This article has been accepted for publication in IEEE Transactions on Computers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TC.2023.3283707



# Zero-Jitter Chains of Periodic LET Tasks via Algebraic Rings

Enrico Bini*, Paolo Pazzaglia†, Martina Maggio‡
*University of Turin, Italy
†Robert Bosch GmbH, Germany
‡Saarland University, Germany & Lund University, Sweden

**Abstract**—In embedded computing domains, including the automotive industry, complex functionalities are split across multiple tasks that form *task chains*. These tasks are functionally dependent and communicate partial computations through shared memory slots based on the *Logical Execution Time* (LET) paradigm. This paper introduces a model that captures the behavior of a producer-consumer pair of tasks in a chain, characterizing the timing of reading and writing events. Using ring algebra, the combined behavior of the pair can be modeled as a single periodic task. The paper also presents a lightweight mechanism to eliminate jitter in an entire chain of any size, resulting in a single periodic LET task with zero jitter. All presented methods are available in a public repository.

## 1 Introduction

Modern commercial real-time applications, especially in the automotive sector, rely on multiple tasks working in sequence to implement complex functionalities such as driving assistance systems and advanced safety functions. These systems require integrating a higher number of sensors and actuators, as well as complex software to process the data. For example, a vision-control pipeline [20], [29] uses multiple sensors to collect data. The data is processed and passed to a vision algorithm that extracts environmental features. From this output, a control algorithm determines the appropriate actions for safe autonomous driving.

On the programmer side, this complexity manifests in the challenge of integrating a large number of tasks exchanging data. These tasks usually run periodically. Possibly, they designed by different teams or suppliers with mismatched rates [18], and communicate through shared memory. To improve determinism, communication mechanisms such as the *Logical Execution Time* (LET) [15] are currently a common choice. The functional dependencies between LET tasks, driven by the pipeline of data, creates so-called *functional task chains* [14]. When the activation of each task in the chain is driven only by its period, these chains are often referred to as *time-triggered* chains.

The high relevance of this application domain has put a great pressure on the research community for developing ready-to-use analysis tools, such as pyCPA [27], that allow to obtain safe estimates of metrics of interest, such as the input-output latency of the chain. This kind of analysis is computationally not very complex and fairly straightforward to implement. Hence, the effort dedicated to analyze in greater depth the details of timing of write and read operations has been limited to the simulation of the chain behavior, and the extraction of safe bounds that give worst-case guarantees on the data propagation inside the chain. Classical approaches [5] rely on the knowledge of a list of indices for each task, to identify those jobs in the chain that are strictly required to communicate data, in order to achieve the correct behavior. In reality, the behavior of time-triggered chains follows repetitive patterns, which can enumerated by anayzing all the jobs in the hyperperiod. This approach, however, suffers from scalability problems. Providing analytical expressions for said patterns would then enable the analysis of real-world workloads, without introducing approximations and costly listings.

In this paper, we provide such an analysis and offer the following contributions:

- Section 4 introduces a modular approach that combines the behavior of a pair of communicating LET tasks into a single task, providing a foundation for an iterative mechanism that combines multiple tasks.
- Section 5 uses ring algebras to analyze the temporal characteristics of a pair of communicating LET tasks in a task chain, providing a closed-form solution that answers questions such as the pattern of read or write phasings of the pair and the jobs with the longest and shortest input-output latency.
- Finally, Section 6, proposes a methodology to eliminate read/write jitter of task chains with arbitrary length, building upon the results obtained for task pairs. The resulting chain behaves as a periodic LET task, paving the way to more deterministic designs for task chains.

These methods are implemented in Python and available in the repository (https://github.com/ebni/periodic-LET).

These results are significant for designing and analyzing distributed applications. By ensuring that a chain of tasks with different periods and phasings behaves as a purely periodic system, designers can abstract the underlying chain complexity. Additionally, this approach can be used to split a periodic task into multiple distributed chained tasks, while maintaining the same functional behavior.

This article has been accepted for publication in IEEE Transactions on Computers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TC.2023.3283707



## 2 Related Work

Task chains are typically categorized in literature based on the triggering mechanism of their components, with the most common types being event-triggered and time-triggered chains [26]. In event-triggered chains, which are common in avionics [11] and robotics [6], each task is activated by its predecessor's termination . In contrast, time-triggered chains are activated based on their periods and are frequently used in automotive systems [1], following the AUTOSAR standard. In these chains, tasks communicate data through shared variables.

Among the first mentions of time-triggered task chains, Gerber et al. [10] proposed a method to assign periods so that functional links are preserved. More recently, the analysis for different end-to-end delays introduced in [9] identifies four different semantics for latency, including the popular reaction delay (*first-to-first*) and data age (*last-to-last*). Many other analyses have been proposed [2], [3], [8], [17], [19], [21], targeting the measure of the worst-case end-to-end input-output latency or the maximum data age, as this is a critical parameter in many application domains. Other research works on task chains addressed the problem of optimizing task parameters to satisfy end-to-end timing constraints [7], [28], [31]. Recent papers targeted more complex scenarios: chains that may share one or more tasks with other chains [16], and globally asynchronous locally synchronous distributed chains [13].

In this paper, we consider chains where tasks communicate at fixed points in time, to ensure determinism and predictability. A popular way to achieve these goals is to implement a communication scheme following the so-called *Logical Execution Time* (LET) paradigm [15]. This can be effectively implemented in a real-time embedded platform by introducing multiple memory slots for shared data, such as double- [4] or triple-buffering [22] with pointer switching, or alternatively with the use of intermediate global labels and a dedicated update task that performs the copies and runs with the highest priority [5]. The LET copies can also be performed with the usage of DMA engines [23], [24].

Recent works started addressing time-triggered task chains communicating with the LET paradigm, comparing their timing properties with respect to other communication mechanisms, e.g., in [2], [21], [22], [30]. Despite being beneficial for time determinism, the usage of a LET-based communication introduces higher (but constant) end-to-end latencies in chains [2], [12]. Furthermore, the effects of deadline misses in such systems has been studied in [25].

This paper offers a complete and analytic solution for a classic time-triggered task chain under the LET paradigm. The paper begins by examining a pair of functionally-dependent tasks and characterizing the sequence of read/write phasings through algebraic rings, resulting in a repeatable pattern described analytically. This pattern is the first step towards a compositional analysis of tasks and leads to proposing a new design that eliminates the jitter of the time-triggered chain with minimal intervention. Notably, this is the first work that enhances a task chain to conform to the model of a single task following the Logical Execution Time paradigm.

## 3 System Model and Background

In this section, we model the communication between tasks through shared memory and we apply it to a single periodic task (Section 3.1). We also introduce the background needed to understand the technical content (Section 3.2).

### 3.1 Model of tasks and communication

We denote a task with $\tau_i$. Each task $\tau_i$ releases an infinite sequence of jobs. $\mathbb{J}_i$ denotes the indices of all jobs of task $\tau_i$. We assume that $\mathbb{J}_i$ is a discrete, infinite, and totally ordered set such as the set of signed integers $\mathbb{Z}$. For any $j \in \mathbb{J}_i$ we denote by $\mathrm{next}(j) \in \mathbb{J}_i$ the index of the following job of $\tau_i$, and by $\mathrm{prev}(j)$ the preceding one. When $\mathbb{J}_i = \mathbb{Z}$, we have $\mathrm{next}(j) = j+1$ and $\mathrm{prev}(j) = j-1$. We keep, however, the more general notations of $\mathrm{next}(j)$ and $\mathrm{prev}(j)$, to have them meaningful for other sets of jobs introduced in the paper. Also, as $\mathbb{Z}$ allows for negative indexes, we have no notion of "first" job released by a task. Nonetheless, if needed, such a first job can be identified as the one indexed by $j = 0$.

Each job executes a stateless computation, i.e., it does not use its internal state but only the input data to calculate its output. This is for example the case with image processing. Specifically, each job $j \in \mathbb{J}_i$ of $\tau_i$ performs the following operations, in this order:

1) it reads its input data from a shared memory location at the *read instant* $\mathsf{rd}_i(j)$;
2) it performs all its operations; and
3) it writes its output data in a shared memory location at the *write instant* $\mathsf{wr}_i(j)$, making it available to others.

In this paper, we specifically target a wide class of *periodic* tasks, with a generic representation as follows.

**Definition 1** (Periodic task)**.** *A periodic task $\tau_i$ is characterized by the following quantities:*

- *$T_i$ determines the periodicity of the released jobs;*
- *$\theta^{\mathsf{r}}_i(j)$ is the phasing of read instant of the job $j$ of $\tau_i$; and*
- *$\theta^{\mathsf{w}}_i(j)$ is the phasing of write instant of the job $j$ of $\tau_i$,*

*so that the read and write instant of the released jobs are*

$$\begin{cases} \mathsf{rd}_i(j) = j\,T_i + \theta^{\mathsf{r}}_i(j) \\ \mathsf{wr}_i(j) = j\,T_i + \theta^{\mathsf{w}}_i(j). \end{cases} \tag{1}$$

*and the phasings $\theta^{\mathsf{r}}_i(j)$, $\theta^{\mathsf{w}}_i(j)$ are both lower and upper bounded.*

The phasings $\theta^{\mathsf{r}}_i(j)$ and $\theta^{\mathsf{w}}_i(j)$ may encode different information, such as the relative release offset between tasks (e.g., in the case of asynchronous periodic tasks) and their deadlines. In the following, we abstract from the actual execution time of the tasks, and we focus on the two instants $\mathsf{rd}_i(j)$ and $\mathsf{wr}_i(j)$, which occur respectively before starting the execution of the job and after the completion of the job execution.

To ensure causality, $\forall i, j$ we have:

$$\begin{cases} \mathsf{rd}_i(\mathrm{prev}(j)) \leqslant \mathsf{rd}_i(j)\,, \\ \mathsf{rd}_i(j) \leqslant \mathsf{wr}_i(j)\,, \\ \mathsf{wr}_i(j) \leqslant \mathsf{wr}_i(\mathrm{next}(j))\,. \end{cases}$$

Also, we define the separation between two consecutive read instants and the separation between two consecutive write instants, as follows

$$\begin{aligned} \Delta\mathsf{rd}_i(j) &= \mathsf{rd}_i(\mathrm{next}(j)) - \mathsf{rd}_i(j) \\ \Delta\mathsf{wr}_i(j) &= \mathsf{wr}_i(\mathrm{next}(j)) - \mathsf{wr}_i(j)\,. \end{aligned} \tag{2}$$

This article has been accepted for publication in IEEE Transactions on Computers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TC.2023.3283707



Finally, we show that the values introduced above can be easily mapped to the commonly used concept of *input-output latency*, which can be defined for the job $j \in \mathbb{J}_i$ of $\tau_i$ by

$$L_i(j) = \mathsf{wr}_i(j) - \mathsf{rd}_i(j) = \theta_i^{\mathsf{w}}(j) - \theta_i^{\mathsf{r}}(j).$$

In the case under analysis, we consider that *LET communication* is enforced between the periodic tasks in a chain.

**Definition 2** (Periodic LET task)**.** *A periodic LET task* $\tau_i$ *is a periodic task with constant phasings* $\theta_i^{\mathsf{r}}(j)$ *and* $\theta_i^{\mathsf{w}}(j)$ *.*

For these tasks, we can drop the job index $j$ and we use the more compact notations $\theta_i^{\mathsf{r}}$, $\theta_i^{\mathsf{w}}$, and $L_i = \theta_i^{\mathsf{w}} - \theta_i^{\mathsf{r}}$ for the phasings and the input-output latency, respectively.

### 3.2 Mathematical Background

We introduce two mathematical operators to write the equations in a compact form, extending the modulo and greatest common divisor operators over the real numbers.

**Definition 3** (Modulo operator over the real numbers)**.** *Given* $x \in \mathbb{R}$ *and* $m \in \mathbb{R}$*, with* $m > 0$*, we define*

$$\lfloor x \rceil_m = x - \left\lfloor \frac{x}{m} \right\rfloor m. \tag{3}$$

From the definition of $\lfloor x \rceil_m$ the following properties follow:

$$0 \leqslant \lfloor x \rceil_m < m \tag{4}$$
$$\forall\, k \in \mathbb{Z},\ \lfloor x + k\, m \rceil_m = \lfloor x \rceil_m \tag{5}$$
$$0 \leqslant x < m \Leftrightarrow x = \lfloor x \rceil_m \tag{6}$$

The next property is useful to relate the ceiling to the modulo operator. In fact, we have

$$\lfloor -x \rceil_m \overset{\text{from (5)}}{=} \lfloor \lceil x/m \rceil m - x \rceil_m \overset{\text{from (6)}}{=} \lceil x/m \rceil m - x. \tag{7}$$

Whereas the next property says that inside a modulo operator, we can re-apply the operator to the addenda, if any. In fact, from

$$\lfloor x \rceil_m + \lfloor y \rceil_m = x + y - \left( \left\lfloor \frac{x}{m} \right\rfloor + \left\lfloor \frac{y}{m} \right\rfloor \right) m,$$

it follows

$$\begin{aligned}\lfloor \lfloor x \rceil_m + \lfloor y \rceil_m \rceil_m &= x + y - \left( \left\lfloor \frac{x}{m} \right\rfloor + \left\lfloor \frac{y}{m} \right\rfloor \right) m \\ &\quad - \left\lfloor \frac{x + y - \left( \left\lfloor \frac{x}{m} \right\rfloor + \left\lfloor \frac{y}{m} \right\rfloor \right) m}{m} \right\rfloor m \\ &= x + y - \left\lfloor \frac{x+y}{m} \right\rfloor m = \lfloor x + y \rceil_m\,, \end{aligned} \tag{8}$$

and with similar steps, we can show that

$$\lfloor - \lfloor x \rceil_m \rceil_m = \lfloor -x \rceil_m\,. \tag{9}$$

Finally, the next property is useful to transform from a modulo to another one. If $x = kd$ for any $k$ and $d$, then

$$\lfloor x \rceil_m = k\, d - \left\lfloor \frac{k\, d}{m} \right\rfloor m = \left( k - \left\lfloor \frac{k}{m/d} \right\rfloor \frac{m}{d} \right) d = \lfloor k \rceil_{m/d}\, d. \tag{10}$$

We remark that if $x \in \mathbb{Z}$ and $m \in \mathbb{N}$, then any $\lfloor x \rceil_m$ belongs to the *commutative ring* $\mathbb{Z}/m\mathbb{Z}$, which consists of a set equipped with addition and multiplication, enjoying then all the properties of such an algebraic structure. In such a case $\lfloor x \rceil_m$ is equivalent to modulo $m$ congruences, that is,

$$\lfloor x \rceil_m \equiv x \mod m.$$

Also, we can tighten (4) over $\mathbb{Z}/m\mathbb{Z}$ by

$$x \in \mathbb{Z}, m \in \mathbb{N}, \quad \Rightarrow \quad 0 \leqslant \lfloor x \rceil_m \leqslant m - 1. \tag{11}$$

We remind the reader some properties over the ring $\mathbb{Z}/m\mathbb{Z}$. If $\gcd(p, m) = 1$, then $p$ has the multiplicative inverse over $\mathbb{Z}/m\mathbb{Z}$. For example, over $\mathbb{Z}/5\mathbb{Z}$

$$2^{-1} \equiv 3 \mod 5, \quad 4^{-1} \equiv 4 \mod 5.$$

**Definition 4** (Greatest common divisor over the real numbers)**.** *Given* $x, y \in \mathbb{R}$, $x, y > 0$,

- *if* $\exists\, G \in \mathbb{R}$ *and* $\exists\, p_1, p_2 \in \mathbb{N}$ *such that* $x = p_1\, G$, $y = p_2\, G$ *and* $\gcd(p_1, p_2) = 1$, *we write* $\gcd(x, y) = G$,
- *if* $\nexists\, G$ *as above, then we write* $\gcd(x, y) = 0$.

Notice that $G$ is unique. In fact, say that $G'$ is another value such that $x = p_1' G'$, $y = p_2' G'$, and $\gcd(p_1', p_2') = 1$, for some $p_1', p_2' \in \mathbb{N}$. Then, $p_i = p_i' G'/G$ from which $1 = \gcd(p_1, p_2) = \gcd(p_1', p_2') G'/G = G'/G$ meaning that $G = G'$.

For example, $\gcd(3/4, 5/6) = 1/12$, $\gcd(5\sqrt{3}, 2\sqrt{3}) = \sqrt{3}$, and $\gcd(1, \pi) = 0$. We remark that over the integers, such a definition coincide with the standard $\gcd(\cdot, \cdot)$ definition of greatest common divisor.

## 4 Preliminaries

The first goal of this paper is to derive a new model that enables to treat in an unified formulation both a single periodic LET task and the composition of multiple periodic LET tasks in a time-triggered task chain (referred to simply as *chain* in the remaining of the paper). Here, we introduce a convenient notation so that $\tau_i$ can be referred to "task" and "chain" interchangeably.

### 4.1 Periodic chains

We will be using the aggregate index $i \blacktriangleright \ell$ to denote the chain $\tau_{i\blacktriangleright\ell}$ in which $\tau_\ell$ reads the output written by $\tau_i$. Following the same spirit, we use the notation of $\mathbb{J}_{i\blacktriangleright\ell}$ for the set of jobs of the chain $\tau_{i\blacktriangleright\ell}$. Coalescing two tasks allows us to compose chains as needed. For example, the chain $\tau_{1\blacktriangleright2\blacktriangleright3}$ is the composition of the two "chains" $\tau_1$ and $\tau_{2\blacktriangleright3}$, which in turn is the composition of $\tau_2$ and $\tau_3$. Alternatively, $\tau_{1\blacktriangleright2\blacktriangleright3}$ can be seen as the composition of $\tau_{1\blacktriangleright2}$ and $\tau_3$.

As tasks are characterized by releasing jobs that process inputs to outputs, a similar definition can then be introduced for chains. We are now aiming at defining the set $\mathbb{J}_{i\blacktriangleright\ell}$ of jobs of the chain $\tau_{i\blacktriangleright\ell}$. Intuitively, a job $(j_i, j_\ell)$ of the chain $\tau_{i\blacktriangleright\ell}$ belongs to $\mathbb{J}_{i\blacktriangleright\ell} \subset \mathbb{J}_i \times \mathbb{J}_\ell$ if and only if the job $j_\ell \in \mathbb{J}_\ell$ is the first one to read the data written by the job $j_i \in \mathbb{J}_i$. Figure 1 illustrates this concept. Using this notation, the chain can be characterized in a similar way with respect to the task model of Definition 1.

**Definition 5** (Periodic chain)**.** *A periodic chain* $\tau_{i\blacktriangleright\ell}$ *is characterized by the following quantities,*

- $T_{i\blacktriangleright\ell}$ *determines the periodicity of the chain,*
- $\theta^{\mathsf{r}}_{i\blacktriangleright\ell}(j_i, j_\ell)$ *is the phasing of read instant of job* $(j_i, j_\ell) \in \mathbb{J}_{i\blacktriangleright\ell}$*, and*

- *$\theta^{\mathsf{w}}_{i\blacktriangleright\ell}(j_i, j_\ell)$ is the phasing of write instant of the same job.*

**Remark 1.** *A periodic task can be modeled as a periodic chain with one task only. Additionally, a periodic chain that includes several periodic tasks can be abstracted as a single periodic task.*

The procedure to define a job $(j_i, j_\ell)$ of the chain $\tau_{i\blacktriangleright\ell}$ requires more efforts and is presented next. First, we partition the jobs of $\mathbb{J}_i$ in the family of subsets $\{\mathbb{J}_i(j_\ell)\}_{j_\ell\in\mathbb{J}_\ell}$. Every $\mathbb{J}_i(j_\ell)$ contains only those jobs in $\mathbb{J}_i$ with write instants between $\mathsf{rd}_\ell(\mathrm{prev}(j_\ell))$ and $\mathsf{rd}_\ell(j_\ell)$, formally:

$$\mathbb{J}_i(j_\ell) = \{j_i \in \mathbb{J}_i : \mathsf{rd}_\ell(\mathrm{prev}(j_\ell)) < \mathsf{wr}_i(j_i) \leqslant \mathsf{rd}_\ell(j_\ell)\}. \quad (12)$$

Analogously, we can also define the set $\mathbb{J}_\ell(j_i)$ of the indices of $\tau_\ell$ jobs with read time between $\mathsf{wr}_i(j_i)$ and $\mathsf{wr}_i(\mathrm{next}(j_i))$

$$\mathbb{J}_\ell(j_i) = \{j_\ell \in \mathbb{J}_\ell : \mathsf{wr}_i(j_i) \leqslant \mathsf{rd}_\ell(j_\ell) < \mathsf{wr}_i(\mathrm{next}(j_i))\}, \quad (13)$$

also illustrated in Figure 1. We remark that both $\mathbb{J}_i(j_\ell)$ and $\mathbb{J}_\ell(j_i)$

- may be empty for some $j_\ell$ or some $j_i$ (for example, this is the case of $\mathbb{J}_1(6)$ or $\mathbb{J}_2(8)$ of Figure 1), and
- if not empty, they have a finite number of elements, since the two sets are bounded respectively by $\mathsf{rd}_\ell(j_\ell)$ and $\mathsf{wr}_i(\mathrm{next}(j_i))$.

From the set of Equation (12) we can now define the set $\mathbb{J}'_{i\blacktriangleright\ell}$ of jobs of the chain $\tau_{i\blacktriangleright\ell}$ as follows

$$\mathbb{J}'_{i\blacktriangleright\ell} = \{(j_i, j_\ell) \in \mathbb{J}_i \times \mathbb{J}_\ell : \mathbb{J}_i(j_\ell) \neq \varnothing, j_i = \max \mathbb{J}_i(j_\ell)\}, \quad (14)$$

and a "dual" definition from Equation (13) as

$$\mathbb{J}''_{i\blacktriangleright\ell} = \{(j_i, j_\ell) \in \mathbb{J}_i \times \mathbb{J}_\ell : \mathbb{J}_\ell(j_i) \neq \varnothing, j_\ell = \min \mathbb{J}_\ell(j_i)\}. \quad (15)$$

Next, we prove that $\mathbb{J}'_{i\blacktriangleright\ell}$ and $\mathbb{J}''_{i\blacktriangleright\ell}$ are equivalent, and this enables a well-posed definition of the set $\mathbb{J}_{i\blacktriangleright\ell}$ of jobs of $\tau_{i\blacktriangleright\ell}$.

**Lemma 1.** The two sets of jobs $\mathbb{J}'_{i\blacktriangleright\ell}$ and $\mathbb{J}''_{i\blacktriangleright\ell}$ are equivalent.

*Proof.* Let $(j'_i, j'_\ell)$ be in $\mathbb{J}'_{i\blacktriangleright\ell}$. This implies that $j'_i \in \mathbb{J}_i(j'_\ell)$ and then

$$\mathsf{wr}_i\big(j'_i\big) \leqslant \mathsf{rd}_\ell\big(j'_\ell\big)\,.$$

Also, it must necessarily be

$$\mathsf{rd}_\ell\big(j'_\ell\big) < \mathsf{wr}_i\big(\mathrm{next}(j'_i)\big)$$

otherwise $\mathrm{next}(j'_i) \in \mathbb{J}_i(j'_\ell)$, which cannot be the case because $j'_i = \max \mathbb{J}_i(j'_\ell)$. Putting the pieces together

$$\mathsf{wr}_i\big(j'_i\big) \leqslant \mathsf{rd}_\ell\big(j'_\ell\big) < \mathsf{wr}_i\big(\mathrm{next}(j'_i)\big)\,,$$

which implies that $j'_\ell \in \mathbb{J}_\ell(j'_i)$. Moreover, from the hypothesis of $j'_i \in \mathbb{J}_i(j'_\ell)$ we have

$$\mathsf{rd}_\ell\big(\mathrm{prev}(j'_\ell)\big) < \mathsf{wr}_i\big(j'_i\big)$$

implying that $\mathrm{prev}(j'_\ell)$ and all jobs earlier than $\mathrm{prev}(j'_\ell)$ do not belong to $\mathbb{J}_\ell(j'_i)$. This means that $j'_\ell = \min \mathbb{J}_\ell(j'_i)$ and then that $(j'_i, j'_\ell) \in \mathbb{J}''_{i\blacktriangleright\ell}$. The dual proof that $(j''_i, j''_\ell) \in \mathbb{J}''_{i\blacktriangleright\ell} \Rightarrow (j''_i, j''_\ell) \in \mathbb{J}'_{i\blacktriangleright\ell}$ can be carried on in a similar way and is omitted here for the sake of brevity. □

We can finally define the set $\mathbb{J}_{i\blacktriangleright\ell}$ of jobs of the chain.

**Definition 6.** *We define the set $\mathbb{J}_{i\blacktriangleright\ell}$ of jobs of the chain $\tau_{i\blacktriangleright\ell}$ by $\mathbb{J}'_{i\blacktriangleright\ell}$ of (14) or, equivalently, by $\mathbb{J}''_{i\blacktriangleright\ell}$ of (15).*

The set $\mathbb{J}_{i\blacktriangleright\ell}$ is discrete, infinite, and totally ordered because it inherits these properties from $\mathbb{J}_\ell$. Hence, for any

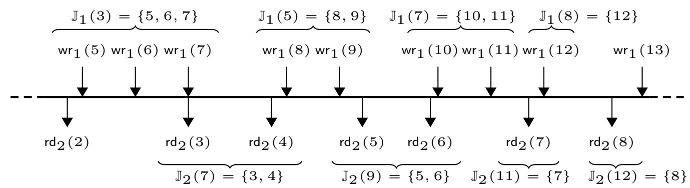


Fig. 1. An example of write and read instants of the chain $\tau_{1\blacktriangleright2}$. In this scenario, the pairs of jobs $\{(7, 3), (9, 5), (11, 7), (12, 8)\}$ belong to $\mathbb{J}_{1\blacktriangleright2}$. Also, the partition $\{\mathbb{J}_1(j_2)\}_{j_2\in\mathbb{J}_2}$ of the jobs in $\mathbb{J}_1$, and the partition $\{\mathbb{J}_2(j_1)\}_{j_1\in\mathbb{J}_1}$ of jobs in $\mathbb{J}_2$ defined in Equations (12) and (13) respectively, are illustrated (empty sets of the partitions such as $\mathbb{J}_1(4)$ or $\mathbb{J}_2(9)$, are not reported).

$(j_i, j_\ell) \in \mathbb{J}_{i\blacktriangleright\ell}$, the pair of jobs $\mathrm{next}(j_i, j_\ell) \in \mathbb{J}_{i\blacktriangleright\ell}$ is the next element in $\mathbb{J}_{i\blacktriangleright\ell}$ following $(j_i, j_\ell)$ in the ordering "$<$" induced by $\mathbb{J}_\ell$. As an example, in the scenario of read and write instants of Figure 1, $\mathrm{next}(9, 5) = (11, 7)$.

For every $(j_i, j_\ell) \in \mathbb{J}_{i\blacktriangleright\ell}$, the read and write instants are defined, not surprisingly, by

$$\begin{cases} \mathsf{rd}_{i\blacktriangleright\ell}(j_i, j_\ell) = \mathsf{rd}_i(j_i) \\ \mathsf{wr}_{i\blacktriangleright\ell}(j_i, j_\ell) = \mathsf{wr}_\ell(j_\ell) \end{cases}. \quad (16)$$

A particular case, which will be of interest in Section 6, is when the periodic chain has both constant read and write phasings, indeed behaving accordingly to the Logical Execution Time.

**Definition 7** (Periodic LET chain)**.** *A* periodic LET chain $\tau_{i\blacktriangleright\ell}$ *is a periodic chain with constant phasings $\theta^{\mathsf{r}}_{i\blacktriangleright\ell}$ and $\theta^{\mathsf{w}}_{i\blacktriangleright\ell}$.*

**Remark 2.** *A periodic LET task can be modeled as a periodic LET chain composed of one task only. Additionally, a periodic LET chain formed of multiple LET tasks can be abstracted as a single periodic LET task.*

## 4.2 Void and redundant jobs

We are now establishing a link between our notation and terminology with accepted definitions and concepts in the field. The definitions of the sets $\mathbb{J}_i(j_\ell)$ and $\mathbb{J}_\ell(j_i)$ are strictly related to the classic concepts of reachability [9], in particular with the definitions of immediate forward and backward job chains [8].

An immediate forward job chain is constructed starting from any job of the first task of a chain, and selecting the first job of the successive task with read instant occurring after the write instant of that job, in an iterative way. In our notation, the jobs $(j_1, j_2, \ldots, j_n)$ are an immediate forward job chain of the task chain $\tau_{1\blacktriangleright2\blacktriangleright\cdots\blacktriangleright n}$ when

$$\forall i = 1, \ldots, n-1, \qquad (j_i, j_{i+1}) \in \mathbb{J}''_{i\blacktriangleright i+1}.$$

Analogously, an immediate backward job chain is constructed starting from any job of the final task of the chain, and including the last job of the previous task writing no later than the read instant of that job, in a backward iterative manner. In our notation, the jobs $(j_1, j_2, \ldots, j_n)$ are an immediate backward job chain when

$$\forall i = 1, \ldots, n-1, \qquad (j_i, j_{i+1}) \in \mathbb{J}'_{i\blacktriangleright i+1}.$$

In literature, immediate forward job chains are used to measure how much time is required for an event sensed by the first task to reach the end of the chain. The longest input-output latency among immediate forward job chains corresponds to the maximum *reaction time* [9] of the chain. On the other hand, an immediate backward job chain represents a freshness value of the data used by the last task of the chain, and the longest input-output latency among immediate backward job chain corresponds to the maximum *data age*.

In general, these state-of-the-art definitions however abstract from the possible overwriting of labels due to oversampling/undersampling between tasks. Indeed, an immediate forward job chain may include jobs whose outputs are overwritten before being consumed by the next task, while an immediate backward job chain may include jobs that re-computed the same outputs of a previous job of the same task. To better highlight this aspects, we introduce the definitions of *void* and *redundant* jobs [1], [25].

A job $j_i$ of $\tau_i$ is void w.r.t. $\tau_\ell$ if it does not propagate data, i.e. the output of $j_i$ is overwritten by job $\mathrm{next}(j_i)$ before being read by any job of $\tau_\ell$.

**Definition 8** (Void job). *Given a chain $\tau_{i\blacktriangleright\ell}$, a job $j_i \in \mathbb{J}_i$ is* void *w.r.t. $\tau_\ell$ if $\nexists j_\ell \in \mathbb{J}_\ell$ such that $(j_i, j_\ell) \in \mathbb{J}_{i\blacktriangleright\ell}$.*

As an example, in Figure 1 job $j_1 = 8$ is void w.r.t. $\tau_2$, as its data is overwritten by job 9 of $\tau_1$. In fact, $(8, 5) \notin \mathbb{J}_{1\blacktriangleright2}$.

Instead, a job $j_\ell$ of $\tau_\ell$ is redundant w.r.t. $\tau_i$ if it repeats the same data processing, i.e. job $j_\ell$ reads the same input read by job $\mathrm{prev}(j_\ell)$ and thus produces the same output.

**Definition 9** (Redundant job). *Given a chain $\tau_{i\blacktriangleright\ell}$, a job $j_\ell \in \mathbb{J}_\ell$ is* redundant *w.r.t. $\tau_i$ if $\nexists j_i \in \mathbb{J}_i$ such that $(j_i, j_\ell) \in \mathbb{J}_{i\blacktriangleright\ell}$.*

Looking again at Figure 1, job $j_2 = 6$ is redundant w.r.t. $\tau_1$, as it reads the same input of the previous one, thus performing a redundant computation with respect to $(9, 5) \in \mathbb{J}_{1\blacktriangleright2}$.

## 5 Pair of Periodic LET Tasks

This section analyzes a task chain consisting of a pair of periodic LET tasks with constant read and write phasings. The chain is denoted as $\tau_{1\blacktriangleright2}$, where task $\tau_2$ reads data written by task $\tau_1$. The goal is to demonstrate that $\tau_{1\blacktriangleright2}$ is periodic with a period of $\max\{T_1, T_2\}$ and to provide a closed formulation of the read and write phases, enabling the identification of the job index for any given phase value. This formulation is also more flexible than classic approaches, e.g. in [5], which requires an iterative check to identify a task-by-task list of job indices of necessary communications (with no easily available information about how these job indices are correlated in the chain) and that is formulated only for tasks with synchronous release. Two examples are introduced in sections 5.1.1 and 5.2.1 to illustrate the proposed approach.

The definition of the set $\mathbb{J}_{1\blacktriangleright2}$ of jobs and the per-job phases $\theta^{\mathsf{r}}_{1\blacktriangleright2}(j)$ and $\theta^{\mathsf{w}}_{1\blacktriangleright2}(j)$ of any job $j \in \mathbb{J}_{1\blacktriangleright2}$ of the chain $\tau_{1\blacktriangleright2}$ requires an in-depth analysis. Such analysis is divided between the two cases $T_1 \geqslant T_2$ and $T_2 \geqslant T_1$. Also, it is convenient to define

$$\Theta = \theta^{\mathsf{r}}_2 - \theta^{\mathsf{w}}_1, \tag{17}$$

whose physical interpretation is the distance between the write instant of the job 0 of $\tau_1$ and the read instant of the job 0 of $\tau_2$.

### 5.1 Case $T_1 \geqslant T_2$

When $T_1 \geqslant T_2$, there is always a read instant $\mathsf{rd}_2(j)$ between any pair of consecutive write instants of $\tau_1$, formally

$$\forall j_1 \in \mathbb{J}_1, \quad \mathbb{J}_2(j_1) \neq \varnothing,$$

with $\mathbb{J}_2(j_1)$ defined earlier by Eq. (13). It is then convenient to define the set $\mathbb{J}_{1\blacktriangleright2}$ of jobs of the chain $\tau_{1\blacktriangleright2}$ from Eq. (15), which is

$$\mathbb{J}_{1\blacktriangleright2} = \{(j_1, j_2) \in \mathbb{J}_1 \times \mathbb{J}_2 : j_2 = \min \mathbb{J}_2(j_1)\}.$$

From Eq. (13) and since $\mathbb{J}_2(j_1)$ is never empty, the index $j_2 \in \mathbb{J}_2$ above is

$$\begin{aligned} j_2 &= \min\{j \in \mathbb{J}_2 : \mathsf{wr}_1(j_1) \leqslant \mathsf{rd}_2(j)\} \\ &= \min\{j \in \mathbb{J}_2 : j_1 T_1 + \theta^{\mathsf{w}}_1 \leqslant j T_2 + \theta^{\mathsf{r}}_2\} \\ &= \left\lceil \frac{j_1 T_1 - \Theta}{T_2} \right\rceil \end{aligned} \tag{18}$$

by using the definition of $\Theta$ of Equation (17). Hence, the set of jobs $\mathbb{J}_{1\blacktriangleright2}$ of the chain $\tau_{1\blacktriangleright2}$ is

$$\mathbb{J}_{1\blacktriangleright2} = \left\{ \left( j_1, \left\lceil \frac{j_1 T_1 - \Theta}{T_2} \right\rceil \right) : j_1 \in \mathbb{J}_1 \right\}. \tag{19}$$

To highlight the fact that the job index $(j_1, j_2) \in \mathbb{J}_{1\blacktriangleright2}$ of the chain $\tau_{1\blacktriangleright2}$ depends on $j_1$ only since $j_2$ is a function of $j_1$ through (18), we will be denoting jobs of the chain by $(j_1, *)$.

We now aim at characterizing the chain $\tau_{1\blacktriangleright2}$ as a periodic one as in Eq. (1), with proper period $T_{1\blacktriangleright2}$ and phasings $\theta^{\mathsf{r}}_{1\blacktriangleright2}(j)$ and $\theta^{\mathsf{w}}_{1\blacktriangleright2}(j)$ to be found. From the definition of Eq. 16, the read instant of any job $(j_1, *) \in \mathbb{J}_{1\blacktriangleright2}$ is

$$\mathsf{rd}_{1\blacktriangleright2}(j_1, *) = \mathsf{rd}_1(j_1) = j_1 T_1 + \theta^{\mathsf{r}}_1,$$

which indicates already that:

1) the period of the chain $\tau_{1\blacktriangleright2}$ is necessarily

$$T_{1\blacktriangleright2} = T_1,$$

2) and that the read phasing of any job $(j_1, *) \in \mathbb{J}_{1\blacktriangleright2}$ of the chain is constant

$$\forall (j_1, *) \in \mathbb{J}_{1\blacktriangleright2}, \quad \theta^{\mathsf{r}}_{1\blacktriangleright2}(j_1, *) = \theta^{\mathsf{r}}_1.$$

The write phasing $\theta^{\mathsf{w}}_{1\blacktriangleright2}(j_1, *)$ requires more efforts.

The write instant of the job $(j_1, *) \in \mathbb{J}_{1\blacktriangleright2}$ of $\tau_{1\blacktriangleright2}$ is, by the definition of Eq. (16), equal to the write instant of the job of $\tau_2$, that satisfies the following

$$\begin{aligned} \mathsf{wr}_{1\blacktriangleright2}(j_1, *) &= \mathsf{wr}_2\left(\left\lceil \frac{j_1 T_1 - \Theta}{T_2} \right\rceil\right) \\ &= \left\lceil \frac{j_1 T_1 - \Theta}{T_2} \right\rceil T_2 + \theta^{\mathsf{w}}_2 \\ &= j_1 T_1 + \underbrace{\left\lceil \frac{j_1 T_1 - \Theta}{T_2} \right\rceil T_2 - j_1 T_1 + \theta^{\mathsf{w}}_2}_{\theta^{\mathsf{w}}_{1\blacktriangleright2}(j_1, *)} \end{aligned}$$

in which we write $\mathsf{wr}_{1\blacktriangleright2}(j_1, *)$ assuming it has a period of $T_1$, with a per-job writing phase $\theta^{\mathsf{w}}_{1\blacktriangleright2}(j_1, *)$.

By exploiting the notion of modulo of Definition 3, the expression of the writing phase $\theta^{\mathsf{w}}_{1\blacktriangleright2}(j_1, *)$ can be split between a constant value plus a varying but bounded term as follows

$$\begin{aligned}\theta^{\mathsf{w}}_{1\blacktriangleright 2}(j_1,*) &= \left\lceil \frac{j_1T_1-\Theta}{T_2}\right\rceil T_2 - j_1T_1 + \theta^{\mathsf{w}}_2 \\ &= \left\lceil \frac{j_1T_1-\Theta}{T_2}\right\rceil T_2 - (j_1T_1-\Theta) - \Theta + \theta^{\mathsf{w}}_2 \\ &\overset{\text{from (7)}}{=} \underbrace{\lfloor \Theta - j_1T_1\rfloor_{T_2}}_{\text{varying with } j_1} + \underbrace{\theta^{\mathsf{w}}_2 - \Theta}_{\text{constant}} \end{aligned} \tag{20}$$

The standard property of Eq. (4) of the modulo-$T_2$ yields the following lower and upper bounds for $\theta^{\mathsf{w}}_{1\blacktriangleright 2}(j_1,*)$

$$\theta^{\mathsf{w}}_2 - \Theta \leqslant \theta^{\mathsf{w}}_{1\blacktriangleright 2}(j_1,*) < \theta^{\mathsf{w}}_2 - \Theta + T_2. \tag{21}$$

If $\gcd(T_1,T_2) = 0$, which necessarily implies that any of two periods is irrational, then the bounds of (21) cannot be further tightened. If instead, $T_1$ and $T_2$ have a non-zero $\gcd(T_1,T_2)$, then

- the upper and lower bounds for $\theta^{\mathsf{w}}_{1\blacktriangleright 2}(j_1,*)$ can be made tight, and
- the expression of the write phasing $\theta^{\mathsf{w}}_{1\blacktriangleright 2}(j_1,*)$ can be inverted, that is, for any possible phasing value $\theta$, we can determine exactly all $j_1$ such that $\theta^{\mathsf{w}}_{1\blacktriangleright 2}(j_1,*) = \theta$, if they exist.

Both results are enabled by the next theorem which processes the per-job varying part $\lfloor \Theta - j_1T_1\rfloor_{T_2}$ of Equation (20).

**Theorem 1.** *Let us set* $G = \gcd(T_1,T_2)$*. If* $G > 0$*, then*

$$\forall j_1 \in \mathbb{Z},\ \lfloor \Theta - j_1T_1\rfloor_{T_2} = \lfloor \phi_1 - j_1p_1\rfloor_{p_2} G + \lfloor\Theta\rfloor_G \tag{22}$$

*with* $p_1, p_2$ *such that* $T_1 = p_1G$, $T_2 = p_2G$, *and* $\phi_1$ *integer quotient of the following division by* $G$

$$\lfloor\Theta\rfloor_{T_2} = \phi_1 G + \lfloor\Theta\rfloor_G\,. \tag{23}$$

*Proof.* As $j_1$ steps over $\mathbb{Z}$, the quantity $\lfloor \Theta - j_1T_1\rfloor_{T_2}$ of (22) jumps along "the ring" depicted in Figure 2 in which the $T_2$-modulo algebraic ring is represented.

Let us now proceed formally. First, we can certainly write

$$\lfloor \Theta - j_1T_1\rfloor_{T_2} = \left\lfloor \lfloor -j_1T_1\rfloor_{T_2} + \lfloor\Theta\rfloor_{T_2}\right\rfloor_{T_2},$$

because of the property of (8). Analyzing the first addendum, we find

$$\lfloor -j_1T_1\rfloor_{T_2} = \lfloor -j_1p_1G\rfloor_{p_2G} = \lfloor -j_1p_1\rfloor_{p_2} G$$

by applying the property of Equation (10). Then from the definition of $\phi_1$ of (23)

$$\lfloor \Theta - j_1T_1\rfloor_{T_2} = \left\lfloor (\lfloor -j_1p_1\rfloor_{p_2} + \phi_1)G + \lfloor\Theta\rfloor_G\right\rfloor_{T_2}.$$

The next and final steps are made to further simplify the right-hand-side of the equation above. It is certainly true that

$$0 \leqslant \lfloor -j_1p_1\rfloor_{p_2} + \phi_1 \leqslant 2p_2 - 1$$

because $0 \leqslant \lfloor -j_1p_1\rfloor_{p_2} \leqslant p_2 - 1$ and it is also $0 \leqslant \phi_1 \leqslant p_2 - 1$ from its definition of (23) and the fact that $T_2 = p_2G$. We consider two cases. If $0 \leqslant \lfloor -j_1p_1\rfloor_{p_2} + \phi_1 \leqslant p_2 - 1$, then

$$0 \leqslant (\lfloor -j_1p_1\rfloor_{p_2} + \phi_1)G \leqslant T_2 - G$$
$$0 \leqslant (\lfloor -j_1p_1\rfloor_{p_2} + \phi_1)G + \lfloor\Theta\rfloor_G < T_2$$

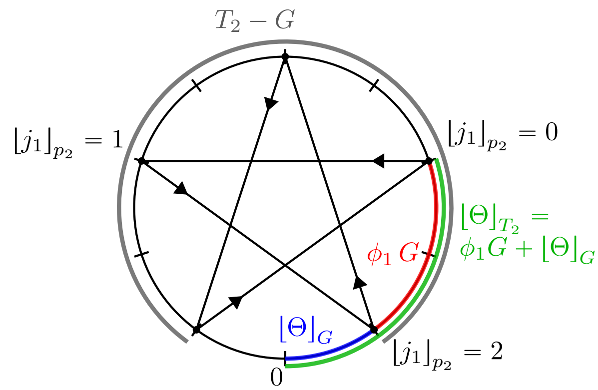


Fig. 2. Picturing the case with $T_1 = 16$, $\theta^{\mathsf{r}}_1 = 1$, $\theta^{\mathsf{w}}_1 = \theta^{\mathsf{r}}_1 + T_1 = 17$, and $T_2 = 10$, $\theta^{\mathsf{r}}_2 = 0$, $\theta^{\mathsf{w}}_2 = \theta^{\mathsf{r}}_2 + T_2 = 10$, fully explained later in Section 5.1.1. In such a case $\gcd(T_1,T_2) = G = 2$, $p_1 = 8$, and $p_2 = 5$. From these values and the definition of $\Theta$ of Eq. (17), we have $\Theta = \theta^{\mathsf{r}}_2 - \theta^{\mathsf{w}}_1 = -17$. When $j_1 = 0$ the value taken by $\lfloor \Theta - j_1T_1\rfloor_{T_2}$ is $\phi_1 G + \lfloor\Theta\rfloor_G = \lfloor\Theta\rfloor_{T_2} = \lfloor -17\rfloor_{10} = 3$. Such a quantity is represented by a green arc and both its addenda $\lfloor\Theta\rfloor_G$ and $\phi_1 G$ are represented by a blue and red arc, respectively. As $j_1$ increases, the term $\lfloor \Theta - j_1T_1\rfloor_{T_2}$ (and the write phasing $\theta^{\mathsf{w}}_{1\blacktriangleright 2}(j_1,*)$ too) varies by "jumping" $\lfloor -p_1\rfloor_{p_2} = \lfloor -8\rfloor_5 = 2$ vertices, as represented by the arrows through the vertices of the "star". As formally proven in Theorem 2, the minimal write phasing occurs at the closest vertex to the right of 0, i.e., when $\lfloor j_1\rfloor_{p_2} = 2$. Similarly, the maximum occurs when $\lfloor j_1\rfloor_{p_2} = 4$. As proven in Corollary 1, the maximum span of the write phasing $\theta^{\mathsf{w}}_{1\blacktriangleright 2}(j_1,*)$ is $T_2 - G$, illustrated by a gray arc.

and then we can remove the $\lfloor\cdots\rfloor_{T_2}$ operator by applying the property of Eq. (6)

$$\begin{aligned}\lfloor \Theta - j_1T_1\rfloor_{T_2} &= (\lfloor -j_1p_1\rfloor_{p_2} + \phi_1)G + \lfloor\Theta\rfloor_G \\ &= \lfloor -j_1p_1 + \phi_1\rfloor_{p_2} G + \lfloor\Theta\rfloor_G\,.\end{aligned}$$

If instead $p_2 \leqslant \lfloor -j_1p_1\rfloor_{p_2} + \phi_1 \leqslant 2p_2 - 1$, then

$$T_2 \leqslant (\lfloor -j_1p_1\rfloor_{p_2} + \phi_1)G \leqslant 2T_2 - G \quad \Rightarrow$$
$$\Rightarrow \quad 0 \leqslant (\lfloor -j_1p_1\rfloor_{p_2} + \phi_1)G + \lfloor\Theta\rfloor_G - T_2 < T_2$$

and then, by first applying the property (5) to subtract $-T_2$ and then the property (6) to remove the outer $\lfloor\cdots\rfloor_{T_2}$, we find

$$\begin{aligned}\lfloor \Theta - j_1T_1\rfloor_{T_2} &= \left\lfloor (\lfloor -j_1p_1\rfloor_{p_2} + \phi_1)G + \lfloor\Theta\rfloor_G - T_2\right\rfloor_{T_2} \\ &= (\lfloor -j_1p_1\rfloor_{p_2} + \phi_1 - p_2)G + \lfloor\Theta\rfloor_G \\ &= \lfloor -j_1p_1 + \phi_1\rfloor_{p_2} G + \lfloor\Theta\rfloor_G\,,\end{aligned}$$

with the last equality holding because in this second analyzed case $0 \leqslant \lfloor -j_1p_1\rfloor_{p_2} + \phi_1 - p_2 \leqslant p_2 - 1$. The expressions of the two cases coincide with the one of Equation (22) to be proven. This concludes the proof. □

In summary, a tighter expression for $\theta^{\mathsf{w}}_{1\blacktriangleright 2}(j_1,*)$ is found by replacing (22) in (20), that is,

$$\theta^{\mathsf{w}}_{1\blacktriangleright 2}(j_1,*) = \theta^{\mathsf{w}}_2 - \Theta + \lfloor\Theta\rfloor_G + \lfloor \phi_1 - j_1p_1\rfloor_{p_2} G \tag{24}$$

in which we highlight:

- a constant term $\theta^{\mathsf{w}}_2 - \Theta + \lfloor\Theta\rfloor_G$ plus
- a per-job varying term $\lfloor \phi_1 - j_1p_1\rfloor_{p_2} G$.

This new expression of Eq. (24) enables us to tighten the bounds of (21).

**Corollary 1.** Let us set $G = \gcd(T_1, T_2)$. If $G > 0$, then

$$\lfloor\Theta\rfloor_G \leqslant \theta^{\mathsf{w}}_{1\blacktriangleright 2}(j_1,*) - (\theta^{\mathsf{w}}_2 - \Theta) \leqslant \lfloor\Theta\rfloor_G + T_2 - G. \quad (25)$$

*Proof.* From the bounds of (11), the expression $\lfloor\phi_1 - j_1p_1\rfloor_{p_2}$ is bounded by

$$0 \leqslant \lfloor\phi_1 - j_1p_1\rfloor_{p_2} \leqslant p_2 - 1,$$

which implies, through Theorem 1, that

$$\lfloor\Theta\rfloor_G \leqslant \lfloor\Theta - j_1T_1\rfloor_{T_2} \leqslant (p_2-1)G + \lfloor\Theta\rfloor_G.$$

Plugging now the bounds above into the expression for $\theta^{\mathsf{w}}_{1\blacktriangleright 2}(j_1,*)$ of (24), we smoothly find (25), concluding then the proof. □

Corollary 1 provides bounds to $\theta^{\mathsf{w}}_{1\blacktriangleright 2}(j_1,*)$. Hence, the natural question is whether these upper and lower bounds are tight or not. The next theorem provides an affirmative answer and it additionally gives the explicit expression of the write phasings for all $j_1$.

**Theorem 2.** *Let us set $G = \gcd(T_1, T_2)$. If $G > 0$, then for any $k = 0, \dots, p_2 - 1$*

$$j_1 \equiv (\phi_1 - k)p_1^{-1} \mod p_2 \quad \Leftrightarrow \quad \theta^{\mathsf{w}}_{1\blacktriangleright 2}(j_1,*) = \theta^{\mathsf{w}}_2 - \Theta + \lfloor\Theta\rfloor_G + kG \quad (26)$$

*with $\phi_1$ as in (23) and $p_1^{-1}$ denoting the multiplicative inverse of $p_1$ over $\mathbb{Z}/p_2\mathbb{Z}$.*

*Proof.* We first observe that, from the definition of $G = \gcd(T_1, T_2)$, we have $\gcd(p_1, p_2) = 1$ with $T_1 = p_1G$ and $T_2 = p_2G$. Consequently, $p_1$ has the multiplicative inverse over the ring $\mathbb{Z}/p_2\mathbb{Z}$. Now, from $j_1 \equiv (\phi_1 - k)p_1^{-1} \mod p_2$ we find

$$\phi_1 - j_1p_1 \equiv k \mod p_2$$
$$\lfloor\phi_1 - j_1p_1\rfloor_{p_2} = k$$

and by applying Eq. (22) and the expression of $\theta^{\mathsf{w}}_{1\blacktriangleright 2}(j_1,*)$ of (20), we immediately find (26) concluding the proof. □

The proven theorem provides the explicit expression of the write phasing $\theta^{\mathsf{w}}_{1\blacktriangleright 2}(j_1,*)$ and, more importantly, it tells what are the indices $j_1 \in \mathbb{J}_1 = \mathbb{Z}$ for which the write phasing takes any given value. For example, the minimum and maximum phasings are taken respectively when

$$\theta^{\mathsf{w}}_{1\blacktriangleright 2}(j_1,*) \text{ is min} \Leftrightarrow k = 0 \Leftrightarrow j_1 \equiv \phi_1 p_1^{-1} \mod p_2$$
$$\theta^{\mathsf{w}}_{1\blacktriangleright 2}(j_1,*) \text{ is max} \Leftrightarrow k = p_2 - 1 \Leftrightarrow j_1 \equiv (\phi_1+1) p_1^{-1} \mod p_2$$

Also, they are equal to the lower and upper bounds of (25), respectively, hence the bounds of (25) are tight.

In the example of Figure 2, we have $p_2 = 5$ and $p_1 = 8$ and then $p_1^{-1} = 2$ because $\lfloor 8 \times 2\rfloor_5 = \lfloor 16\rfloor_5 = 1$. Since $\phi_1 = 1$, then the minimum write phasing is taken when

$$j_1 \equiv \phi_1 p_1^{-1} \equiv 2 \mod 5,$$

while the maximum occurs when

$$j_1 \equiv (\phi_1 + 1)p_1^{-1} \equiv 4 \mod 5.$$

While the separation between the read instant of two consecutive jobs of $\tau_{1\blacktriangleright 2}$ is constantly equal to $T_1$, the separation between write instants has a more complex expression.

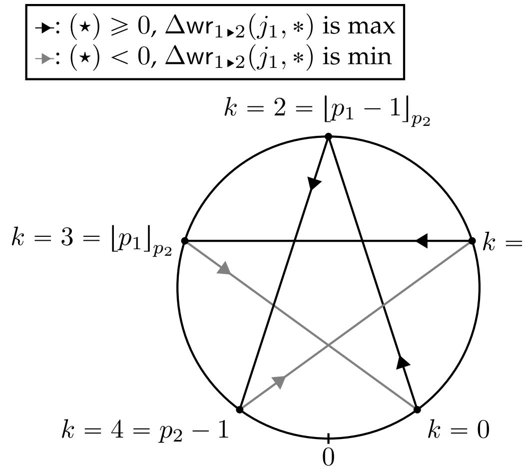


Fig. 3. Illustrating the proof of Lemma 2, with the same parameters as the example of Figure 2. In such a case, we have $p_2 = 5$ and $\lfloor p_1\rfloor_{p_2} = 3$. For each value of $k = \lfloor\phi_1 - j_1p_1\rfloor_{p_2}$, the arrow points to $\lfloor k - p_1\rfloor_{p_2}$. The difference $(\star) = \lfloor k - p_1\rfloor_{p_2} - k$ is non-negative when "moving" from $k$ to $\lfloor k - p_1\rfloor_{p_2}$ we do not cross the 0 (black arrows). It is negative otherwise (gray arrows).

**Lemma 2.** If $G = \gcd(T_1, T_2) > 0$, then

$$\Delta\mathsf{wr}_{1\blacktriangleright 2}(j_1,*) = T_1 + \begin{cases} -\lfloor p_1\rfloor_{p_2} G & \text{if } \lfloor\phi_1 - j_1p_1\rfloor_{p_2} \geqslant \lfloor p_1\rfloor_{p_2} \\ \lfloor -p_1\rfloor_{p_2} G & \text{otherwise} \end{cases} \quad (27)$$

with $\phi_1$ as in (23).

*Proof.* First, from the expression of $\theta^{\mathsf{w}}_{1\blacktriangleright 2}(j_1,*)$ of (24), we find

$$\Delta\mathsf{wr}_{1\blacktriangleright 2}(j_1,*) = (\lfloor -(j_1+1)p_1 + \phi_1\rfloor_{p_2} - \lfloor -j_1p_1 + \phi_1\rfloor_{p_2})G + T_1$$

By setting

$$k = \lfloor\phi_1 - j_1p_1\rfloor_{p_2}$$

the only term of $\Delta\mathsf{wr}_{1\blacktriangleright 2}(j_1,*)$ which has a dependency on $j_1$ can be rewritten as

$$\lfloor\phi_1 - (j_1+1)p_1\rfloor_{p_2} - \lfloor\phi_1 - j_1p_1\rfloor_{p_2} = \lfloor k - p_1\rfloor_{p_2} - k = (\star).$$

We now study in greater depth the quantity $(\star)$, also represented in Figure 3. If $k \geqslant \lfloor p_1\rfloor_{p_2}$ then

$$(\star) = \left\lfloor k - \lfloor p_1\rfloor_{p_2}\right\rfloor_{p_2} - k = k - \lfloor p_1\rfloor_{p_2} - k = -\lfloor p_1\rfloor_{p_2},$$

in which we have applied

- the properties (8) and (9) for the first equality, and
- the property (6), applicable because $k - \lfloor p_1\rfloor_{p_2} \geqslant 0$, for the second equality.

If instead $k \leqslant \lfloor p_1\rfloor_{p_2} - 1$, then

$$(\star) = \left\lfloor k + p_2 - \lfloor p_1\rfloor_{p_2}\right\rfloor_{p_2} - k = k + p_2 - \lfloor p_1\rfloor_{p_2} - k = \lfloor -p_1\rfloor_{p_2}.$$

By replacing the two expressions we find exactly Eq. (27), which concludes the proof. □

From the expression of the $\Delta\mathsf{wr}_{1\blacktriangleright 2}(j_1,*)$ of (27) it is not immediately apparent that $\Delta\mathsf{wr}_{1\blacktriangleright 2}(j_1,*)$ is a multiple of $T_2$

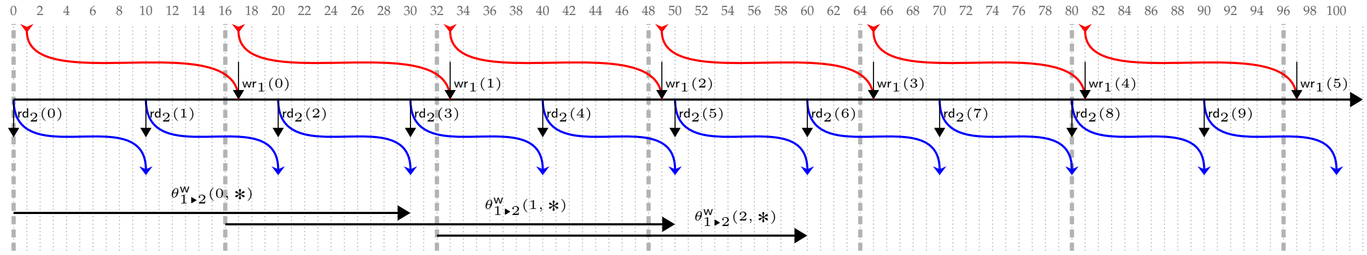


Fig. 4. Illustrating the example of Section 5.1.1, with $T_1 = 16$, $T_2 = 10$, $\theta_1^{\mathsf{r}} = 1$, $\theta_2^{\mathsf{r}} = 0$, and $\theta_i^{\mathsf{w}} = \theta_i^{\mathsf{r}} + T_i$ for both tasks. Every job is represented by a curvy arrow starting at its read instant and ending at the write one. Any job $j$ of $\tau_1$ (in red) reads at $\mathsf{rd}_1(j) = \theta_1^{\mathsf{r}} + jT_1$ and writes at $\mathsf{wr}_1(j) = \theta_1^{\mathsf{w}} + jT_1$. Jobs of $\tau_2$ are in blue. Black vertical arrows are placed at write and read instants. The write phases of the jobs of the whole chain $\tau_{1\blacktriangleright 2}$ are also shown by horizontal arrows. Since we aim at modeling the chain with the same formalism of periodic tasks of (1), the read/write phasings are relative to multiples of the chain period $T_{1\blacktriangleright 2} = T_1$, represented by thick vertical dashed gray segments.

for all $j_1$, as one would expect it to be. The next corollary confirms that such intuition is correct.

**Corollary 2.** If $G = \gcd(T_1, T_2) > 0$, then

$$\Delta\mathsf{wr}_{1\blacktriangleright 2}(j_1, *) = \begin{cases} \lfloor T_1/T_2 \rfloor T_2 & \text{if } \lfloor \phi_1 - j_1 p_1 \rfloor_{p_2} \geqslant \lfloor p_1 \rfloor_{p_2} \\ \lceil T_1/T_2 \rceil T_2 & \text{otherwise} \end{cases} \tag{28}$$

with $\phi_1$ as in (23).

*Proof.* The corollary can be easily proven by exploiting the definition of $\lfloor \cdot \rfloor_m$ in the formulation of Equation (27), which can be rewritten as follows:

$$T_1 - \lfloor p_1 \rfloor_{p_2} G = T_1 - \left( p_1 - \left\lfloor \frac{p_1}{p_2} \right\rfloor p_2 \right) G = \left\lfloor \frac{T_1}{T_2} \right\rfloor T_2$$

$$T_1 + \lfloor -p_1 \rfloor_{p_2} G = T_1 + \left( -p_1 - \left\lfloor \frac{-p_1}{p_2} \right\rfloor p_2 \right) G$$

$$= - \left\lfloor \frac{-p_1}{p_2} \right\rfloor T_2 = \left\lceil \frac{p_1}{p_2} \right\rceil T_2 = \left\lceil \frac{T_1}{T_2} \right\rceil T_2.$$

This concludes the proof. □

We conclude the analysis of the case with $T_1 \geqslant T_2$ by showing that in the harmonic case, all these results are in accordance to the common intuition. If $T_1$ is an integer multiple of $T_2$, then $G = \gcd(T_1, T_2) = T_2$. In turn $p_2 = 1$, meaning that the write phasing is constantly equal to $\theta_{1\blacktriangleright 2}^{\mathsf{w}}(j_1, *) = \theta_2^{\mathsf{w}} - \Theta + \lfloor \Theta \rfloor_{T_2}$ and that the separation between two consecutive writes is also constant and equal to $T_1$.

### 5.1.1 An example

Let us now illustrate the results found through an example. We consider a pair of tasks $\tau_1$ and $\tau_2$ with parameters

$$T_1 = 16 \qquad T_2 = 10$$
$$\theta_1^{\mathsf{r}} = 1 \qquad \theta_2^{\mathsf{r}} = 0$$
$$\theta_1^{\mathsf{w}} = \theta_1^{\mathsf{r}} + T_1 = 17 \qquad \theta_2^{\mathsf{w}} = \theta_2^{\mathsf{r}} + T_2 = 10.$$

and $\tau_2$ reading from $\tau_1$. The greatest common divisor between the two periods is $G = \gcd(16, 10) = 2$, thus $p_1 = 8$ and $p_2 = 5$ so that $T_1 = p_1 G$ and $T_2 = p_2 G$, respectively. From Equations (17) and (23), we obtain $\Theta = \theta_2^{\mathsf{r}} - \theta_1^{\mathsf{w}} = -17$ and $\phi_1 = 1$. Read and write instants are drawn in Figure 4. This same parameter values can be found in our repository at https://github.com/ebni/periodic-LET.

Since $T_1 \geqslant T_2$ the period of the chain $T_{1\blacktriangleright 2}$ is equal to $T_1$. Also, Table 1 reports, for each job $j_1$ of $\tau_1$,

- the value of $\lfloor \phi_1 - j_1 p_1 \rfloor_{p_2} \in \mathbb{Z}/p_2\mathbb{Z}$, which is the expression determining the per-job variability of the write phasing, as shown in (24). We remind that such expression can be visualized as the position over the ring shown in Figure 2,
- the write phasing $\theta_{1\blacktriangleright 2}^{\mathsf{w}}(j_1, *)$ of the job $(j_1, *)$ of the chain $\tau_{1\blacktriangleright 2}$ originated from job $j_1$ of $\tau_1$, found from its expression of (24),
- the absolute write instant $\mathsf{wr}_{1\blacktriangleright 2}(j_1, *) = j_1 T_{1\blacktriangleright 2} + \theta_{1\blacktriangleright 2}^{\mathsf{w}}(j_1, *)$, and
- the separation $\Delta\mathsf{wr}_{1\blacktriangleright 2}(j_1, *)$ to the next write instant.

As stated in Theorem 2, the write phasing is minimal when $\lfloor \phi_1 - j_1 p_1 \rfloor_{p_2} = 0$, which happens if and only if

$$j_1 \equiv \phi_1 p_1^{-1} \equiv 2 \mod 5$$

because the inverse $p_1^{-1}$ of $p_1$ over $\mathbb{Z}/p_2\mathbb{Z} = \mathbb{Z}/5\mathbb{Z}$ is 2. Hence, the job indices with minimal $\theta_{1\blacktriangleright 2}^{\mathsf{w}}(j_1, *)$ are $j_1 = \ldots, -8, -3, 2, 7, 12, 17, \ldots$ With the same argument, the write phasing $\theta_{1\blacktriangleright 2}^{\mathsf{w}}(j_1, *)$ is maximal, when $\lfloor \phi_1 - j_1 p_1 \rfloor_{p_2} = p_2 - 1$, i.e., $j_1 = \ldots, -6, -1, 4, 9, \ldots$

Finally, Lemma 2 allows us to determine analytically the separation $\Delta\mathsf{wr}_{1\blacktriangleright 2}(j_1, *)$ of two consecutive writes for any given job. When $\lfloor \phi_1 - j_1 p_1 \rfloor_{p_2} \in \{\lfloor p_1 \rfloor_{p_2}, \ldots, p_2 - 1\}$ equal to $\{3, 4\}$ in our example, then $\Delta\mathsf{wr}_{1\blacktriangleright 2}(j_1, *)$ is equal to its smaller value $\lfloor T_1/T_2 \rfloor T_2$. Otherwise, it is equal to $\lceil T_1/T_2 \rceil T_2$. In our example

$$\Delta\mathsf{wr}_{1\blacktriangleright 2}(j_1, *) = \begin{cases} 20 & \text{if } k \in \{0, 1, 2\} \\ 10 & \text{if } k \in \{3, 4\} \end{cases}$$
$$\text{with } j_1 \equiv (\phi - k) p_1^{-1} \mod p_2.$$

TABLE 1
Job indexing and latency characterization for the example of Section 5.1.1.

| $j_1$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | ... |
|---|---|---|---|---|---|---|---|---|---|
| $\lfloor \phi_1 - j_1 p_1 \rfloor_{p_2}$ | 1 | 3 | 0 | 2 | 4 | 1 | 3 | 0 | ... |
| $\theta_{1\blacktriangleright 2}^{\mathsf{w}}(j_1, *)$ | 30 | 34 | 28 | 32 | 36 | 30 | 34 | 28 | ... |
| $\mathsf{wr}_{1\blacktriangleright 2}(j_1, *)$ | 30 | 50 | 60 | 80 | 100 | 110 | 130 | 140 | ... |
| $\Delta\mathsf{wr}_{1\blacktriangleright 2}(j_1, *)$ | 20 | 10 | 20 | 20 | 10 | 20 | 10 | 20 | ... |

### 5.2 Case $T_2 \geqslant T_1$

We are now assuming that $T_2 \geqslant T_1$. This section mimics the same steps of 5.1. Hence, we may omit some details when they do not add value to the discussion.

When $T_2 \geqslant T_1$, there is always a write instant $\mathsf{wr}_1(j)$ between any pair of consecutive read instants of $\tau_2$. Borrowing the definitions of Section 4, this fact is formally expressed by

$$\forall j_2 \in \mathbb{J}_2, \quad \mathbb{J}_1(j_2) \neq \varnothing,$$

which allows us to define $\mathbb{J}_{1\blacktriangleright 2}$ through Equation (14) as

$$\mathbb{J}_{1\blacktriangleright 2} = \{(j_1, j_2) \in \mathbb{J}_1 \times \mathbb{J}_2 : j_1 = \max \mathbb{J}_1(j_2)\} = \left\{ \left( \left\lfloor \frac{j_2 T_2 + \Theta}{T_1} \right\rfloor, j_2 \right) : j_2 \in \mathbb{J}_2 \right\}$$

with $\Theta$ as defined earlier in (17). Also, to recall that jobs $(j_1, j_2) \in \mathbb{J}_{1\blacktriangleright 2}$ are identified by $j_2$ only, we write them as $(*, j_2)$.

Aiming now at describing the chain $\tau_{1\blacktriangleright 2}$ as a periodic one, the expression of the write instants of any job $(*, j_2) \in \mathbb{J}_2$

$$\mathsf{wr}_{1\blacktriangleright 2}(*, j_2) = \mathsf{wr}_2(j_2) = j_2 T_2 + \theta_2^{\mathsf{w}}$$

reveals that

1) the period of the chain $\tau_{1\blacktriangleright 2}$ is
$$T_{1\blacktriangleright 2} = T_2$$
2) and that the write phasing of any job $(*, j_2) \in \mathbb{J}_{1\blacktriangleright 2}$ of the chain is constant
$$\forall (*, j_2) \in \mathbb{J}_{1\blacktriangleright 2}, \quad \theta^{\mathsf{w}}_{1\blacktriangleright 2}(*, j_2) = \theta_2^{\mathsf{w}}.$$

The read instant of job $(*, j_2)$ is instead more involved

$$\begin{aligned} \mathsf{rd}_{1\blacktriangleright 2}(*, j_2) &= \left\lfloor \frac{j_2 T_2 + \Theta}{T_1} \right\rfloor T_1 + \theta_1^{\mathsf{r}} \\ &= j_2 T_2 + \left\lfloor \frac{j_2 T_2 + \Theta}{T_1} \right\rfloor T_1 - j_2 T_2 + \theta_1^{\mathsf{r}} \\ &= j_2 T_2 + \theta^{\mathsf{r}}_{1\blacktriangleright 2}(*, j_2) \end{aligned} \tag{29}$$

with the job-dependent read phasing equal to

$$\begin{aligned} \theta^{\mathsf{r}}_{1\blacktriangleright 2}(*, j_2) &= \left\lfloor \frac{j_2 T_2 + \Theta}{T_1} \right\rfloor T_1 - j_2 T_2 + \theta_1^{\mathsf{r}} \\ &= \left\lfloor \frac{j_2 T_2 + \Theta}{T_1} \right\rfloor T_1 - (j_2 T_2 + \Theta) + \Theta + \theta_1^{\mathsf{r}} \\ &= -\lfloor j_2 T_2 + \Theta \rfloor_{T_1} + \theta_1^{\mathsf{r}} + \Theta. \end{aligned} \tag{30}$$

Similarly as in Section 5.1, if $\gcd(T_1, T_2) > 0$ the next theorem enables us to find a tighter value for $\theta^{\mathsf{r}}_{1\blacktriangleright 2}(*, j_2)$.

**Theorem 3.** *Let us set $G = \gcd(T_1, T_2)$. If $G > 0$, then*

$$\forall j_2 \in \mathbb{Z}, \ \lfloor j_2 T_2 + \Theta \rfloor_{T_1} = \lfloor j_2 p_2 + \phi_2 \rfloor_{p_1} G + \lfloor \Theta \rfloor_G \tag{31}$$

*with $p_1, p_2$ such that $T_1 = p_1 G$, $T_2 = p_2 G$, and $\phi_2$ integer quotient of the following division by $G$*

$$\lfloor \Theta \rfloor_{T_1} = \phi_2 G + \lfloor \Theta \rfloor_G \tag{32}$$

*Proof.* By making the following substitutions

$$\begin{aligned} j_2 &\longrightarrow -j_1 \\ T_2 &\longrightarrow T_1 \\ T_1 &\longrightarrow T_2 \end{aligned}$$

the statement of (31) is the same as the one of (22) of Theorem 1. Hence, this theorem holds too. □

The just proved theorem allows us writing the following tighter expression for the read phasing

$$\theta^{\mathsf{r}}_{1\blacktriangleright 2}(*, j_2) = \theta_1^{\mathsf{r}} + \Theta - \lfloor \Theta \rfloor_G - \lfloor j_2 p_2 + \phi_2 \rfloor_{p_1} G \tag{33}$$

and to determine tighter lower and upper bounds.

**Corollary 3.** Let us set $G = \gcd(T_1, T_2)$. If $G > 0$, then

$$-\lfloor \Theta \rfloor_G - T_1 + G \leqslant \theta^{\mathsf{r}}_{1\blacktriangleright 2}(*, j_2) - (\theta_1^{\mathsf{r}} + \Theta) \leqslant -\lfloor \Theta \rfloor_G \tag{34}$$

*Proof sketch.* The result follows from the bounds to $\lfloor j_2 p_2 + \phi_2 \rfloor_{p_1}$ from Eq. (11) and the expression of $\theta^{\mathsf{r}}_{1\blacktriangleright 2}(*, j_2)$ of Eq. (33). □

Following the same steps of Section 5.1, we now "invert" the expression of Eq. (33) to find explicitly the job indices $(*, j_2)$ for which the $\theta^{\mathsf{r}}_{1\blacktriangleright 2}(*, j_2)$ takes any given value.

**Corollary 4.** Let us set $G = \gcd(T_1, T_2)$. If $G > 0$, then for any $k = 0, \ldots, p_1 - 1$

$$j_2 \equiv (k - \phi_2) p_2^{-1} \mod p_1 \quad \Leftrightarrow \quad \theta^{\mathsf{r}}_{1\blacktriangleright 2}(*, j_2) = \theta_1^{\mathsf{r}} + \Theta - \lfloor \Theta \rfloor_G - kG \tag{35}$$

with $\phi_2$ as in (32) and $p_2^{-1}$ denoting the multiplicative inverse of $p_2$ over $\mathbb{Z}/p_1\mathbb{Z}$.

*Proof sketch.* The corollary follows by finding $j_2$ as function of $k$ from $k = \lfloor j_2 p_2 + \phi_2 \rfloor_{p_1}$. Then substituting in (33), we find (35). □

Somehow "dually" to the case $T_1 \geqslant T_2$, Corollary 4 affirms that the upper and lower bounds of (34) are taken when $k = 0$ and $k = p_1 - 1$, respectively (while the upper and lower bounds for $\theta^{\mathsf{w}}_{1\blacktriangleright 2}(j_1, *)$ in the case with $T_1 \geqslant T_2$ were taken when $k = p_2 - 1$ and $k = 0$). From (35), the job indices $(*, j_2) \in \mathbb{J}_{1\blacktriangleright 2}$ with minimum and maximum read phasing are

$$\begin{aligned} &\theta^{\mathsf{r}}_{1\blacktriangleright 2}(*, j_2) \text{ is min} \Leftrightarrow k = p_1 - 1 \Leftrightarrow j_2 \equiv -(\phi_2 + 1) p_2^{-1} \mod p_1 \\ &\theta^{\mathsf{r}}_{1\blacktriangleright 2}(*, j_2) \text{ is max} \Leftrightarrow \quad k = 0 \quad \Leftrightarrow j_2 \equiv -\phi_2 p_2^{-1} \mod p_1 \end{aligned}$$

The next lemma determines tight bounds on the separation between two consecutive readings.

**Lemma 3.** If $G = \gcd(T_1, T_2) > 0$, then

$$\Delta \mathsf{rd}_{1\blacktriangleright 2}(*, j_2) = \begin{cases} \lceil T_2/T_1 \rceil T_1 & \text{if } \lfloor j_2 p_2 + \phi_2 \rfloor_{p_1} \geqslant \lfloor -p_2 \rfloor_{p_1} \\ \lfloor T_2/T_1 \rfloor T_1 & \text{otherwise} \end{cases} \tag{36}$$

with $\phi_2$ as in (32).

*Proof.* From (33),

$$\Delta \mathsf{rd}_{1\blacktriangleright 2}(*, j_2) = T_2 - (\lfloor (j_2 + 1) p_2 + \phi_2 \rfloor_{p_1} - \lfloor j_2 p_2 + \phi_2 \rfloor_{p_1}) G.$$

If $T_2$ is a multiple of $T_1$, that is, $p_1 = 1$, then

$$\Delta \mathsf{rd}_{1\blacktriangleright 2}(*, j_2) = T_2$$

because $\lfloor x \rfloor_1 = 0$ for any integer $x$. For the rest of the proof, we assume then that $T_2$ is not a multiple of $T_1$

Let us now investigate the only term depending on $j_2$.

$$\lfloor (j_2 + 1) p_2 + \phi_2 \rfloor_{p_1} - \lfloor j_2 p_2 + \phi_2 \rfloor_{p_1} = \lfloor k + p_2 \rfloor_{p_1} - k = (\star)$$

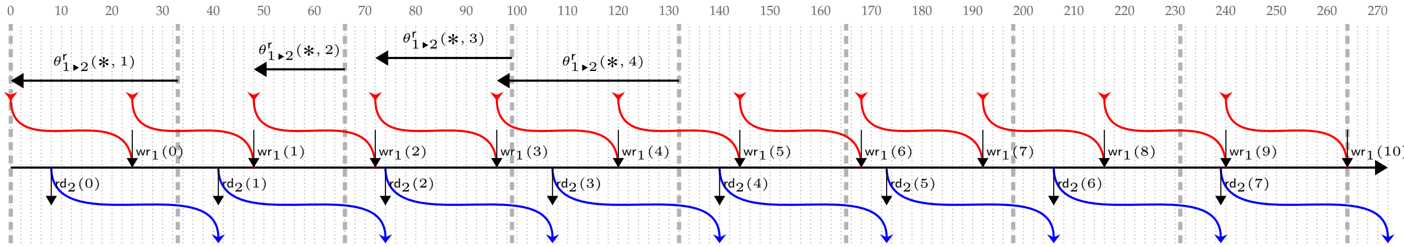


Fig. 5. Illustrating the example of Section 5.2.1. In this case $T_2 \geqslant T_1$ implies that the period $T_{1\blacktriangleright 2}$ is equal to $T_2$. The write phasing of the chain is then constant over the jobs and it is $\theta^{w}_{1\blacktriangleright 2}(*,j_2) = \theta^{w}_2(j_2) = 41$. The read phasings of the jobs of the chain, instead, vary with the pattern reported in Table 2. Observe that the labeling of the $\tau_{1\blacktriangleright 2}$ jobs by the job index $j_2$ of $\tau_2$ produces negative read phases $\theta^{r}_{1\blacktriangleright 2}(*,j_2)$. This is represented by arrows pointing backward from the multiple of periods at $j_2T_2$.

with

$$k = \lfloor j_2p_2 + \phi_2\rfloor_{p_1}.$$

If $k = 0$ then $(\star) \geqslant 0$. The expression $(\star)$ remains non-negative for all next values of $k$ until $\lfloor k+p_2\rfloor_{p_1} = p_1 - 1$, which is reached when $k = \lfloor -p_2 - 1\rfloor_{p_1}$. Hence, for all $k \in [0,\ldots,\lfloor -p_2-1\rfloor_{p_1}]$, $(\star) \geqslant 0$ and then

$$(\star) = \lfloor k+p_2\rfloor_{p_1} - k = \Big\lfloor \lfloor k+p_2\rfloor_{p_1} - k\Big\rfloor_{p_1} = \lfloor p_2\rfloor_{p_1}$$
$$\Delta\mathsf{rd}_{1\blacktriangleright 2}(*,j_2) = T_2 - \lfloor p_2\rfloor_{p_1}G = T_2 - (p_2 - \lfloor p_2/p_1\rfloor p_1)G = \lfloor T_2/T_1\rfloor T_1$$

If instead $k \geqslant \lfloor -p_2\rfloor_{p_1}$ then $-(p_1-1) \leqslant (\star) \leqslant -1$. By adding and subtracting $p_1$ we find

$$(\star) = \overbrace{\lfloor k+p_2\rfloor_{p_1} - k + p_1}^{\in[0,p_1-1]} - p_1 = \lfloor p_2\rfloor_{p_1} - p_1$$
$$\Delta\mathsf{rd}_{1\blacktriangleright 2}(*,j_2) = T_2 - \lfloor p_2\rfloor_{p_1}G + T_1 = \lfloor T_2/T_1\rfloor T_1 + T_1 = \lceil T_2/T_1\rceil T_1$$

because $T_2$ is not a multiple of $T_1$. This concludes the proof. □

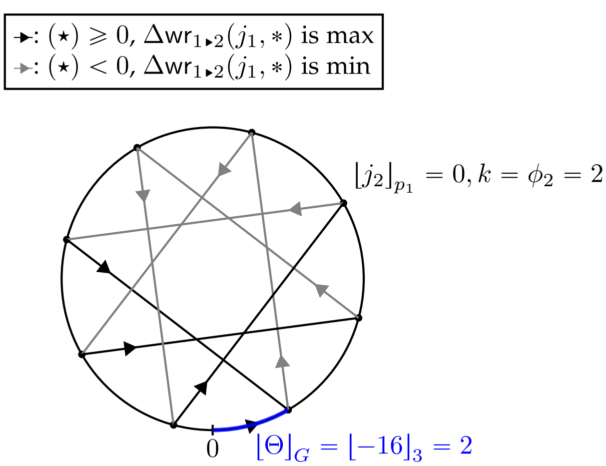


Fig. 6. Similarly as in Figure 2, this one represents the variable part $\lfloor j_2T_2 + \Theta\rfloor_{T_1}$ of the read phasing $\theta^{r}_{1\blacktriangleright 2}(*,j_2)$ as function of the job $j_2$ of $\tau_2$. In this example, $T_1 = 24$ and $T_2 = 33$, meaning that $G = 3$, $p_1 = 8$ and $p_2 = 11$. The phasings of $\theta^{r}_1 = 0$, $\theta^{w}_1 = \theta^{r}_1 + T_1 = 24$, and $\theta^{r}_2 = 8$, $\theta^{w}_2 = \theta^{r}_2 + T_2 = 41$ makes $\Theta = \theta^{r}_2 - \theta^{w}_1 = 8 - 24 = -16$. From these parameters, the first job $j_2 = 0$ starts at vertex $k = \phi_2 = 2$.

#### 5.2.1 *An example*

For the case of $T_2 \geqslant T_1$, we consider again the case of $\tau_2$ reading from $\tau_1$, with the following parameters

$$\begin{aligned} T_1 &= 24 & T_2 &= 33\\ \theta^{r}_1 &= 0 & \theta^{r}_2 &= 8\\ \theta^{w}_1 &= \theta^{r}_1 + T_1 = 24 & \theta^{w}_2 &= \theta^{r}_2 + T_2 = 41. \end{aligned}$$

The greatest common divisor between the two task periods is $\gcd(24,33) = G = 3$, thus $p_1 = 8$ and $p_2 = 11$. From Equations (17) and (32), we obtain $\Theta = -16$ and $\phi_2 = 1$.

The period and write phasing of the chain $\tau_{1\blacktriangleright 2}$ are determined by task $\tau_2$, that is,

$$T_{1\blacktriangleright 2} = T_2 \qquad \forall j_2,\quad \theta^{w}_{1\blacktriangleright 2}(*,j_2) = \theta^{w}_2.$$

The read phasing $\theta^{r}_{1\blacktriangleright 2}(*,j_2)$, instead, is not constant and follows the pattern shown over time in Figure 5, and shown over the ring in Figure 6. Table 2 reports the sequence of jobs and the corresponding quantities:

- first, it is reported the index $j_2$ of $\tau_2$, which also indicates the job $(*,j_2) \in \mathbb{J}_{1\blacktriangleright 2}$. Then,
- the value $\lfloor j_2p_2 + \phi_2\rfloor_{p_1}$ represents the index among the $p_1$ vertices of the "star" of Figure 6. All vertices are span as $j_2$ varies.
- The read phasing $\theta^{r}_{1\blacktriangleright 2}(*,j_2)$ is also reported as explicitly computed from Eq. (33). For the values of this example, we have $\theta^{r}_{1\blacktriangleright 2}(*,j_2) = -18 - 3\lfloor j_2p_2 + \phi_2\rfloor_{p_1}$. We remark that the necessary choice of indexing the job of the chain $\tau_{1\blacktriangleright 2}$ by the index $j_2$ of the job of $\tau_2$ produces a negative read phasing. This, however, does not contradict any given definition or hypothesis.
- From the definition $\mathsf{rd}_{1\blacktriangleright 2}(*,j_2) = j_2T_{1\blacktriangleright 2} + \theta^{r}_{1\blacktriangleright 2}(*,j_2)$ of Equation (1), we can also find the read instant of any job, and finally
- the separation $\Delta\mathsf{rd}_{1\blacktriangleright 2}(*,j_2)$ between the read instants of two consecutive jobs in $\mathbb{J}_{1\blacktriangleright 2}$ is also computed.

TABLE 2
Job indexing and characterization of the read instants and phasings for the example of Section 5.2.1.

| $j_2$ | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | ... |
|---|---|---|---|---|---|---|---|---|---|
| $\lfloor j_2p_2+\phi_2\rfloor_{p_1}$ | 2 | 5 | 0 | 3 | 6 | 1 | 4 | 7 | ... |
| $\theta^{r}_{1\blacktriangleright 2}(*,j_2)$ | -24 | -33 | -18 | -27 | -36 | -21 | -30 | -39 | |
| $\mathsf{rd}_{1\blacktriangleright 2}(*,j_2)$ | -24 | 0 | 48 | 72 | 96 | 144 | 168 | 192 | |
| $\Delta\mathsf{rd}_{1\blacktriangleright 2}(*,j_2)$ | 24 | 48 | 24 | 24 | 48 | 24 | 24 | 48 | ... |

This article has been accepted for publication in IEEE Transactions on Computers. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TC.2023.3283707



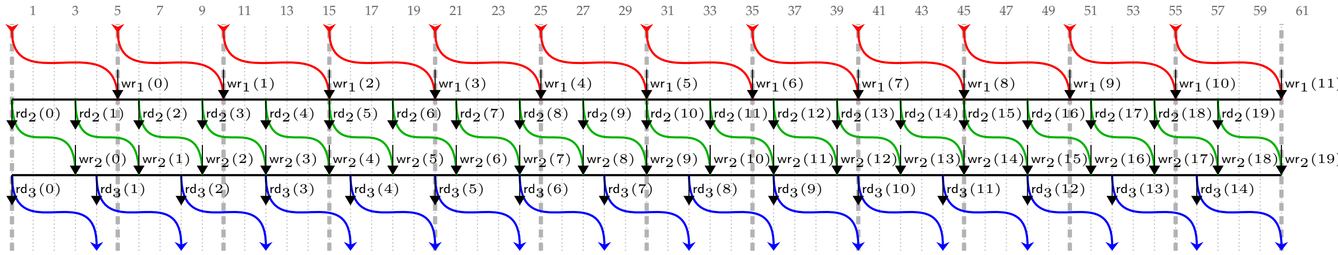


Fig. 7. When $T_1 = 5$, $T_2 = 3$, $T_3 = 4$, one job of $\tau_1$ every $60 = \text{lcm}(T_1, T_2, T_3)$ time units is necessarily erased from the jobs $\mathbb{J}_{1\blacktriangleright 2\blacktriangleright 3}$ of the chain $\tau_{1\blacktriangleright 2\blacktriangleright 3}$. This implies that the period $T_{1\blacktriangleright 2\blacktriangleright 3}$ of the chain is not equal to $\max\{T_1, T_2, T_3\} = T_3 = 5$.

As stated in Lemma 3, the interval between two reading instants can be computed from

$$\Delta\mathsf{rd}_{1\blacktriangleright 2}(*, j_2) = \begin{cases} 48 & \text{if } \lfloor j_2 p_2 + \phi_2 \rfloor_{p_1} \geqslant \lfloor -p_2 \rfloor_{p_1} = 5 \\ 24 & \text{otherwise.} \end{cases}$$

## 6 Zero-Jitter Task Chains

When extending the analysis to chains of more than two tasks the first question is about the chain periodicity. Can we say that the period $T_{1\blacktriangleright 2\blacktriangleright\cdots\blacktriangleright n}$ of an arbitrary chain $\tau_{1\blacktriangleright 2\blacktriangleright\cdots\blacktriangleright n}$ is equal to $\max\{T_1, T_2, \ldots, T_n\}$? Unfortunately, this is not true in general. The next two examples illustrate this fact, and show that the relative phasings of tasks may affect, with our initial surprise, the period of the chain.

### 6.1 The chain period may be larger than the maximum

In this first example we show a chain composed of three LET tasks, where the period of the chain is strictly larger than the maximum period across its tasks.

**Example 1.** *Let us consider a chain $\tau_{1\blacktriangleright 2\blacktriangleright 3}$, with*

$$\begin{aligned} &T_1 = 5, \quad T_2 = 3, \quad T_3 = 4, \\ &\theta_1^{\mathsf{r}} = \theta_2^{\mathsf{r}} = \theta_3^{\mathsf{r}} = 0, \\ &\theta_i^{\mathsf{w}} = \theta_i^{\mathsf{r}} + T_i, \qquad i = 1, 2, 3. \end{aligned}$$

*The schedule repeats after 60 time units. Figure 7 shows the read and write instants of jobs over a portion of time. From the Definition 6, the jobs of $\tau_{1\blacktriangleright 2}$ are*

$$\mathbb{J}_{1\blacktriangleright 2} = \{\ldots, (-1, 0), (0, 2), (1, 4), (2, 5), \ldots\}.$$

*Let us now focus on the jobs in $\mathbb{J}_{1\blacktriangleright 2}(12)$, which are the jobs in $\mathbb{J}_{1\blacktriangleright 2}$ with write instant between $\mathsf{rd}_3(11) = 44$ and $\mathsf{rd}_3(12) = 48$ (recall the Definition of (12)). We have*

$$\mathbb{J}_{1\blacktriangleright 2}(12) = \{(7, 14), (8, 15)\},$$

*which implies, from Eq. (14) and Definitions 6 and 8, that job $(7, 14) \in \mathbb{J}_{1\blacktriangleright 2}$ is void, as its output is overwritten by the job $\text{next}(7, 14) = (8, 15)$ before being read by the job $j_3 = 12$ of $\tau_3$. This erasure happens recurrently every 60 time units, i.e., every job of $\tau_1$ indexed by $j_1 = 7 + k \cdot 12$, with $k = 1, 2, \ldots$, does not belong to $\mathbb{J}_{1\blacktriangleright 2\blacktriangleright 3}$. As a consequence, the job set $\mathbb{J}_{1\blacktriangleright 2\blacktriangleright 3}$ contains only 11 every 12 jobs of $\tau_1$, which means that the period of the whole chain is $T_{1\blacktriangleright 2\blacktriangleright 3} = 60/11 > 5 = T_1$.*

For the chain of the example above, we performed an exhaustive search showing that no combination of (constant) phasings exist for $\tau_1$, $\tau_2$ and $\tau_3$, such that $T_{1\blacktriangleright 2\blacktriangleright 3} = \max_i\{T_i\} = 5$. More in general, the chain of Example 1 is representative of a class of chains with $n > 2$ LET tasks, with chain period strictly larger than the largest period among its tasks, regardless of the combination of phasing of those tasks.

We believe that the next example even more interesting as it shows that the period of a chain may depend on the phasings too.

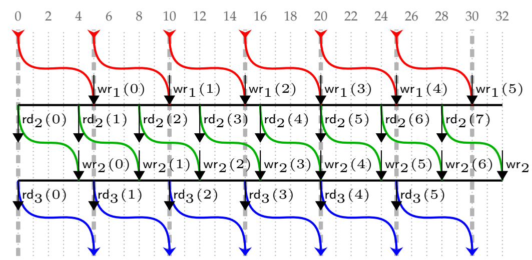


Fig. 8. When $T_1 = 5$, $T_2 = 4$, and $T_3 = 5$, and the read phasings $\theta_i^{\mathsf{r}}$ are all zero, then the periodicity of the chain $T_{1\blacktriangleright 2\blacktriangleright 3}$ **is not equal** to the maximum period $T_1 = T_3 = 5$.

**Example 2.** *Let us consider a chain $\tau_{1\blacktriangleright 2\blacktriangleright 3}$, with*

$$\begin{aligned} &T_1 = 5, \quad T_2 = 4, \quad T_3 = 5, \\ &\theta_i^{\mathsf{r}} = 0, \quad \theta_i^{\mathsf{w}} = T_i, \quad i = 1, 2, 3. \end{aligned}$$

*In this scenario, represented in Figure 8, the jobs of $\tau_{1\blacktriangleright 2}$ are*

$$\mathbb{J}_{1\blacktriangleright 2} = \{\ldots, (-1, 0), (0, 2), (1, 3), (2, 4), (3, 5), \ldots\}.$$

*When $\tau_3$ reads from $\tau_{1\blacktriangleright 2}$, however, job $(1, 3) \in \mathbb{J}_{1\blacktriangleright 2}$ gets overwritten by $(2, 4)$. Hence, job $(1, 3)$ is void w.r.t. $\tau_3$ and the period of the chain is strictly larger than $T_1 = 5$.*

*In this example, however, a careful tuning of the phasings can fix the problem. As shown in Figure 9, by setting the read phasing of $\tau_3$ equal to $\theta_3^{\mathsf{r}} = 2$, all jobs of $\tau_1$ and $\tau_3$ belong to $\mathbb{J}_{1\blacktriangleright 2\blacktriangleright 3}$, which implies that $T_{1\blacktriangleright 2\blacktriangleright 3} = T_1 = 5$. Also, such a task phasing produces a **constant write phasing** $\theta_{1\blacktriangleright 2\blacktriangleright 3}^{\mathsf{w}}$ of the chain $\tau_{1\blacktriangleright 2\blacktriangleright 3}$ equal to 17.*

In the example illustrated in Figure 9, we showed that a careful choice of the task phasings can give the chain the same properties of a LET task: periodic read/write instants and constant phasings. Such a choice is not a fortunate coincidence. Rather, we will be showing that any chain can be made equivalent to a LET task.

### 6.2 Creating a LET chain with a copier task

We consider a chain $\tau_{1\blacktriangleright 2}$, where $\tau_1$ and $\tau_2$ are periodic LET tasks. From the results of Section 5, $\tau_{1\blacktriangleright 2}$ will have

This article has been accepted for publication in IEEE Transactions on Computers. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/TC.2023.3283707



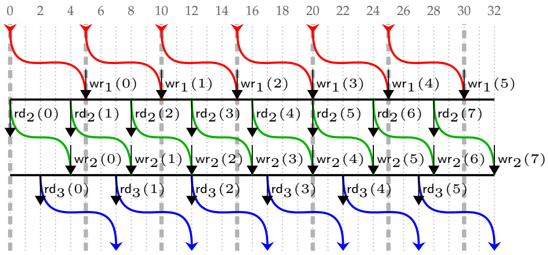


Fig. 9. In the very same scenario of Figure 8 with the only difference of setting the read phasing of $\tau_3$ to $\theta^{\mathsf{r}}_{\underline{2}} = 2$, the period of the chain **is equal** to the maximum period, $T_{1\blacktriangleright 2\blacktriangleright 3} = T_1 = 5$.

$T_{1\blacktriangleright 2} = \max\{T_1, T_2\}$ and either read or write phasing constant, depending if $T_1 > T_2$ or $T_1 < T_2$ (note that when $T_1 = T_2$ both phasings are constant). We are interested in transforming the chain $\tau_{1\blacktriangleright 2}$ in a *periodic LET chain* (see Definition 7).

Our method is based on the introduction of a "copier" task, with negligible execution time, whose aim is to regularize the phasing pattern of the chain. The analysis presented in Section 5 is crucial to find the characteristics of such additional task, as described in next theorem.

**Theorem 4.** *Given a chain $\tau_{1\blacktriangleright 2}$ of LET tasks, with $T_1 \geqslant T_2$. If the LET task $\tau_{\underline{2}}$ with*

- *period $T_{\underline{2}} = T_1$,*
- *read phasing $\theta^{\mathsf{r}}_{\underline{2}}$ such that $\exists \hat{\jmath} \in \mathbb{Z}$*

$$\begin{cases} \theta^{\mathsf{r}}_{\underline{2}} \geqslant \theta^{\mathsf{w}}_2 - \Theta + \lfloor \Theta \rfloor_G + T_2 - G - \hat{\jmath} T_1 \\ \theta^{\mathsf{r}}_{\underline{2}} < \theta^{\mathsf{w}}_2 - \Theta + \lfloor \Theta \rfloor_G + T_1 - \hat{\jmath} T_1 \end{cases} \tag{37}$$

- *and any constant write phasing $\theta^{\mathsf{w}}_{\underline{2}} \geqslant \theta^{\mathsf{r}}_{\underline{2}}$*

*is appended to $\tau_{1\blacktriangleright 2}$, then the chain $\tau_{1\blacktriangleright 2\blacktriangleright \underline{2}}$ has:*

- *period $T_{1\blacktriangleright 2\blacktriangleright \underline{2}}$ equal to $T_1$,*
- *constant read and write phasings, and*
- *set of jobs*

$$\mathbb{J}_{1\blacktriangleright 2\blacktriangleright \underline{2}} = \left\{ \left( j_1, \left\lceil \frac{j_1 T_1 - \Theta}{T_2} \right\rceil, j_1 + \hat{\jmath} \right) : j_1 \in \mathbb{Z} \right\}.$$

*Proof.* We prove the theorem by showing that if $\tau_{\underline{2}}$ satisfies the hypotheses, then no job in $\mathbb{J}_{1\blacktriangleright 2}$ is void w.r.t. $\tau_{\underline{2}}$ (overwritten by the next job, see Def. 8). To make sure that the output of any job $(j_1, *)$ is read by some job $j_{\underline{2}}$ of $\tau_{\underline{2}}$ we need that

$$\mathsf{wr}_{1\blacktriangleright 2}(j_1, *) \leqslant \mathsf{rd}_{\underline{2}}(j_{\underline{2}}) < \overbrace{\mathsf{wr}_{1\blacktriangleright 2}(j_1, *) + \Delta\mathsf{wr}_{1\blacktriangleright 2}(j_1, *)}^{\mathsf{wr}_{1\blacktriangleright 2}(\mathrm{next}(j_1, *))}. \tag{38}$$

From Corollary 2, $\Delta\mathsf{wr}_{1\blacktriangleright 2}(j_1, *)$ can take only two values: $\lceil T_1/T_2 \rceil T_2$, or $\lfloor T_1/T_2 \rfloor T_2$. When $\Delta\mathsf{wr}_{1\blacktriangleright 2}(j_1, *) = \lceil T_1/T_2 \rceil T_2 \geqslant T_1 = T_{\underline{2}}$, then the condition of (38) is certainly true because the period of $\tau_{\underline{2}}$ is $T_{\underline{2}} = T_1$ and there is always a read instant $\mathsf{rd}_{\underline{2}}(j_{\underline{2}})$ of some job $j_{\underline{2}}$ of $\tau_{\underline{2}}$ between $\mathsf{wr}_{1\blacktriangleright 2}(j_1, *)$ and $\mathsf{wr}_{1\blacktriangleright 2}(\mathrm{next}(j_1, *))$.

On the contrary, when $\Delta\mathsf{wr}_{1\blacktriangleright 2}(j_1, *) = \lfloor T_1/T_2 \rfloor T_2 = T_1 - \lfloor p_1 \rfloor_{p_2} G < T_1$ (from Lemma 2 and Corollary 2), some care is necessary because the read instant $\mathsf{rd}_{\underline{2}}(j_{\underline{2}})$ may fall outside the boundaries of (38) causing job $(j_1, *)$ to be overwritten by the next one before being read. We prove that if $\exists \hat{\jmath} \in \mathbb{Z}$ such that the read phase of task $\tau_{\underline{2}}$ satisfies (37), then job $j_{\underline{2}} = j_1 + \hat{\jmath}$ of $\tau_{\underline{2}}$ is the one that reads the output of job $(j_1, *) \in \mathbb{J}_{1\blacktriangleright 2}$. First, we rewrite Eq. (38) as

$$\begin{aligned} 0 \leqslant \mathsf{rd}_{\underline{2}}(j_{\underline{2}}) - \mathsf{wr}_{1\blacktriangleright 2}(j_1, *) &< \Delta\mathsf{wr}_{1\blacktriangleright 2}(j_1, *) \\ 0 \leqslant \mathsf{rd}_{\underline{2}}(j_1 + \hat{\jmath}) - \mathsf{wr}_{1\blacktriangleright 2}(j_1, *) &< \Delta\mathsf{wr}_{1\blacktriangleright 2}(j_1, *) \\ 0 \leqslant (j_1 + \hat{\jmath})T_1 + \theta^{\mathsf{r}}_{\underline{2}} - (j_1 T_1 + \theta^{\mathsf{w}}_{1\blacktriangleright 2}(j_1, *)) &< \Delta\mathsf{wr}_{1\blacktriangleright 2}(j_1, *) \\ 0 \leqslant \theta^{\mathsf{r}}_{\underline{2}} - (\theta^{\mathsf{w}}_{1\blacktriangleright 2}(j_1, *) - \hat{\jmath} T_1) &< T_1 - \lfloor p_1 \rfloor_{p_2} G. \end{aligned}$$

Then, by replacing the value of $\theta^{\mathsf{w}}_{1\blacktriangleright 2}(j_1, *)$ from Eq. (24), we obtain

$$\begin{aligned} 0 \leqslant \theta^{\mathsf{r}}_{\underline{2}} - (\theta^{\mathsf{w}}_2 - \Theta + \lfloor \Theta \rfloor_G + \lfloor \phi_1 - j_1 p_1 \rfloor_{p_2} G - \hat{\jmath} T_1) \\ < T_1 - \lfloor p_1 \rfloor_{p_2} G. \end{aligned} \tag{39}$$

We now should determine the conditions on $\theta^{\mathsf{r}}_{\underline{2}}$ that make Eq. (39) above always hold regardless of $j_1$. From the first case of Eq. (27) in Lemma 2, the case of interest with $\Delta\mathsf{wr}_{1\blacktriangleright 2}(j_1, *) = T_1 - \lfloor p_1 \rfloor_{p_2} G$ happens when $\lfloor p_1 \rfloor_{p_2} \leqslant \lfloor -j_1 p_1 + \phi_1 \rfloor_{p_2}$. The upper inequality of (39) must hold when $\lfloor -j_1 p_1 + \phi_1 \rfloor_{p_2}$ is at its lowest value $\lfloor p_1 \rfloor_{p_2}$, that is,

$$\begin{aligned} \theta^{\mathsf{r}}_{\underline{2}} - (\theta^{\mathsf{w}}_2 - \Theta + \lfloor \Theta \rfloor_G + \lfloor p_1 \rfloor_{p_2} G - \hat{\jmath} T_1) &< T_1 - \lfloor p_1 \rfloor_{p_2} G \\ \theta^{\mathsf{r}}_{\underline{2}} &< \theta^{\mathsf{w}}_2 - \Theta + \lfloor \Theta \rfloor_G + T_1 - \hat{\jmath} T_1 \end{aligned}$$

while the lower inequality of (39) must hold when $\lfloor -j_1 p_1 + \phi_1 \rfloor_{p_2} = p_2 - 1$, that is,

$$\begin{aligned} \theta^{\mathsf{r}}_{\underline{2}} - (\theta^{\mathsf{w}}_2 - \Theta + \lfloor \Theta \rfloor_G + (p_2 - 1)G - \hat{\jmath} T_1) &\geqslant 0 \\ \theta^{\mathsf{r}}_{\underline{2}} &\geqslant \theta^{\mathsf{w}}_2 - \Theta + \lfloor \Theta \rfloor_G + T_2 - G - \hat{\jmath} T_1. \end{aligned}$$

The two found conditions on $\theta^{\mathsf{r}}_{\underline{2}}$ are exactly the same as (37). Hence, if $\tau_{\underline{2}}$ conforms to the hypothesis of the theorem, no job in $\mathbb{J}_{1\blacktriangleright 2}$ is void w.r.t. $\tau_{\underline{2}}$, the period of the chain is $T_{1\blacktriangleright 2\blacktriangleright \underline{2}}$ then is $T_1$, and the chain $\tau_{1\blacktriangleright 2\blacktriangleright \underline{2}}$ has constant read phasing equal to $\theta^{\mathsf{r}}_{1\blacktriangleright 2\blacktriangleright \underline{2}} = \theta^{\mathsf{r}}_1$ and constant write phasing equal to $\theta^{\mathsf{w}}_{1\blacktriangleright 2\blacktriangleright \underline{2}} = \theta^{\mathsf{w}}_{\underline{2}}$, as required. Moreover, the job $(j_1 + \hat{\jmath}) \in \mathbb{J}_{\underline{2}}$ is the one reading from $(j_1, *)$, confirming then the expression of $\mathbb{J}_{1\blacktriangleright 2\blacktriangleright \underline{2}}$ and concluding the theorem. □

Let us now reveal the motivations behind the choice of $\tau_3$ parameters of Example 2 depicted in Figure 9. For these parameters $\theta^{\mathsf{w}}_2 - \Theta + \lfloor \Theta \rfloor_G = 9$ and the condition on the read phase $\theta^{\mathsf{r}}_3$ of Eq. (37) which guarantees that $\tau_{1\blacktriangleright 2\blacktriangleright 3}$ has period equal to $T_1$ and constant phasings, is

$$\exists \hat{\jmath} \in \mathbb{Z}, \quad 9 + \underbrace{3}_{T_2 - G} - \hat{\jmath} \underbrace{5}_{T_1} \leqslant \theta^{\mathsf{r}}_3 < 9 + \underbrace{5}_{T_1} - \hat{\jmath} 5.$$

The task $\tau_3$ of Figure 9 has phasing $\theta^{\mathsf{r}}_3 = 2$, which satisfies the constraint above with $\hat{\jmath} = 2$.

When $T_2 \geqslant T_1$ an analogous theorem holds, by adding a task prior to $\tau_1$.

**Theorem 5.** *Given a chain $\tau_{1\blacktriangleright 2}$ of LET tasks, with $T_2 \geqslant T_1$, if the LET task $\tau_{\underline{1}}$ with*

- *period $T_{\underline{1}} = T_2$,*
- *write phasing $\theta^{\mathsf{w}}_{\underline{1}}$ such that $\exists \hat{\jmath} \in \mathbb{Z}$:*

$$\begin{cases} \theta^{\mathsf{w}}_{\underline{1}} > \theta^{\mathsf{r}}_1 + \Theta - \lfloor \Theta \rfloor_G - T_2 - \hat{\jmath} T_2 \\ \theta^{\mathsf{w}}_{\underline{1}} \leqslant \theta^{\mathsf{r}}_1 + \Theta - \lfloor \Theta \rfloor_G - T_1 + G - \hat{\jmath} T_2 \end{cases} \tag{40}$$

- *and any read phasings $\theta^{\mathsf{r}}_{\underline{1}} \leqslant \theta^{\mathsf{w}}_{\underline{1}}$*

*writes at the head of $\tau_{1\blacktriangleright 2}$, then the chain $\tau_{\underline{1}\blacktriangleright 1\blacktriangleright 2}$ has:*

- *period $T_{\underline{1}\blacktriangleright 1\blacktriangleright 2}$ equal to $T_2$,*

- *constant read and write phasings, and*
- *set of jobs*

$$\mathbb{J}_{\underline{1}\blacktriangleright 1\blacktriangleright 2} = \left\{ \left( j_2 + \hat{\jmath}, \left\lfloor \frac{j_2 T_2 + \Theta}{T_1} \right\rfloor, j_2 \right) : j_2 \in \mathbb{Z} \right\}.$$

*Proof.* When $\Delta\mathsf{rd}_{1\blacktriangleright 2}(*, j_2) \geqslant T_2$ then job $\mathrm{next}(*, j_2)$ is not redundant w.r.t. $\tau_{\underline{1}}$. We are now verifying that jobs are not redundant even when $\Delta\mathsf{rd}_{1\blacktriangleright 2}(*, j_2) < T_2$.

When $\Delta\mathsf{rd}_{1\blacktriangleright 2}(*, j_2) = \lfloor T_2/T_1 \rfloor T_1 = T_2 - \lfloor p_2 \rfloor_{p_1} G < T_2$ then the condition to have no redundant jobs is that $\exists j_{\underline{1}} \in \mathbb{Z}$ such that

$$\mathsf{rd}_{1\blacktriangleright 2}(*, j_2) < \mathsf{wr}_{\underline{1}}(j_{\underline{1}}) \leqslant \mathsf{rd}_{1\blacktriangleright 2}(*, j_2) + \Delta\mathsf{rd}_{1\blacktriangleright 2}(*, j_2)$$

We aim at showing that if the phasings are chosen in accordance to (40), then the job of $\tau_{\underline{1}}$ ensuring the inequality above is $j_{\underline{1}} = j_2 + \hat{\jmath} + 1$. In fact,

$$\begin{aligned} 0 < \mathsf{wr}_{\underline{1}}(j_2 + \hat{\jmath} + 1) - \mathsf{rd}_{1\blacktriangleright 2}(*, j_2) &\leqslant \Delta\mathsf{rd}_{1\blacktriangleright 2}(*, j_2) \\ 0 < (j_2 + \hat{\jmath} + 1)T_2 + \theta^{\mathsf{w}}_{\underline{1}} - (j_2 T_2 + \theta^{\mathsf{r}}_{1\blacktriangleright 2}(*, j_2)) &\leqslant \Delta\mathsf{rd}_{1\blacktriangleright 2}(*, j_2) \\ 0 < \theta^{\mathsf{w}}_{\underline{1}} - (\theta^{\mathsf{r}}_{1\blacktriangleright 2}(*, j_2) - (\hat{\jmath} + 1)\, T_2) &\leqslant T_2 - \lfloor p_2 \rfloor_{p_1} G \end{aligned}$$

and by replacing the value of $\theta^{\mathsf{r}}_{1\blacktriangleright 2}(*, j_2)$ of (33), we find

$$\begin{aligned} 0 < \theta^{\mathsf{w}}_{\underline{1}} - (\theta^{\mathsf{r}}_1 + \Theta - \lfloor \Theta \rfloor_G - \lfloor j_2 p_2 + \phi_2 \rfloor_{p_1} G - (\hat{\jmath} + 1)T_2) \\ \leqslant T_2 - \lfloor p_2 \rfloor_{p_1} G. \quad (41) \end{aligned}$$

As in the proof of Theorem 4, we aim at ensuring that the value for $\theta^{\mathsf{w}}_{\underline{1}}$ of (40) makes (41) true regardless of the fluctuations of $\lfloor j_2 p_2 + \phi_2 \rfloor_{p_1}$ through $j_2$. The upper inequality of (41) must hold when $\lfloor j_2 p_2 + \phi_2 \rfloor_{p_1}$ is at its largest value, i.e., $\lfloor -p_2 \rfloor_{p_1} - 1$, from the second case of (36) in Lemma 3

$$\begin{aligned} \theta^{\mathsf{w}}_{\underline{1}} - (\theta^{\mathsf{r}}_1 + \Theta - \lfloor \Theta \rfloor_G - (\lfloor -p_2 \rfloor_{p_1} - 1)G - (\hat{\jmath} + 1)T_2) \leqslant \\ \leqslant T_2 - \lfloor p_2 \rfloor_{p_1} G \\ \theta^{\mathsf{w}}_{\underline{1}} \leqslant \theta^{\mathsf{r}}_1 + \Theta - \lfloor \Theta \rfloor_G - (\lfloor p_2 \rfloor_{p_1} + \lfloor -p_2 \rfloor_{p_1} - 1)G - \hat{\jmath}\, T_2 \\ \theta^{\mathsf{w}}_{\underline{1}} \leqslant \theta^{\mathsf{r}}_1 + \Theta - \lfloor \Theta \rfloor_G - T_1 + G - \hat{\jmath}\, T_2, \end{aligned}$$

in which we exploited the property that $\lfloor p_2 \rfloor_{p_1} + \lfloor -p_2 \rfloor_{p_1} = p_1$. The lower inequality of (41) must be true when $\lfloor j_2 p_2 + \phi_2 \rfloor_{p_1} = 0$, that is,

$$\begin{aligned} 0 < \theta^{\mathsf{w}}_{\underline{1}} - (\theta^{\mathsf{r}}_1 + \Theta - \lfloor \Theta \rfloor_G - (\hat{\jmath} + 1)T_2) \\ \theta^{\mathsf{w}}_{\underline{1}} > \theta^{\mathsf{r}}_1 + \Theta - \lfloor \Theta \rfloor_G - (\hat{\jmath} + 1)T_2 \end{aligned}$$

The two found conditions coincide with (40). Hence, the choice of $\theta^{\mathsf{w}}_{\underline{1}}$ in accordance to (40) guarantees that job $(*, j_2 + 1) \in \mathbb{J}_{1\blacktriangleright 2}$ always read the data written by job $(j_2 + \hat{\jmath} + 1) \in \mathbb{J}_{\underline{1}}$, which demonstrates the expression for $\mathbb{J}_{\underline{1}\blacktriangleright 1\blacktriangleright 2}$ and closes the proof. □

We conclude by showing that a proper addition of two copier tasks can also regularize the chain $\tau_{1\blacktriangleright 2\blacktriangleright 3}$ of Figure 7 with task periods $T_1 = 5$, $T_2 = 3$, and $T_3 = 4$.

From Theorem 5, the introduction of a task $\tau_{\underline{2}}$ writing data to $\tau_2$ with period $T_{\underline{2}} = T_3 = 4$ and write phasing such that

$$-7 - \hat{\jmath}\, T_3 < \theta^{\mathsf{w}}_{\underline{2}} \leqslant -5 - \hat{\jmath}\, T_3$$

for some $\hat{\jmath} \in \mathbb{Z}$, makes $\tau_{\underline{2}\blacktriangleright 2\blacktriangleright 3}$ a periodic LET chain. The choice of $\theta^{\mathsf{w}}_{\underline{2}} = \theta^{\mathsf{r}}_{\underline{2}} = 3$ gives the following parameters

$$T_{\underline{2}\blacktriangleright 2\blacktriangleright 3} = T_3 = 4, \qquad \theta^{\mathsf{r}}_{\underline{2}\blacktriangleright 2\blacktriangleright 3} = 3, \qquad \theta^{\mathsf{w}}_{\underline{2}\blacktriangleright 2\blacktriangleright 3} = 12.$$

Let us now consider the chain made by concatenating $\tau_1$ with $\tau_{\underline{2}\blacktriangleright 2\blacktriangleright 3}$. Theorem 4 states that by appending $\tau_{\underline{3}}$ with period $T_{\underline{3}} = T_1 = 5$ and $\theta^{\mathsf{r}}_{\underline{3}}$ such that

$$12 - \hat{\jmath}\, T_1 \leqslant \theta^{\mathsf{r}}_{\underline{3}} < 14 - \hat{\jmath}\, T_1$$

for some integer $\hat{\jmath}$, the period of the whole chain is equal to $T_1 = 5$. For example, the choice of $\theta^{\mathsf{r}}_{\underline{3}} = \theta^{\mathsf{w}}_{\underline{3}} = 2$ produces a chain $\tau_{1\blacktriangleright \underline{2}\blacktriangleright 2\blacktriangleright 3\blacktriangleright \underline{3}}$ with period equal to $T_1 = 5$ and phasings

$$\theta^{\mathsf{r}}_{1\blacktriangleright \underline{2}\blacktriangleright 2\blacktriangleright 3\blacktriangleright \underline{3}} = 0, \quad \theta^{\mathsf{w}}_{1\blacktriangleright \underline{2}\blacktriangleright 2\blacktriangleright 3\blacktriangleright \underline{3}} = 17.$$

Hence, the application of the method described in this section can transform any chain $\tau_{1\blacktriangleright 2\blacktriangleright \cdots \blacktriangleright n}$ of $n$ tasks into a periodic LET chain with period $T_{1\blacktriangleright 2\blacktriangleright \cdots \blacktriangleright n} = \max_i\{T_i\}$ and constant read/write phasings, at the price of adding at most $n - 1$ copier tasks.

# 7 Conclusion

This paper proposes a full analysis for a pair of functionally-dependent tasks, in which the first task writes input data that is read by the second one. Explicit expressions for the input-output latency, minimum/maximum delays are given. Such a characterization enables a method to transform any task chain into another one with maximum throughput and constant read/write phasings.

# References


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

[2] 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, 2017.

[3] M. Becker, D. Dasari, S. Mubeen, M. Behnam, and T. Nolte. Analyzing end-to-end delays in automotive systems at various levels of timing information. *ACM SIGBED Review*, 14(4), 2018.

[4] M. Beckert, M. Möstl, and R. Ernst. Zero-time communication for automotive multi-core systems under SPP scheduling. In *Conference on Emerging Technologies and Factory Automation (ETFA)*, 2016.

[5] A. Biondi and M. Di Natale. Achieving predictable multicore execution of automotive applications using the let paradigm. In *Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2018.

[6] T. Blaß, D. Casini, S. Bozhko, and B. B. Brandenburg. A ros 2 response-time analysis exploiting starvation freedom and execution-time variance. In *2021 IEEE Real-Time Systems Symposium (RTSS)*, pages 41–53. IEEE, 2021.

[7] A. Davare, Q. Zhu, M. Di Natale, C. Pinello, S. Kanajan, and A. Sangiovanni-Vincentelli. Period optimization for hard real-time distributed automotive systems. In *Design Automation Conference (DAC)*, 2007.

[8] M. Dürr, G. V. D. Brüggen, K.-H. Chen, and J.-J. Chen. End-to-end timing analysis of sporadic cause-effect chains in distributed systems. *Transactions on Embedded Computing Systems*, 18(5), 2019.

[9] 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 *Real-Time Systems Symposium (RTSS)*, 2009.

[10] R. Gerber, S. Hong, and M. Saksena. Guaranteeing real-time requirements with resource-based calibration of periodic processes. *Transactions on Software Engineering*, 21(7), 1995.

[11] A. Girault, C. Prévot, S. Quinton, R. Henia, and N. Sordon. Improving and estimating the precision of bounds on the worst-case latency of task chains. *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 37(11), 2018.

[12] P. Gohari, M. Nasri, and J. Voeten. Data-age analysis for multi-rate task chains under timing uncertainty. In *Proceedings of the 30th International Conference on Real-Time Networks and Systems*, RTNS 2022, page 24–35, New York, NY, USA, 2022. Association for Computing Machinery.

[13] 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 *Real Time and Embedded Technology and Applications Symposium (RTAS)*, 2021.

[14] A. Hamann, D. Dasari, S. Kramer, M. Pressler, and F. Wurst. Communication centric design in complex automotive embedded systems. In *Euromicro Conference on Real-Time Systems (ECRTS)*, 2017.

[15] T. A. Henzinger, B. Horowitz, and C. M. Kirsch. Giotto: A time-triggered language for embedded programming. *Proceedings of the IEEE*, 91(1), 2003.

[16] 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 *Real Time and Embedded Technology and Applications Symposium (RTAS)*, 2021.

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

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

[19] M. Lauer, F. Boniol, C. Pagetti, and J. Ermont. End-to-end latency and temporal consistency analysis in networked real-time systems. *Journal of Critical Computer-Based Systems*, 5(3-4), 2014.

[20] S. Liu, J. Tang, Z. Zhang, and J.-L. Gaudiot. Computer architectures for autonomous driving. *Computer*, 50(8):18–25, 2017.

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

[22] J. Martinez, I. Sañudo, and M. Bertogna. End-to-end latency characterization of task communication models for automotive systems. *Real-Time Systems*, 56(3), 2020.

[23] P. Pazzaglia, D. Casini, A. Biondi, and M. Di Natale. Optimal memory allocation and scheduling for dma data transfers under the let paradigm. In *Design Automation Conference (DAC)*, 2021.

[24] 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*, 2022.

[25] P. Pazzaglia and M. Maggio. Characterising the effect of deadline misses on time-triggered task chains. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2022.

[26] A. Sangiovanni-Vincentelli, P. Giusto, C. Pinello, W. Zheng, and M. Di Natale. Optimizing end-to-end latencies by adaptation of the activation events in distributed automotive systems. In *Real Time and Embedded Technology and Applications Symposium (RTAS)*, 2007.

[27] J. Schlatow and R. Ernst. Response-time analysis for task chains in communicating threads. In *2016 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 1–10, 2016.

[28] J. Schlatow, M. Mostl, S. Tobuschat, T. Ishigooka, and R. Ernst. Data-age analysis and optimisation for cause-effect chains in automotive control systems. In *Symposium on Industrial Embedded Systems (SIES)*, 2018.

[29] A. Tampuu, T. Matiisen, M. Semikin, D. Fishman, and N. Muhammad. A survey of end-to-end driving: Architectures and training methods. *IEEE Transactions on Neural Networks and Learning Systems*, 33(4):1364–1384, 2022.

[30] 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.

[31] Y. Zhao, V. Gala, and H. Zeng. A unified framework for period and priority optimization in distributed hard real-time systems. *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018.

## ACKNOWLEDGMENTS

The authors would like to thank the reviewers for the very detailed comments and suggestions and the editor for overseeing smoothly the overall process.

This research is partially supported by the Spoke “FutureHPC & BigData” of the ICSC — Centro Nazionale di Ricerca in “High Performance Computing, Big Data and Quantum Computing”, funded by European Union — NextGenerationEU. Also, it has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 871259 (ADMORPH project). This publication reflects only the authors’ view and the European Commission is not responsible for any use that may be made of the information it contains.

**Enrico Bini** is Associate Professor at University of Turin. Until 2016, he was with Scuola Superiore Sant’Anna, Pisa, Italy. In 2012-14, he was Marie-Curie IEF fellow at Lund University (Sweden), Department of Automatic Control. In 2004, he received the PhD on Real-Time Systems at Scuola Superiore Sant’Anna (receiving the “Spitali Award” for best PhD thesis of the whole university). In January 2010, he also completed a Master degree in Mathematics with a thesis on optimal sampling for linear systems.

**Paolo Pazzaglia** is a Research Engineer at Robert Bosch GmbH, Corporate Research, Renningen. From 2020 to 2022, he was a postdoctoral researcher at the Computer Science Department at Saarland University. He completed his Ph.D. in 2020 at the ReTiS Lab at Scuola Superiore Sant’Anna in Pisa. He was a visiting Ph.D. student at the Department of Automatic Control, Lund University in the winter semester 2018/19.

**Martina Maggio** is a Professor at the Computer Science Department, Saarland University (since 2020) and an Associate Professor at the Department of Automatic Control, Lund University (since 2017). She completed her Ph.D. at Politecnico di Milano (in February 2012), and was a visiting graduate student at Massachussetts Institute of Technology (in 2010 and 2011). In 2019, she spent a sabbatical year at Bosch Corporate Research.