. □

In the following, we identify the reason for gaps. To that end, we reconsider the example from Figure 5. We observe that the jobs of τ_2 which are part of pc^1_0 and pc^1_1 (marked in red and blue) are released 5 time units apart. However, the release of jobs of τ_3 can only be multiples of 2 time units apart. Therefore, a gap must be introduced either for pc^1_1 or for pc^1_2. We generalize this observation in two steps. First, we show that a gap must be introduced whenever the distance between the relevant jobs changes (Lemma 19), and second, we derive bounds on the sum of gaps by determining how often the distance between the relevant jobs changes (Lemmas 20 and 21). For a proper formalization, we introduce the notion of *distance* between partitioned job chains.

Definition 18 (Distance)**.** Let p, $pc^p_{W_p}$ and $pc^p_{W_p+1}$ be as in Definition 16. We define $\Delta_1, \dots, \Delta_n$ by:

$$\Delta_i \coloneqq \begin{cases} \mathrm{re}(\tau_i(\tilde{j}'_i)) - \mathrm{re}(\tau_i(j'_i)) & \text{, if } i < p \\ \mathrm{re}(\tau_i(\tilde{j}_i)) - \mathrm{re}(\tau_i(j_i)) & \text{, if } i \geq p \end{cases} \tag{47}$$

Furthermore, we say that Δ_i is the distance between $pc^p_{W_p}$ and $pc^p_{W_p+1}$ at task τ_i.

Referring to Example 15, the distance between $pc^p_{W_p}$ and $pc^p_{W_p+1}$ is $\Delta_1 = \Delta_2 = 5$ and $\Delta_3 = \Delta_4 = 6$ for Figure 5a and $\Delta_1 = \Delta_2 = 5$ and $\Delta_3 = \Delta_4 = 4$ for Figure 5b. A gap γ_i is introduced, whenever the distance changes, i.e., $\Delta_i \neq \Delta_{i+1}$.

Lemma 19. *If $\mathbb{T}_E$ is $(2,k)$-max-harmonic, and $\Delta_i \neq \Delta_{i+1}$ for some $i \in \{1, \dots, n\}$, then $\gamma_i \geq \Gamma_E$ or $\tilde{\gamma}_i \geq \Gamma_E$ with*

$$\Gamma_E \coloneqq T^E_{max,1} \bmod T^E_{max,2}. \tag{48}$$

Proof. The proof is divided in two parts: first, we prove that for all $i \in \{1, \dots, n\}$, the distance Δ_i is an integer multiple of $T^E_{max,1}$ or $T^E_{max,2}$, i.e.,

$$T^E_{max,1} \mid \Delta_i \qquad \text{or} \qquad T^E_{max,2} \mid \Delta_i, \tag{49}$$

and second, we utilize Equation (49) to prove Equation (48).

Proof of Equation (49)**:** Let $I \subseteq \{1, \dots, n\}$ be the set of indices i such that Equation (49) does *not* hold. We want to show that $I = \emptyset$. First, we observe that $p \notin I$ because $\Delta_p = T^E_{max,1}$ by definition. Second, we show that all $i < p$ are not in I by contradiction. To that end, we assume $I \cap \{1, \dots, p-1\} \neq \emptyset$, and let i be the largest element of $I \cap \{1, \dots, p-1\}$. We will lead different cases to contradiction.

(i) $T_{\tau_i} \mid \Delta_{i+1}$. First, we observe that we have $\Delta_{i+1} = \mathrm{re}(\tau_{i+1}(\tilde{j}'_{i+1})) - \mathrm{re}(\tau_{i+1}(j'_{i+1}))$, which follows directly from Definition 18 if $i \leq p-2$, or is because of $\Delta_{i+1} = \mathrm{re}(\tau_p(\tilde{j}_p)) - \mathrm{re}(\tau_p(j_p)) = \mathrm{re}(\tau_p(\tilde{j}'_p)) - \mathrm{re}(\tau_p(j'_p))$ if $i = p-1$. Since $\tau_i(j'_i)$ is (by definition) the latest job of τ_i with write-event no later than $\mathrm{re}(\tau_{i+1}(j'_{i+1}))$, the job $\tau_i(j'_i + \frac{\Delta_{i+1}}{T_{\tau_i}})$ is the latest job of τ_i with write-event no later than $\mathrm{re}(\tau_{i+1}(\tilde{j}'_{i+1}))$, i.e., $\tilde{j}'_i = j'_i + \frac{\Delta_{i+1}}{T_{\tau_i}}$. Therefore, we obtain $\Delta_i = \mathrm{re}(\tau_i(\tilde{j}'_i)) - \mathrm{re}(\tau_i(j'_i)) = \tau_i(j'_i + \frac{\Delta_{i+1}}{T_{\tau_i}}) - \mathrm{re}(\tau_i(j'_i)) = \Delta_{i+1}$. Hence, also $i+1 \in I$ which contradicts the maximality of i.

(ii) $T_{\tau_i} \nmid \Delta_{i+1}$ and $T^E_{max,1} \mid \Delta_{i+1}$. We know that $\mathbb{T}_E$ is $(2,k)$-max-harmonic. Therefore, $T_{\tau_i} \mid T^E_{max,1}$ for all $T_{\tau_i} \neq T^E_{max,2}$. If $T_{\tau_i} \mid T^E_{max,1}$, then $T_{\tau_i} \mid T^E_{max,1} \mid \Delta_{i+1}$ which contradicts $T_{\tau_i} \nmid \Delta_{i+1}$. Therefore, $T_{\tau_i} = T^E_{max,2}$ must hold. Since Δ_i is by definition an integer multiple of T_{τ_i}, we have $T^E_{max,2} \mid \Delta_i$ and Equation (49) holds for i which contradicts $i \in I$.

(iii) $T_{\tau_i} \nmid \Delta_{i+1}$ and $T^E_{max,2} \mid \Delta_{i+1}$. The proof is analogous to (ii). That is, either $T_{\tau_i} \mid T^E_{max,2}$ or $T_{\tau_i} = T^E_{max,1}$ must hold, because $\mathbb{T}_E$ is $(2,k)$-max-harmonic. However, $T_{\tau_i} \mid T^E_{max,2}$ contradicts $T_{\tau_i} \nmid \Delta_{i+1}$ because $T^E_{max,2} \mid \Delta_{i+1}$. Furthermore, $T_{\tau_i} = T^E_{max,1}$ contradicts $i \in I$.

Analogously, we show that $I \cap \{p+1, \dots, n\} = \emptyset$ by contradiction. To that end, we let i be the smallest element of $I \cap \{p+1, \dots, n\}$ and distinguish the cases (i) $T_{\tau_i} \mid \Delta_{i-1}$, (ii) $T_{\tau_i} \nmid \Delta_{i-1}$ and $T^E_{max,1} \mid \Delta_{i-1}$, and (iii) $T_{\tau_i} \nmid \Delta_{i-1}$ and $T^E_{max,2} \mid \Delta_{i-1}$. We conclude $I = \emptyset$, i.e., Equation (49) holds.

Proof of Equation (48)**:** First, we prove that

$$\tilde{\gamma}_i = \gamma_i + \Delta_{i+1} - \Delta_i. \tag{50}$$

For $i < p$, we obtain $\tilde{\gamma}_i \overset{(42)}{=} \mathrm{re}(\tau_{i+1}(\tilde{j}'_{i+1})) - T_{\tau_i} - \mathrm{re}(\tau_i(\tilde{j}'_i)) \overset{(47)}{=} \Delta_{i+1} + \mathrm{re}(\tau_{i+1}(j'_{i+1})) - T_{\tau_i} - \Delta_i - \mathrm{re}(\tau_i(j'_i)) \overset{(39)}{=} \gamma_i + \Delta_{i+1} - \Delta_i$. The case $i \geq p$ is analogous.

Due to Equation (50), we know that $\tilde{\gamma}_i \geq \Delta_{i+1} - \Delta_i$ and $\gamma_i \geq \Delta_i - \Delta_{i+1}$. Therefore, $\max(\gamma_i, \tilde{\gamma}_i) \geq |\Delta_{i+1} - \Delta_i|$, and it is left to show that $|\Delta_{i+1} - \Delta_i| \geq \Gamma_E$. Since Equation (49) holds, it is sufficient to distinguish the following four cases:

(i) $T^E_{max,1} \mid \Delta_i$ and $T^E_{max,1} \mid \Delta_{i+1}$. Since $\Delta_i \neq \Delta_{i+1}$, $|\Delta_{i+1} - \Delta_i| \geq T^E_{max,1}$, which is $\geq \Gamma_E$.

(ii) $T^E_{max,2} \mid \Delta_i$ and $T^E_{max,2} \mid \Delta_{i+1}$. Since $\Delta_i \neq \Delta_{i+1}$, $|\Delta_{i+1} - \Delta_i| \geq T^E_{max,2}$, which is $\geq \Gamma_E$.

(iii) $T^E_{max,1} \mid \Delta_i$ and $T^E_{max,2} \mid \Delta_{i+1}$. Since $\Delta_{i+1} \neq \Delta_i$, there exist $a, b \in \mathbb{Z}$ such that $|a \cdot T^E_{max,1} - b \cdot T^E_{max,2}| > 0$ and $|\Delta_{i+1} - \Delta_i| = |a \cdot T^E_{max,1} - b \cdot T^E_{max,2}|$. Since we know that $T^E_{max,1} = \frac{k}{2} \cdot T^E_{max,2}$, we obtain $|a \cdot T^E_{max,1} - b \cdot T^E_{max,2}| = |(a \cdot \frac{k}{2} - b) \cdot T^E_{max,2}|$. If $2 \mid a$, then $|(a \cdot \frac{k}{2} - b) \cdot T^E_{max,2}|$ (under the assumption that $|(a \cdot \frac{k}{2} - b) \cdot T^E_{max,2}| > 0$) is minimized with $b = a \cdot \frac{k}{2} + 1$, i.e., $|(a \cdot \frac{k}{2} - b) \cdot T^E_{max,2}| \geq T^E_{max,2}$. On the other hand, if $2 \nmid a$, then $|(a \cdot \frac{k}{2} - b) \cdot T^E_{max,2}|$ is minimized with $b = \lceil a \cdot \frac{k}{2} \rceil$, i.e., $|(a \cdot \frac{k}{2} - b) \cdot T^E_{max,2}| \geq \frac{1}{2} T^E_{max,2} = T^E_{max,1} \bmod T^E_{max,2}$.

(iv) $T^E_{max,2} \mid \Delta_i$ and $T^E_{max,1} \mid \Delta_{i+1}$. This case is analogous to the case (iii).

In particular, Equation (48) holds if $\Delta_i \neq \Delta_{i+1}$, which proves Lemma 19. □

Using the previous lemma, to quantify the sum of gaps, we need to check how often gaps are introduced. To that end, we distinguish two cases: either $\Delta_i \bmod H(\mathbb{T}_E) \neq 0$ for all $i = 1, \ldots, n$, or there exists an $i \in \{1, \ldots, n\}$ such that $\Delta_i \bmod H(\mathbb{T}_E) = 0$. We start by examining the former case.

The case '$\Delta_i \bmod H(\mathbb{T}_E) \neq 0 \quad \forall i = 1, \ldots, n$' is depicted in Figure 5. We observe that Δ_i changes whenever the period of tasks in the cause-effect chain switches between $T^E_{max,1}$ and $T^E_{max,2}$. Specifically, Δ_i has to change once between τ_1 and τ_3 in Figure 5. Formally, we quantify the number of necessary changes of Δ_i as $|\nu_E|$ with

$$\nu_E \coloneqq \left\{ \tau_i \in \mathbb{T}_E \;\middle|\; \begin{array}{l} \exists j < i \colon T_{\tau_j} \neq T_{\tau_i} \in \left\{T^E_{max,1}, T^E_{max,2}\right\}, \\ T_{\tau_{j+1}}, \ldots, T_{\tau_{i-1}} \notin \left\{T^E_{max,1}, T^E_{max,2}\right\} \end{array} \right\}. \tag{51}$$

In Figure 5, only τ_1 and τ_3 have a period in $\left\{T^E_{max,1}, T^E_{max,2}\right\}$. Therefore, $\nu_E = \{\tau_3\}$, which directly corresponds to the tasks where a change of the gap is observed. We formalize this observation in the following lemma.

Lemma 20. *Let τ_p be the task with the lowest index with $T_{\tau_p} = T^E_{max,1}$. If $\mathbb{T}_E$ is $(2,k)$-max-harmonic, and if further $\Delta_i \bmod H(\mathbb{T}_E) \neq 0$ for all $i = 1, \ldots, n$, then*

$$\max\left(\sum_{i=1}^{n-1} \gamma_i, \sum_{i=1}^{n-1} \tilde{\gamma}_i\right) \geq \left\lceil \frac{|\nu_E|}{2} \right\rceil \cdot \Gamma_E \tag{52}$$

with Γ_E as defined in Equation (48)*.*

Proof. For each task $\tau_i \in \nu_E$, there exists a $j < i$ with $T_{\tau_j} \neq T_{\tau_i} \in \left\{T^E_{max,1}, T^E_{max,2}\right\}$ and $T_{\tau_{j+1}}, \ldots, T_{\tau_{i-1}} \notin \left\{T^E_{max,1}, T^E_{max,2}\right\}$. Since $T_{\tau_j} \neq T_{\tau_i} \in \left\{T^E_{max,1}, T^E_{max,2}\right\}$ and $\Delta_i \bmod H(\mathbb{T}_E) \neq 0$, we know that $\Delta_i \neq \Delta_j$. Hence, there exists a $\xi(j) \in \{j, \ldots, i-1\}$ such that $\Delta_{\xi(j)} \neq \Delta_{\xi(j)+1}$. By Lemma 19, we obtain $\gamma_{\xi(j)} \geq \Gamma_E$ or $\tilde{\gamma}_{\xi(j)} \geq \Gamma_E$.

Due to $T_{\tau_{j+1}}, \ldots, T_{\tau_{i-1}} \notin \left\{T^E_{max,1}, T^E_{max,2}\right\}$, the indices $\xi(j)$ are different for each $\tau_j \in \nu_E$, i.e., $\xi(j) \neq \xi(j')$ for all $\tau_j \neq \tau_{j'} \in \nu_E$. Consequently, among all gaps, there are at most $|\nu_E|$ gaps of size $\geq \Gamma_E$. Hence, $\sum_{i=1}^{n-1} \gamma_i \geq \left\lceil \frac{|\nu_E|}{2} \right\rceil \cdot \Gamma_E$ or $\sum_{i=1}^{n-1} \tilde{\gamma}_i \geq \left\lceil \frac{|\nu_E|}{2} \right\rceil \cdot \Gamma_E$. This proves Equation (52). □

For the case '$\exists i \in \{1, \ldots, n\} : \Delta_i \bmod H(\mathbb{T}_E) \neq 0$', the sum of gaps must be at least $T^E_{max,1}$ for $pc^p_{W_p}$ or $pc^p_{W_p+1}$, as formalized by the following lemma.

Lemma 21. *Let τ_p be the task with the lowest index with $T_{\tau_p} = T^E_{max,1}$. If $\mathbb{T}_E$ is $(2,k)$-max-harmonic, and if further there exists an $i \in \{1, \ldots, n\}$ such that $\Delta_i \bmod H(\mathbb{T}_E) = 0$, then*

$$\max\left(\sum_{i=1}^{n-1} \gamma_i, \sum_{i=1}^{n-1} \tilde{\gamma}_i\right) \geq T^E_{max,1}. \tag{53}$$

Proof. We know that $\Delta_p = T^E_{max,1}$ by construction. Further, if there exists a $\xi \in \{1, \ldots, n\}$ with $\Delta_\xi \bmod H(\mathbb{T}_E) = 0$, then $\Delta_\xi = 0$ or $\Delta_\xi \geq H(\mathbb{T}_E) = 2 \cdot T^E_{max,1}$. In any case,

$$|\Delta_p - \Delta_\xi| \geq T^E_{max,1}. \tag{54}$$

Furthermore, by definition of gaps, we have

$$|\Delta_\xi - \Delta_p| = \left| \sum_{i \in I} \gamma_i - \sum_{i \in I} \tilde{\gamma}_i \right| \leq \max\left(\sum_{i=1}^{n-1} \gamma_i, \sum_{i=1}^{n-1} \tilde{\gamma}_i\right), \tag{55}$$

with indices $I = \{\xi, \ldots, p-1\}$ if $\xi < p$ and $I = \{p, \ldots, \xi-1\}$ if $\xi > p$. Combining Equations (54) and (55) proves the lemma. □

We combine Lemmas 20 and 21 to obtain a lower bound for $(2,k)$-max-harmonic systems.

Lemma 22 (Lower Bound)**.** *If $\mathbb{T}_E$ is $(2,k)$-max-harmonic, then*

$$\mathrm{Lat}(E) \geq \sum_{i=1}^{n} T_{\tau_i} + T^E_{max,1} + \min\left(\left\lceil \frac{|\nu_E|}{2} \right\rceil \cdot \Gamma_E, T^E_{max,1}\right) \tag{56}$$

is a lower bound on the end-to-end latency under any task phasing.

Proof. This is achieved by combining the results of Lemmas 20 and 21. □

Next, we construct a task phasing that achieves that lower bound. To that end, we need to adjust the phasing strategy depending on the lower bound to be achieved. That is, if $\left\lceil \frac{|\nu_E|}{2} \right\rceil \cdot \Gamma_E < T^E_{max,1}$, then we distribute the gaps between the two partitioned chains $pc^p_{W_p}$ and $pc^p_{W_p+1}$ equally. However, if $\left\lceil \frac{|\nu_E|}{2} \right\rceil \cdot \Gamma_E \geq T^E_{max,1}$, we ensure that all gaps after τ_p are allocated to pc^p_1.

Definition 23 (Suggested Phasing for $(2,k)$-Max-Harmonic)**.** Let $E = (\tau_1 \to \cdots \to \tau_n)$ be a cause-effect chain and let p be the lowest index with $T_{\tau_p} := T^E_{max,1}$. For the first task, we set the phasing to

$$\phi^{(2,k)}_{\tau_1} := 0 \tag{57}$$

For the remaining tasks, we define the phasing iteratively. Specifically, for $i = 2, \ldots, n$, we define

$$\phi^{(2,k)}_{\tau_i} := \phi^{(2,k)}_{\tau_{i-1}} + \Gamma_E + T_{\tau_{i-1}} \tag{58}$$

if $\left\lceil \frac{|\nu_E|}{2} \right\rceil \cdot \Gamma_E < T^E_{max,1}$ and $\tau_i \neq \tau_p \in \nu_E$ with $T_{\tau_i} = T^E_{max,1}$, and

$$\phi^{(2,k)}_{\tau_i} := \phi^{(2,k)}_{\tau_{i-1}} + T_{\tau_{i-1}} \tag{59}$$

otherwise.

Theorem 24 (End-to-end latency with suggested phasing)**.** *If $\mathbb{T}_E$ is $(2,k)$-max-harmonic and the task offsets are chosen as $\phi_{\tau_i} = \phi^{(2,k)}_{\tau_i}$, then*

$$\mathrm{Lat}(E) = \sum_{i=1}^{n} T_{\tau_i} + T^E_{max,1} + \min\left(\left\lceil \frac{|\nu_E|}{2} \right\rceil \cdot \Gamma_E, T^E_{max,1} \right). \tag{60}$$

Proof. By Lemma 22, we already know that Equation (60) is a lower bound on the end-to-end latency of E. It is left to show that Equation (60) is also an upper bound. Let τ_p be a task with the largest period of E, i.e., $T_{\tau_p} = T^E_{max,1}$.

By definition of the phases, the immediate backward job chain $\overleftarrow{c}_0(E)$ exists. We show $W_p = 0$ by contradiction. To that end, assume $W_p \geq 1$, then $\ell(\overleftarrow{c}_0((\tau_p \to \cdots \to \tau_n))) \leq \phi^{(2,k)}_{\tau_n} - (\phi^{(2,k)}_{\tau_p} + T_{\tau_p})$. However, by definition of the phases, if $\left\lceil \frac{|\nu_E|}{2} \right\rceil \cdot \Gamma_E < T^E_{max,1}$, then $\phi^{(2,k)}_{\tau_n} - \phi^{(2,k)}_{\tau_p} \leq \sum_{\xi=p}^{n} T_{\tau_\xi} + \left\lceil \frac{|\nu_E|}{2} \right\rceil \cdot \Gamma_E < \sum_{\xi=p}^{n} T_{\tau_\xi} + T^E_{max,1}$, and $\phi^{(2,k)}_{\tau_n} - \phi^{(2,k)}_{\tau_p} \leq \sum_{\xi=p}^{n} T_{\tau_\xi}$, otherwise. We conclude $\ell(\overleftarrow{c}_0((\tau_p \to \cdots \to \tau_n))) < \sum_{\xi=p}^{n} T_{\tau_\xi} + T^E_{max,1} - T_{\tau_p} = \sum_{\xi=p}^{n} T_{\tau_\xi}$, which contradicts the fact, that a backward job chain for $(\tau_p \to \cdots \to \tau_n)$ has a length of at least $\sum_{\xi=p}^{n} T_{\tau_\xi}$. Therefore, this proves $W_p = 0$.

Therefore, we have $pc^p_{W_p} = pc^p_0 = (\overleftarrow{c}^p_0, \vec{c}^p_1)$ and $pc^p_{W_p+1} = pc^p_1 = (\overleftarrow{c}^p_1, \vec{c}^p_2)$, and by Lemma 14, the end-to-end latency is $\max(\ell(pc^p_0), \ell(pc^p_1))$, which is

$$\mathrm{Lat}(E) = \max(\ell(\overleftarrow{c}^p_0) + \ell(\vec{c}^p_1), \ell(\overleftarrow{c}^p_1) + \ell(\vec{c}^p_2)). \tag{61}$$

First, we quantify $\ell(\overleftarrow{c}^p_0)$ and $\ell(\overleftarrow{c}^p_1)$. Since p is the lowest index with $T_{\tau_p} = T^E_{max,1}$, we know that $\tau_i \notin \nu_E$ for all $i < p$. Therefore, $\tau_1, \ldots, \tau_p$ follow the phasing from Equation (59). Hence, $\overleftarrow{c}^p_0 = (\tau_1(0) \to \cdots \to \tau_p(0))$ and $\ell(\overleftarrow{c}^p_0) = \sum_{i=1}^{p} T_{\tau_i}$. We determine $\ell(\overleftarrow{c}^p_1)$ by distinguishing two cases:

Case 1 ($\tau_p \notin \nu_E$)**:** In that case, $T_{\tau_i} \mid T^E_{max,1}$ for all $i \leq p$. Therefore, $\overleftarrow{c}^p_1 = (\tau_1(\frac{T^E_{max,1}}{T_{\tau_1}}) \to \cdots \to \tau_p(\frac{T^E_{max,1}}{T_{\tau_p}}))$, i.e., $\overleftarrow{c}^p_1$ is just $\overleftarrow{c}^p_0$ shifted by $T^E_{max,1}$ time units, and $\ell(\overleftarrow{c}^p_1) = \sum_{i=1}^{p} T_{\tau_i}$.

Case 2 ($\tau_p \in \nu_E$)**:** In that case, let $j < p$ be the largest index such that $T_{\tau_j} = T^E_{max,2}$. Then $T_{\tau_1}, \ldots, T_{\tau_j} \mid T^E_{max,2}$ and $T_{\tau_{j+1}}, \ldots, T_{\tau_p} \mid T^E_{max,1}$. Hence, $\overleftarrow{c}^p_1 = \big(\tau_1\big(\frac{T^E_{max,1} - \Gamma_E}{T_{\tau_1}}\big) \to \cdots \to \tau_j\big(\frac{T^E_{max,1} - \Gamma_E}{T_{\tau_j}}\big) \to \tau_{j+1}\big(\frac{T^E_{max,1}}{T_{\tau_{j+1}}}\big) \to \cdots \to \tau_p\big(\frac{T^E_{max,1}}{T_{\tau_p}}\big)\big)$. Therefore, $\ell(\overleftarrow{c}^p_1) = \ell(\overleftarrow{c}^p_0) + \Gamma_E = \sum_{i=1}^{p} T_{\tau_i} + \Gamma_E$.

Second, we quantify $\ell(\vec{c}^p_1)$ and $\ell(\vec{c}^p_2)$. To that end, we again distinguish two cases:

Case A ($\left\lceil \frac{|\nu_E|}{2} \right\rceil \cdot \Gamma_E \geq T^E_{max,1}$)**:** In that case, all tasks follow the phasing from Equation (59), and $(\tau_p(0) \to \cdots \to \tau_n(0))$ is an immediate forward job chain. Furthermore, $\vec{c}^p_2$ is just the forward chain $(\tau_p(0) \to \cdots \to \tau_n(0))$ shifted by $2 \cdot T^E_{max,1} = H(\mathbb{T}_E)$ time units, i.e., $\ell(\vec{c}^p_2) = \ell((\tau_p(H(\mathbb{T}_E)/T_{\tau_p}), \ldots, \tau_n(H(\mathbb{T}_E)/T_{\tau_n}))) = \ell((\tau_p(0), \ldots, \tau_n(0))) = \sum_{i=p}^{n} T_{\tau_i}$

For the chain $\vec{c}^p_1$, we know that the write-event of the last job cannot be later than the write-event of the last job of $\vec{c}^p_2$. Therefore, we obtain: $\ell(\vec{c}^p_1) \leq \mathrm{we}(\tau_n(H(\mathbb{T}_E)/T_{\tau_n})) - \mathrm{re}(\tau_p(1)) = \ell(\vec{c}^p_2) + T^E_{max,1} = \sum_{i=p}^{n} T_{\tau_i} + T^E_{max,1}$

Case B ($\left\lceil \frac{|\nu_E|}{2} \right\rceil \cdot \Gamma_E < T^E_{max,1}$)**:** For every task $\tau_i \in \nu_E$ with $i \geq p$ and $T_{\tau_i} = T^E_{max,1}$, there is a gap of size Γ_E introduced for $\vec{c}_0((\tau_p \to \cdots \to \tau_n))$. These are $\left\lfloor \frac{|\nu_E \setminus \{\tau_p\}|}{2} \right\rfloor$ many. Hence, $\ell(\vec{c}_0((\tau_p \to \cdots \to \tau_n))) = \sum_{i=p}^{n} T_{\tau_i} + \left\lfloor \frac{|\nu_E \setminus \{\tau_p\}|}{2} \right\rfloor \cdot \Gamma_E$, and since $\vec{c}^p_2$ is just $\vec{c}_0((\tau_p \to \cdots \to \tau_n))$ shifted by one hyperperiod, we obtain $\ell(\vec{c}^p_2) = \sum_{i=p}^{n} T_{\tau_i} + \left\lfloor \frac{|\nu_E \setminus \{\tau_p\}|}{2} \right\rfloor \cdot \Gamma_E$.

Furthermore, for every task $\tau_i \in \nu_E$ with $i \geq p$ and $T_{\tau_i} = T^E_{max,1}$ (i.e., $T_{\tau_i} = T^E_{max,2}$), a gap of size Γ_E is introduced for $\vec{c}^p_1$. These are $\left\lceil \frac{|\nu_E \setminus \{\tau_p\}|}{2} \right\rceil$ many. This results in $\ell(\vec{c}^p_1) = \sum_{i=p}^{n} T_{\tau_i} + \left\lceil \frac{|\nu_E \setminus \{\tau_p\}|}{2} \right\rceil \cdot \Gamma_E$.

All four combinations of cases 1 and 2, and A and B result in the bound from Equation (60). This proves the theorem. □

Finally, we state that our suggested phasing is optimal for non-max-harmonic tasks, since the end-to-end latency from Theorem 24 coincides with the lower bound from Lemma 22.

Theorem 25 (Optimality)**.** *If $\mathbb{T}_E$ is $(2,k)$-max-harmonic, then the suggested phasing from Definition 23 minimizes the end-to-end latency.*

Proof. Since the end-to-end latency under our proposed phasing (Theorem 24) coincides with the lower bound under any phasing (Lemma 22), our task phasing is optimal. □

VIII. Evaluation

In this section, we evaluate our approach based on the Autonomous Emergency Braking System (AEBS) use-case described in [21], and based on synthetic cause-effect chains[3]. For the Autonomous Emergency Braking System (AEBS), already depicted in Figure 1, we study two configurations: (i) with the original harmonic periods and (ii) a modified use-case where sensor sampling frequency is reduced, leading to semi-harmonic (specifically, $(2,5)$-max-harmonic) periods. For the synthetic evaluation, a first set of experiments is based on cause-effect chains with semi-harmonic automotive periods to evaluate the proposed task phasing quantitatively, followed

[3]https://github.com/ESRTS/OptimalTaskPhasing

Table II
RESULTS OF THE USE-CASE IN ITS HARMONIC AND SEMI-HARMONIC VARIANT. RUNTIME COMPARED TO THE HEURISTIC OF MARTINEZ [28].

	Harmonic	**Semi-Harmonic**
Synchronous $\mathrm{Lat}(E)$	210 ms	230 ms
Optimal $\mathrm{Lat}(E)$	170 ms	210 ms
Runtime Martinez	179.893 ms	669.251 ms
Runtime Ours	0.004 ms	0.018 ms

by experiments that evaluate different $(2, k)$-max-harmonic configurations. All experiments are performed on a platform containing an Intel Xeon Silver 4114 processor with 10 cores (20 threads) at 2.2 GHz and 32 GB RAM, running Linux.

A. Autonomous Emergency Braking System (AEBS) Use-Case

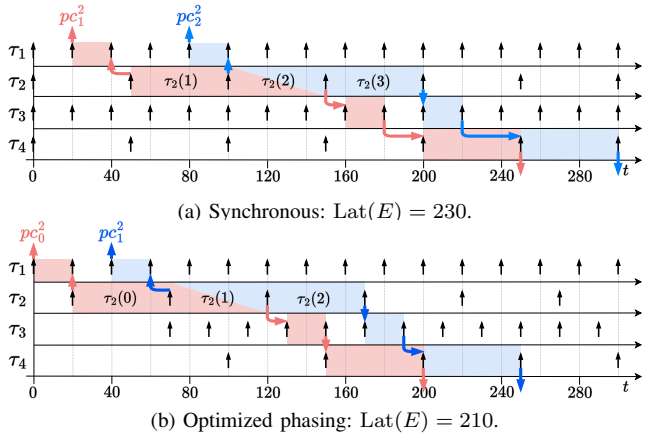
In this section, we compare the AEBS use-case in two configurations. For both, the runtime and minimal latency of the AEBS cause-effect chain is compared with our proposed approach and with the state-of-the-art (SOTA) approach of Martinez et al. [28]. All results are shown in Table II.

As discussed in Section V, by optimally configuring the tasks phases, the latency of the AEBS cause-effect chain under its original harmonic periods is reduced by 19% from 210 ms (synchronous release) to 170 ms. The proposed phasing (Definition 10) and latency bound (Theorem 11), as well as the SOTA analysis and heuristic correctly identify a phasing that leads to the minimal latency of the cause-effect chain. Our approach configures the task phases and analyzes the latency within 0.004 ms, while the heuristic requires significantly more time, 179.893 ms, in which 5000 configurations are explored.

To evaluate the AEBS use-case with our proposed phasing for semi-harmonic automotive periods, the period of sensor tasks (i.e. τ_1 and τ_3) is increased to 20 ms. With a synchronous release, the cause-effect chain has a latency of 230 ms, as depicted in Figure 6a. Applying the suggested phasing for $(2, k)$-max-harmonic periods (Definition 23) leads to the phasing $\phi_{\tau_1} = 0, \phi_{\tau_2} = 20, \phi_{\tau_3} = 70, \phi_{\tau_4} = 100$, as depicted in Figure 6b. We notice that $\Gamma_E = 10$ ms are additionally inserted at the second occurrence of $T^E_{max,1}$. Figure 6 shows the two 2-partitioned job chains under synchronous release, as well as under the proposed phasing. The gaps between the two 2-partitioned job chains under synchronous release are 10, 10, 20, for the red, and 0, 0, 30 for the blue 2-partitioned job chain. With our suggested phasing, the gaps are reduced to 0, 10, 0, for the red, and 10, 0, 10 for the blue 2-partitioned job chain. The proposed latency bound (Theorem 24), as well as the result of the heuristic lead to a latency of 210 ms, a reduction of 8.7%. Configuring task phases and latency calculation with our approach requires 0.018 ms, while the heuristic needs to explore 10 000 configurations, which takes 669.251 ms.

B. Synthetic Chains with Automotive Periods

We evaluate synthetic cause-effect chains, generated based on the benchmark reported in [26]. For each data point, 1000 cause-effect chains with automotive semi-harmonic periods are generated. To be able to compare latency values of different



(a) Synchronous: $\mathrm{Lat}(E) = 230$.

(b) Optimized phasing: $\mathrm{Lat}(E) = 210$.

Figure 6. The AEBS use-case with semi-harmonic automotive periods. The two 2-partitioned job chains relevant for the end-to-end latency are marked in red and blue.

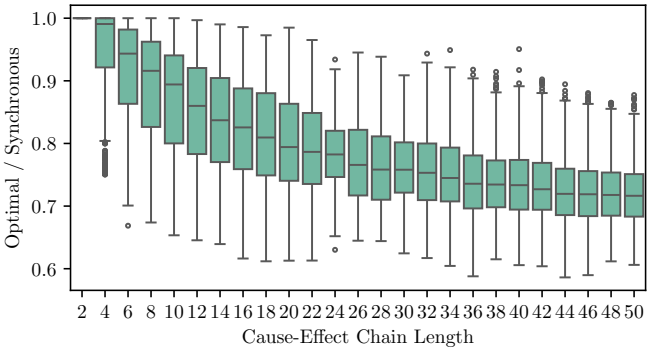


Figure 7. End-to-end latency with suggested phasing normalized to the latency under synchronous release for a varying chain length.

cause-effect chains, the computed latency is normalized to the hyperperiod of the cause-effect chain. For cause-effect chains with max-harmonic periods, the results of Section VI are applied. For cause-effect chains which are $(2, k)$-max-harmonic, the results of Section VII are applied.

1) Latency Evaluation: In the following, we compare the latency achieved with the optimal phasing to the synchronous release of all tasks. The number of tasks in the task chain is varied between 2 and 50 in steps of 2. Figure 7 shows the latency of the cause-effect chain under the proposed phasing normalized to the latency under synchronous release as boxplot. While for chains of length two, the synchronous case already constitutes the optimal phasing, a clear reduction of the end-to-end latency is observed if the cause-effect chain has a length of at least 3. The latency gradually decreases with the length of the cause-effect chain to a median latency of approximately 0.72 of the latency under synchronous release, at length 50. At the same chain length, the lowest achieved latency is 0.88, and the best observed latency is 0.61 of the latency under synchronous release. From this evaluation, we can conclude that an optimal configuration of task phases significantly reduces the end-to-end latency of cause-effect chains, especially with a large number of tasks in the chain.

2) Runtime Evaluation: In this experiment, we evaluate the runtime of our approach. To do so, we compare our

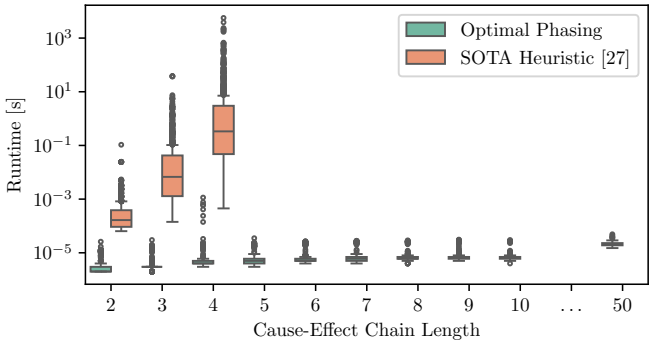


Figure 8. Runtime of the proposed approach (phase configuration and latency analysis) compared to the SOTA heuristic [28].

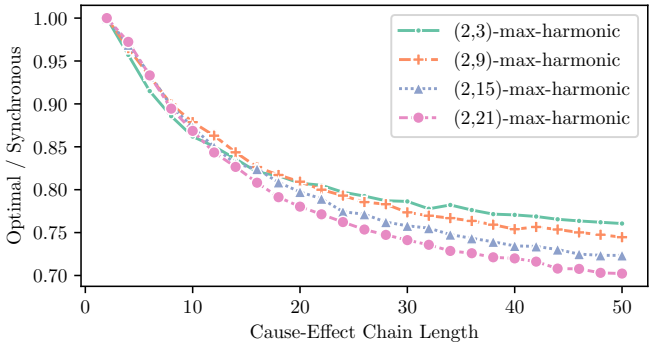


Figure 9. Geometric mean of the optimal end-to-end latency normalized to the end-to-end latency with synchronous release. Results are shown for task chains with (2,k)-max-harmonic periods, with different values of k.

proposed phasing and latency bound to the state-of-the-art heuristic by Martinez et al. [28]. In contrast to our approach, the heuristic does not have assumptions on the periods and can be used with any set of periods. To find the optimal assignment of task phases, the heuristic evaluates all non-equivalent phase assignments of the cause-effect chain. Since the time granularity for the optimization is not defined in [28], we choose the length of the shortest period, i.e., 1 ms. The chain length is varied between 2 and 10 in steps of 1.

Figure 8 shows the runtime of our approach (measured as the time to assign optimal offsets to chains plus the computation of the worst-case latency of the cause-effect chain) compared to the SOTA heuristic on a logarithmic scale. For cause-effect chains of two to four tasks, the heuristic can be executed to completion. For a length of four tasks, the runtime of the heuristic reached up to 3854 s (around 1 h). During this time, the heuristic evaluates up to 40 000 000 configurations. At a chain length of four, the runtime of our approach is between 2 µs and 19 µs (avrg. 4 µs.) For chains longer than five tasks, the runtime of the SOTA heuristic is to large for comparison. This is due to the number of non-equivalent phase configurations that are evaluated by the heuristic, which increases exponentially. For a chain length of ten, in average, ca. 10^{10} phase configurations must be evaluated by the heuristic to guarantee that the optimal phasing is found. The maximal observed runtime of our approach with cause-effect chains of length 10 is 30 µs. At a chain length of 50, this value increases to 49 µs.

The results show the benefit of our phasing compared to the SOTA heuristic, if periods are max-harmonic or $(2, k)$-max-harmonic. With its low runtime, our proposed approach can be used in design space exploration of chains of any length.

C. Synthetic Chains with (2,k)-Max-Harmonic Periods

In this section, we evaluate synthetic cause-effect chains with $(2, k)$-max-harmonic periods. Exhaustively, all possible sets of periods that are $(2, k)$-max-harmonic are generated, with the constraints that the maximum allowed period is 500 ms (i.e., leading to a maximum hyperperiod of 1 s) and a valid period set must include at least 5 different periods. For each individual cause-effect chain, a random $(2, k)$-max-harmonic period set is selected and used as the basis for the chain generation. If a generated chain has no $(2, k)$-max-harmonic periods, it is discarded and a new chain is generated.

The number of tasks in the task chain is varied between 2 and 50 in steps of 2, and results are reported for values of k between 3 and 21, in steps of 6 (note that only odd values of k are possible, as otherwise $T^E_{max,2}$ would divide $T^E_{max,1}$). For each configuration, 1000 random cause-effect chains are evaluated using the results of Section VII. With this experiment, the influence of k on the chain's latency is evaluated. Figure 9 shows the geometric mean of the latency with optimal phasing normalized to the latency with a synchronous task release. The results show that for each value of k, the achieved latency reduction is dependent on the chain length. It can also be seen that results are comparable until a chain length of 16. For larger chains, larger values of k lead to a smaller normalized data age, from 0.76 for $(2, 3)$-max-harmonic chains up to 0.70 for $(2, 21)$-max-harmonic chains.

IX. Conclusion

The end-to-end latency of cause-effect chains is a crucial requirement for automotive systems. To increase the determinism, the LET paradigm has received significant attention in this domain. In this paper, we address the configuration of task phases to minimize the end-to-end latency of cause-effect chains. We study the problem in two settings: (i) for cause-effect chains with max-harmonic periods, and (ii) for cause-effect chains with $(2, k)$-max-harmonic periods. In particular, this allows to apply our results to harmonic task sets, as well as to semi-harmonic automotive periods described in [26].

For both settings, we present a suggested task phasing that minimizes the end-to-end latency. We further show how to compute the end-to-end latency of the cause-effect chain under the suggested phasing and prove that it is optimal. Evaluations highlight the approach using an industrial use-case. Synthetic results compare the approach against the state of the art, showing that we require only a fraction of the runtime to find the optimal phasing. Future work will extend the approach to cause-effect chains with arbitrary periods, and examine the configuration of task phases to optimize the end-to-end latency along multiple cause-effect chains with shared tasks.

ACKNOWLEDGEMENT

This result is part of a project (PropRT) that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 865170). This work was partially supported by the Swedish Research Council (VR) under the project nr. 2023-04773.

REFERENCES

[1] AUOTOSAR Specification of Timing Extensions, AUTOSAR CP R22-11, 2022.

[2] B. Akesson, M. Nasri, G. Nelissen, S. Altmeyer, and R. I. Davis. A comprehensive survey of industry practice in real-time systems. *Real-time Systems*, 58(3):358–398, 2021.

[3] M. Becker, D. Dasari, S. Mubeen, M. Behnam, and T. Nolte. Mechaniser-a timing analysis and synthesis tool for multi-rate effect chains with job-level dependencies. In *International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems (WATERS)*, 2016.

[4] M. Becker, D. Dasari, S. Mubeen, M. Behnam, and T. Nolte. Synthesizing job-level dependencies for automotive multi-rate effect chains. In *International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pages 159–169, 2016.

[5] M. Becker, D. Dasari, S. Mubeen, M. Behnam, and T. Nolte. End-to-end timing analysis of cause-effect chains in automotive embedded systems. *Journal of Systems Architecture*, 80:104–113, 2017.

[6] D. Bellassai, A. Biondi, A. Biasci, and B. Morelli. Supporting logical execution time in multi-core POSIX systems. *Journal of Systems Architecture*, 144:102987, 2023.

[7] A. Benveniste, P. Caspi, P. L. Guernic, H. Marchand, J.-P. Talpin, and S. Tripakis. A protocol for loosely time-triggered architectures. In *EMSOFT*, pages 252–265, 2002.

[8] E. Bini, P. Pazzaglia, and M. Maggio. Zero-jitter chains of periodic let tasks via algebraic rings. *IEEE Transactions on Computers*, 72(11):3057–3071, 2023.

[9] A. Biondi and M. Di Natale. Achieving predictable multicore execution of automotive applications using the LET paradigm. In *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 240–250, 2018.

[10] A. Biondi, P. Pazzaglia, A. Balsini, and M. Di Natale. Logical execution time implementation and memory optimization issues in autosar applications for multicores. In *International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems (WATERS)*, 2017.

[11] T. Bourke, V. Bregeon, and M. Pouzet. Scheduling and Compiling Rate-Synchronous Programs with End-To-End Latency Constraints. In *35th Euromicro Conference on Real-Time Systems (ECRTS 2023)*, pages 1:1–1:22, 2023.

[12] A. Davare, Q. Zhu, M. D. Natale, C. Pinello, S. Kanajan, and A. L. Sangiovanni-Vincentelli. Period optimization for hard real-time distributed automotive systems. In *Design Automation Conference, DAC*, pages 278–283, 2007.

[13] M. Dürr, G. von der Brüggen, K.-H. Chen, and J.-J. Chen. End-to-end timing analysis of sporadic cause-effect chains in distributed systems. *ACM Trans. Embedded Comput. Syst. (Special Issue for CASES)*, 18(5s):58:1–58:24, 2019.

[14] N. Feiertag, K. Richter, J. Nordlander, and J. Jonsson. A compositional framework for end-to-end path delay calculation of automotive systems under different path semantics. In *Workshop on Compositional Theory and Technology for Real-Time Embedded Systems (CRTS)*, 2009.

[15] J. Forget, F. Boniol, and C. Pagetti. Verifying end-to-end real-time constraints on multi-periodic models. In *IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA)*, pages 1–8, 2017.

[16] K.-B. Gemlau, L. KÖHLER, R. Ernst, and S. Quinton. System-level logical execution time: Augmenting the logical execution time paradigm for distributed real-time automotive software. *ACM Trans. Cyber-Phys. Syst.*, 5(2), jan 2021.

[17] M. Günzel, K.-H. Chen, N. Ueter, G. von der Brüggen, M. Dürr, and J.-J. Chen. Timing analysis of asynchronized distributed cause-effect chains. In *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 40–52, 2021.

[18] M. Günzel, H. Teper, K. Chen, G. von der Brüggen, and J. Chen. On the equivalence of maximum reaction time and maximum data age for cause-effect chains. In *35th Euromicro Conference on Real-Time Systems (ECRTS)*, 2023.

[19] A. Hamann, D. Dasari, S. Kramer, M. Pressler, and F. Wurst. Communication Centric Design in Complex Automotive Embedded Systems. In *29th Euromicro Conference on Real-Time Systems (ECRTS 2017)*, pages 10:1–10:20, 2017.

[20] T. A. Henzinger, B. Horowitz, and C. M. Kirsch. Embedded control systems development with giotto. In *Proceedings of the ACM SIGPLAN workshop on Languages, compilers and tools for embedded systems*, pages 64–72, 2001.

[21] P. Iyenghar, L. Huning, and E. Pulvermueller. Automated end-to-end timing analysis of autosar-based causal event chains. In *15th International Conference on Evaluation of Novel Approaches to Software Engineering (ENASE)*, pages 477–489, 2020.

[22] C. M. Kirsch and A. Sokolova. The logical execution time paradigm. In *Advances in Real-Time Systems*, pages 103–120. Springer, 2012.

[23] T. Klaus, M. Becker, W. Schröder-Preikschat, and P. Ulbrich. Constrained data-age with job-level dependencies: How to reconcile tight bounds and overheads. In *IEEE 27th Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 66–79, 2021.

[24] T. Kloda, A. Bertout, and Y. Sorel. Latency analysis for data chains of real-time periodic tasks. In *IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 360–367, 2018.

[25] T. Kloda, A. Bertout, and Y. Sorel. Latency upper bound for data chains of real-time periodic tasks. *Journal of Systems Architecture*, 109:101824, 2020.

[26] S. Kramer, D. Ziegenbein, and A. Hamann. Real world automotive benchmarks for free. In *International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems (WATERS)*, 2015.

[27] L. Maia and G. Fohler. Reducing End-to-End Latencies of Multi-Rate Cause-Effect Chains in Safety Critical Embedded Systems. In *12th European Congress on Embedded Real Time Software and Systems (ERTS)*, 2024.

[28] J. Martinez, I. Sañudo, and M. Bertogna. Analytical characterization of end-to-end communication delays with logical execution time. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(11):2244–2254, 2018.

[29] F. Paladino, A. Biondi, E. Bini, and P. Pazzaglia. Optimizing Per-Core Priorities to Minimize End-To-End Latencies. In *36th Euromicro Conference on Real-Time Systems (ECRTS)*, pages 6:1–6:25, 2024.

[30] P. Pazzaglia, A. Biondi, and M. Di Natale. Optimizing the functional deployment on multicore platforms with logical execution time. In *IEEE Real-Time Systems Symposium (RTSS)*, pages 207–219, 2019.

[31] P. Pazzaglia, D. Casini, A. Biondi, and M. Di Natale. Optimizing inter-core communications under the LET paradigm using DMA engines. *IEEE Transactions on Computers*, 72(1):127–139, 2022.

[32] A. Rajeev, S. Mohalik, M. G. Dixit, D. B. Chokshi, and S. Ramesh. Schedulability and end-to-end latency in distributed ECU networks: formal modeling and precise estimation. In *International Conference on Embedded Software*, pages 129–138, 2010.

[33] S. Resmerita, A. Naderlinger, and S. Lukesch. Efficient realization of logical execution times in legacy embedded software. In *Proceedings of the 15th ACM-IEEE International Conference on Formal Methods and Models for System Design (MEMOCODE)*, page 36–45, 2017.

[34] J. Schlatow, M. Möstl, S. Tobuschat, T. Ishigooka, and R. Ernst. Data-age analysis and optimisation for cause-effect chains in automotive control systems. In *IEEE International Symposium on Industrial Embedded Systems (SIES)*, pages 1–9, 2018.

[35] S. Sinja and R. West. End-to-end scheduling of real-time task pipelines on multiprocessors. *Journal of Systems Research*, 1(2), 2022.

[36] Y. Tang, X. Jiang, N. Guan, D. Ji, X. Luo, and W. Yi. Comparing communication paradigms in cause-effect chains. *IEEE Transactions on Computers*, 72(1):82–96, 2023.

[37] S. Wang, D. Li, A. H. Sifat, S. Huang, X. Deng, C. Jung, R. K. Williams, and H. Zeng. Optimizing logical execution time model for both determinism and low latency. *CoRR*, abs/2310.19699, 2023.

[38] R. Xu, M. Kühl, H. Von Hasseln, and D. Nowotka. Reducing overall path latency in automotive logical execution time scheduling via reinforcement learning. In *31st International Conference on Real-Time Networks and Systems (RTNS)*, page 212–223, 2023.