



## OPEN ACCESS

## EDITED BY

Ivan Cibrario Bertolotti,  
National Research Council (CNR), Italy

## REVIEWED BY

Tihomir Opetuk,  
University of Zagreb, Croatia

## \*CORRESPONDENCE

Paulo Garcia,  
✉ paulo.g@chula.ac.th

RECEIVED 25 July 2023

ACCEPTED 03 November 2023

PUBLISHED 23 November 2023

## CITATION

Garcia P and Sritriratanarak W (2023), We need a theoretical framework for the modernization of industrial legacy systems.  
*Front. Ind. Eng.* 1:1266651.  
doi: 10.3389/fieng.2023.1266651

## COPYRIGHT

© 2023 Garcia and Sritriratanarak. This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# We need a theoretical framework for the modernization of industrial legacy systems

Paulo Garcia<sup>1\*</sup> and Warisa Sritriratanarak<sup>2</sup>

<sup>1</sup>International School of Engineering, Chulalongkorn University, Bangkok, Thailand, <sup>2</sup>Department of Computer Engineering, Chulalongkorn University, Bangkok, Thailand

Industrial informatics brings computational intelligence to industry, powering the “software-ization” of manufacturing processes. However, when faced with the myriad of legacy systems that cannot be fully replaced cost-effectively, practitioners must retrofit computational intelligence into legacy systems. This modernization of legacy industrial systems is deceptively challenging: poor retrofitting can cause more harm than good, hindering overall metrics. We argue for a theoretical framework for modernizing legacy industrial systems. We illustrate the challenge within the context of the real-time performance of industrial cyber-physical systems by depicting a formalization of the problem and illustrating its impact through Monte Carlo methods. We show how knowledge of extant system internals constrains possible optimizations. We conclude by highlighting several research directions, including some recommendations, that must be pursued to establish a common theoretical underpinning that can inform practitioners.

## KEYWORDS

industrial informatics, real-time, cyber-physical, legacy, retrofitting

## 1 Introduction

Industrial informatics (Bi, 2017) is a field that broadly consists of applying computing technologies to industrial environments (Lee et al., 2014). Its goal, within the context of Industry 4.0 (Lasi et al., 2014), is to move away from traditional, fully electro-mechanical systems toward cyber-components, that is, software-controlled industrial processes (Parri et al., 2021). This has several advantages over established practices, including safer and more secure environments, thanks to new monitoring technologies (Zhou and Peng, 2020); resilience and fault-tolerance (Gao et al., 2015); distributed and remote control and operation (Buse and Wu, 2004); and improved efficiency, thanks to the integration of optimization algorithms and artificial intelligence (Gaudio et al., 2021). The practice of developing industrial informatics is fairly codified in the literature, starting from industrial internet of things (Butun et al., 2020) to the broader umbrella of Industry 4.0 (Ghobakhloo, 2020).

However, one key aspect the literature lacks in the deployment of industrial informatics is a comprehensive theoretical framework (and associated tooling) for how to modernize extant industries—retrofitting state-of-the-art industrial informatics into legacy systems. In the industrial context, blank-slate designs are seldom, if ever, the case; rather, (costly) legacy equipment is augmented to meet the goals of upgrades, requiring practitioners to employ *ad hoc* approaches to the scenario at hand (Lins et al., 2017).

Examples of such challenges can be found in the literature, including Ben-Guang et al. (2000), Hills et al. (2016), Guerreiro et al. (2018), Lins and Oliveira (2020), and Tran et al.

(2022). Unfortunately, almost all are confined in scope to the specific environment in which the authors were working, reflecting the lack of, and need for, a more holistic understanding of retrofitting practices in the discipline to help bring new technology to the industry (Fryer and Garcia, 2020).

Therefore, in alignment with the authors' experience with retrofitting real-time systems (Tavares et al., 2012; Gomes et al., 2015), this paper presents a perspective on the problem regarding real-time operation which attempts to begin the scholarly development of such a comprehensive framework. We offer the following contributions:

- Formalizing the problem of retrofitting real-time (cyber-physical) industrial systems with computational intelligence as a minimization problem.
- Illustrating that many free terms in this formalism are not easily obtainable, nor reasonable bounds easily estimated, constituting the fundamental problem of dealing with legacy systems.
- Demonstrating the potential negative impact of poorly chosen approaches through a Monte Carlo simulation method.
- Deriving how knowledge of the internals of extant systems constrains possible optimization.
- Providing some insights into what is required to inform these practices in the future.

## 2 Retrofitting legacy systems: a model

### 2.1 Legacy execution

Consider an industrial environment with  $N$  legacy systems. Prior to the integration of computational intelligence, each system  $S_i$ ,  $i \in [0, N[$  is running a set of  $T_i$  tasks, where  $t_{i,j}$  denotes task index  $j$  in set  $T_i$ , and  $j \in [0, f(i) - 1[$  for each system, where  $f(i)$  is the number of total tasks in the function of a system (i.e., each system runs an arbitrarily different number of tasks), with  $e(t_{i,j})$  denoting the execution time of task  $t_{i,j}$ . For simplicity, we assume that each set  $T_i$  consists exclusively of periodic tasks, where  $p(t_{i,j})$  denotes the period of task  $t_{i,j}$ —expressed as the percentage of task execution over a total unit of work for a system, such that  $\frac{1}{p(t_{i,j})}$  represents the number of times the task executes per total unit of work. In the authors' professional experience, this is generally true, with the exception of asynchronous fault-handling tasks. This model excludes these tasks, as, in the event of faults, handling is of higher priority (and, typically, safety-critical); thus, regular scheduling is typically paused. It is certainly possible to incorporate asynchronous tasks in the model, but the increased complexity is beyond the scope of this paper and is reserved for the future.

If all tasks are periodic, optimum scheduling (if feasible) can be achieved using rate monotonic scheduling (RMS) (Baruah and Goossens, 2003)). Assuming a full unit of work is achieved whenever the task with the longest period executes, then the total time required per unit of work for a system  $S_i$ , denoted  $e(S_i)$ , is given by

$$e(S_i) = \sum_{k=0}^{f(i)-1} \frac{e(t_{i,k})}{P(t_{i,k})}. \quad (1)$$

It will be relevant for subsequent sections to note that, in many cases,  $\exists t_{i,j} \in T_i$  such that  $e(S_i) \gg e(t_{i,j})$ .

### 2.2 Retrofitting intelligence

Now let us consider the integration of computational intelligence on a legacy industrial system. Each system  $S_i$  is equipped with additional functionality, responsible for the following operations:

1. Monitoring internal operation, including several diagnostic and performance metrics.
2. Communicating that information to a centralized computational intelligence, responsible for using that data to obtain actionable information, make global optimizations, and refer updates back to each system.
3. Receiving updates from the central system and updating the system's internal operation accordingly.

A notable example of this sort of operation is workload balancing. For example, in simple serial manufacturing systems, effective throughput is dominated by the system with the lowest individual throughput. Thus, when any one system decreases its throughput (due to supply availability, maintenance operation, etc.), central intelligence can update other systems in that serial chain to match the lowest system's throughput, decreasing unnecessary operation that could lead to costs and lifetime degradation.

In more complex systems, especially when other metrics other than just throughput are being considered, more sophisticated updates can be performed to meeting several distinct goals.

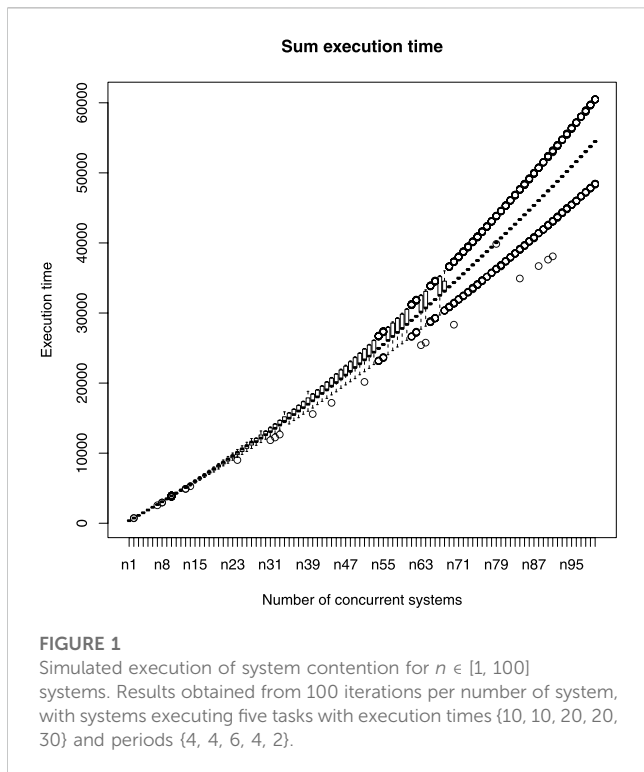
In a system with augmented intelligence, denoted  $S'_i$ , this additional functionality is contained in a new task  $t_{i,c}$ , such that each system is now running a new set of tasks  $T'_i = T_i \cup t_{i,c}$  with corresponding execution time and period. New total time required per unit of work  $e(S'_i)$  is now given by

$$e(S'_i) = \sum_{k=0}^{f(i)-1} \frac{e(t_{i,k})}{P(t_{i,k})} + \frac{e(t_{i,c})}{P(t_{i,c})} > e(S_i) = \sum_{k=0}^{f(i)-1} \frac{e(t_{i,k})}{P(t_{i,k})}. \quad (2)$$

Note that, in the period of  $t_{i,c}$ ,  $p(t_{i,c})$  constrains the level of temporal granularity upon which system  $S_i$  can be measured and acted upon. A short period offers fine-grained measurement and control, at the expense of potentially severe performance degradation; conversely, a long period has a minimal effect on performance, at the expense of coarse-grained operation. For every system, there is likely a minimum period  $p_{min}^i$ , below which actuation is no longer feasible.

### 2.3 Global scheduling

Synchronizing distributed systems is notoriously challenging. In the case of industrial systems at scale, the optimum scenario would be all tasks  $t_{i,c}$ ,  $i \in [0, N[$  running at different times, such that there is no contention for the communication medium nor for the central intelligence compute power. In practice, there is significant



**FIGURE 1** Simulated execution of system contention for  $n \in [1, 100]$  systems. Results obtained from 100 iterations per number of system, with systems executing five tasks with execution times {10, 10, 20, 20, 30} and periods {4, 4, 6, 4, 2}.

overlap, resulting in an execution time for each task  $t_{i,c}$  given by  $e'(t_{i,j}) = e(t_{i,j}) + \lambda(x)$ ,  $x \in [0, N[$ , where  $\lambda(x)$  represents delay as a function of  $x$  concurrently executing tasks, modeling medium and compute power contention.

With preliminaries outlined, we can finally formalize the overarching problem:

$$\min_{p(t_{n,c}), \forall n \in [0, N[} \sum_{n=0}^{i-1} e(S'_n), \text{ subject to: } p(t_{n,c}) < p_{min}^n, \forall n \in [0, N[ \quad (3)$$

or, in full,

$$\min_{p(t_{n,c}), \forall n \in [0, N[} \sum_{n=0}^{i-1} \sum_{k=0}^{f(n)-1} \frac{e(t_{n,k})}{p(t_{n,k})} + \frac{e(t_{n,c}) + \lambda(x)}{p(t_{n,c})}, \quad (4)$$

subject to:  $p(t_{n,c}) < p_{min}^n, \forall n \in [0, N[$ .

Note that this is not an exclusive formalization: we have chosen to use sum minimization (arguably, the simplest problem) when we could have chosen mean, median, or several other norms as the metric to minimize. The specific choice is highly domain-specific, depending on the needs of the specific global system. The goal here is to illustrate that the optimization problem is not particularly difficult from a mathematical perspective but that un-optimized systems can be severely affected by a poor choice of parameters and, critically, the difficulties arise from effectively determining  $p(t_{i,j})$  and finding realistic bounds for  $\lambda(x)$ .

### 2.4 Simulated evaluation

To illustrate the impact of this challenge, consider the following simulation implemented as a Monte Carlo method.

**TABLE 1** Average execution time and standard deviation per system.

Number of concurrent systems	Mean	$\sigma$
1	27.544	5.649
2	30.179	3.844
3	31.379	3.876
4	33.888	4.275
5	37.349	3.881
6	38.969	4.011
7	41.707	4.289
8	43.408	4.377
9	46.212	4.579
10	47.921	4.671

We implement 100 scenarios, where the number of systems is sequentially increased, such that, for each scenario, there are  $S_i, i \in [0, N[$  systems where  $N$  is the scenario number. In each scenario, each system executes  $f(i)$  tasks, with corresponding periods and execution times. For simplicity, these values are heuristics and are kept consistent across scenarios and homogeneously across systems.

For each scenario, we introduce a computational intelligence task per system, whose execution time and periods are random variables drawn from a normal distribution centered at a positive integer, left-truncated at 1 and right-truncated at the smallest execution time and highest period of the system, respectively, rounded to the nearest integer. The delay  $\lambda(x)$  is approximated as the percentage of concurrent execution time, as given by the smallest execution time of all computational tasks multiplied by the highest period—a percentage of the amount of time the system will exhibit contention (Carley et al., 2003). We repeated this random experiment 100 times per scenario, reporting the sum of total execution times for all systems (the function to be minimized in Eq. 4), with results depicted in Figure 1. Table 1 shows the corresponding mean and standard deviation for the average system.

### 3 Leveraging global intelligence

Once systems are retrofitted with computational intelligence, a central coordinating system can exploit this additional information and actuation power to perform global optimizations (which is, of course, the motivation behind retrofitting in the first place). Regardless of what precise optimization is performed, new system sum execution time  $e(total)$  can be modeled as follows:

$$e(total) = \alpha \left( \sum_{n=0}^{i-1} \sum_{k=0}^{f(n)-1} \frac{e(t_{n,k})}{p(t_{n,k})} + \frac{e(t_{n,c}) + \lambda(x)}{p(t_{n,c})} \right), \alpha \in ]0, 1], \quad (5)$$

where  $\alpha$  is the inverse of the optimization factor, assuming this is always positive (a negative optimization factor would correspond to performance degradation).

The overarching problem can be formally expressed as satisfying

$$\alpha \left( \sum_{n=0}^{i-1} \sum_{k=0}^{f(n)-1} \frac{e(t_{n,k})}{p(t_{n,k})} + \frac{e(t_{n,c}) + \lambda(x)}{p(t_{n,c})} \right) < \sum_{n=0}^{i-1} \sum_{k=0}^{f(n)-1} \frac{e(t_{n,k})}{p(t_{n,k})}, \alpha \in ]0, 1], \tag{6}$$

which simplifies into

$$\frac{\alpha}{1 - \alpha} \left( \sum_{n=0}^{i-1} \frac{e(t_{n,c}) + \lambda(x)}{p(t_{n,c})} \right) < \sum_{n=0}^{i-1} \sum_{k=0}^{f(n)-1} \frac{e(t_{n,k})}{p(t_{n,k})}, \alpha \in ]0, 1]. \tag{7}$$

The sum of execution times of computational tasks  $\sum_{n=0}^{i-1} \frac{e(t_{n,c})}{p(t_{n,c})}$  is approximately constant for a given value of  $\alpha = \alpha_a$ , such that we can re-formulate the aforementioned statement as

$$\frac{\alpha}{1 - \alpha} (a + i\lambda(x)) < \sum_{n=0}^{i-1} \sum_{k=0}^{f(n)-1} \frac{e(t_{n,k})}{p(t_{n,k})}, \alpha = \alpha_a. \tag{8}$$

Borrowing from scheduling theory (Sinnen and Sousa, 2005; Blagodurov et al., 2010; Johnson et al., 2010), it is safe to assume there is a lower bound (typically non-0 but sufficiently close to 0) for contention as a function of phase choices for the execution of computational tasks (minimizing alignment of periodic execution). An optimal formulation  $F$  for choosing these phases is predicated on knowledge of periods and execution times of all extant tasks across systems, formulated as  $K(S_i), i \in [0, N], K(S_i) \in [0, 1]$ , such that

$$\lambda(x) = \frac{1}{F \circ K(S_i)}, \lambda(x) \in [\lambda_{min}, \infty[, \tag{9}$$

where  $F \circ K(S_i)$  denotes the function composition of  $F$  over  $K(S_i)$  equivalent to  $F(K(S_i))$  (note that contention is typically modeled as having an infinite upper bound to denote system deadlock). Replacing  $\lambda(x)$  in Eq. 8 and solving for  $K(S_i)$ , we obtain

$$\frac{\alpha}{1 - \alpha} \left( a + \frac{i}{F \circ K(S_i)} \right) < \sum_{n=0}^{i-1} \sum_{k=0}^{f(n)-1} \frac{e(t_{n,k})}{p(t_{n,k})}, \alpha = \alpha_a, \tag{10}$$

$$\equiv \frac{i}{F \circ K(S_i)} < \left( \frac{1 - \alpha}{\alpha} \sum_{n=0}^{i-1} \sum_{k=0}^{f(n)-1} \frac{e(t_{n,k})}{p(t_{n,k})} \right) - a, \alpha = \alpha_a, \tag{11}$$

$$\equiv F \circ K(S_i) > \frac{i}{\left( \frac{1 - \alpha}{\alpha} \sum_{n=0}^{i-1} \sum_{k=0}^{f(n)-1} \frac{e(t_{n,k})}{p(t_{n,k})} \right) - a}, \alpha = \alpha_a, \tag{12}$$

$$\equiv K(S_i) > F^{-1} \left( \frac{i}{\left( \frac{1 - \alpha}{\alpha} \sum_{n=0}^{i-1} \sum_{k=0}^{f(n)-1} \frac{e(t_{n,k})}{p(t_{n,k})} \right) - a} \right), \alpha = \alpha_a. \tag{13}$$

Thus, showing that total knowledge about the tasks of extant systems required to optimize them when retrofitting computational intelligence is a function of a possible optimization factor (conversely, that optimization is a function of prior knowledge). Thus, it is important to develop methodologies, theories, and tools to determine system behavior (Section 5).

### 4 Case study

To illustrate the aspects considered in the model thus far, consider the example industrial system depicted in Figure 2. The

two parallel systems produce items at different rates, depending on operational parameters. The robotic arm picks items from the intermediate item buffers and transfers them to subsequent stations (not depicted) for further processing.

Through modernization efforts, all systems have been connected to central intelligence through a communication medium, such as an industrial ethernet. New computational tasks have been added to each system; the goal is to communicate real-time production rates to central intelligence so that it can optimize robotic arm operation by moving it as little as possible (preserving power and potentially using it for other operations in the downtime) as long as item buffer rates are not exceeded.

Figure 3 depicts timing behavior before modernization (top); parallel systems produce items at different rates, depicted in blue and red, respectively, where a hyper-period corresponds to three executions of the first system and two executions of the second. In this example, assuming buffer capacity is greater than 1, and executing a robotic arm operation as frequently as the production rate of the first system is wasteful.

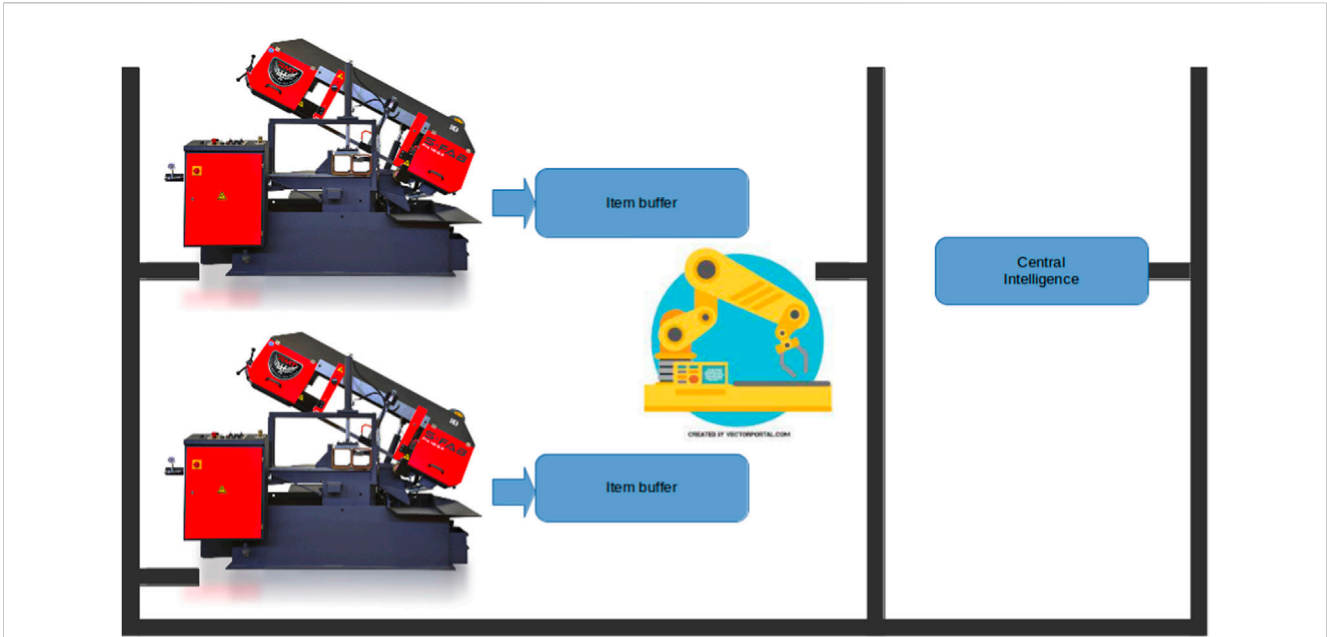
An incorrect approach to computational task execution is shown in the middle of Figure 3, where the computational task is depicted in green. Here, periods in both systems are chosen the same as production tasks, resulting in both systems attempting to access the shared medium at the same time, resulting in contention. Total execution time is thus extended, delaying production. On the other hand, a better chosen set of periods results in the behavior depicted in Figure 3 (bottom), where no contention is exhibited, resulting in non-decreased performance, whilst preserving the advantages of optimized robotic arm operation. Formally, systems 1 and 2 contain a single task, with periods  $\frac{1}{p(t_{i,j})}$  equal to  $\frac{1}{p(t_{1,0})} = 3$  and  $\frac{1}{p(t_{2,0})} = 2$ , respectively, and execution times  $e(t_{i,j})$  equal to  $e(t_{1,0}) = 1$  and  $e(t_{2,1}) = 2$ , respectively. From Figure 3,  $\lambda(2) = 1$  for the poorly selected case; thus, we can replace variables in Eq 4 with:

$$\begin{aligned} \min_{p(t_{n,c}), \forall n \in [0,2]} & \sum_{n=0}^0 \sum_{k=0}^0 \frac{e(t_{n,k})}{p(t_{n,k})} + \frac{e(t_{n,c}) + \lambda(x)}{p(t_{n,c})} \\ & = 1 \times 3 + 2 \times 2 + \min_{p(t_{n,c}), \forall n \in [0,2]} \left\{ (1 \times 2 + 1 \times 1), \right. \\ & \left. \left( \left( 1 + \frac{\lambda(x)}{3} \right) \times 3 + \left( 1 + \frac{\lambda(x)}{2} \right) \times 2 \right) \right\} \\ & = 7 + 3 = 10. \end{aligned} \tag{14}$$

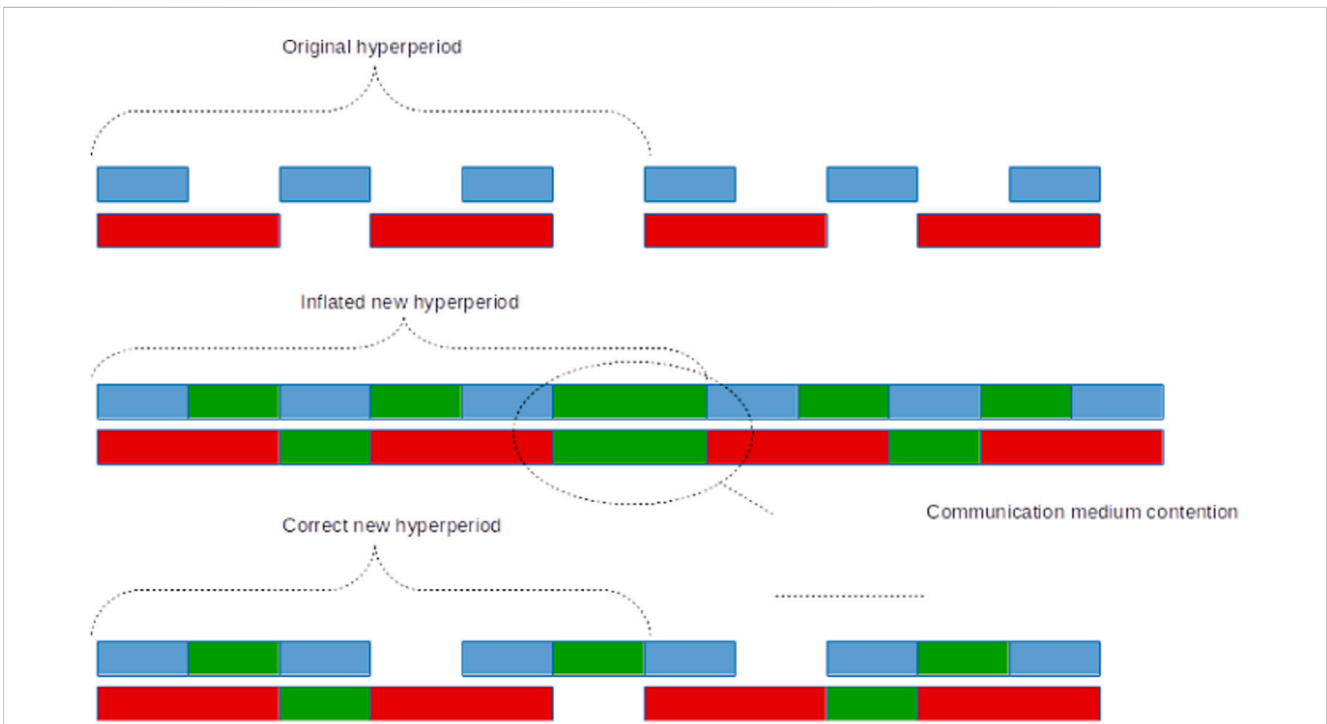
This shows that selecting periods for the computational tasks equal to half the rate of the original tasks (and out of phase with each other) results in no performance degradation. Critically, the values in this case study can be verified experimentally and fairly simply; in more complex systems, it would be difficult to estimate  $\lambda(x)$  and  $\frac{1}{p(t_{i,j})}$  without significant profiling.

### 5 Challenges and directions

Our simulated example, despite its simplicity, already clearly illustrates the pitfalls of poor retrofitting of legacy systems. With stochastically generated execution times and periods of computational tasks, total sum execution time increases



**FIGURE 2**  
 Example of a manufacturing system. Two parallel production systems output items to intermediate buffers at different rates. A moving system transfers items to next processing stage. Solid gray lines represent communication medium (e.g., industrial ethernet).



**FIGURE 3**  
 Task execution slots; time progresses from left to right. Blue and red blocks represent systems 1 and 2 executing, respectively. Green blocks represent corresponding computational tasks, with contention depicted. Top diagram depicts the non-intelligent (legacy) system; middle diagram depicts effects of poorly chosen computational tasks periods, with significant contention; bottom diagram depicts correct choices, without contention.

exponentially as the number of concurrent systems grows (Figure 1). Per system, we can see that average execution time increases (Table 1).

The negative impact of poor retrofitting decisions has several impacts on practitioners. It greatly complicates the design effort, adding to engineers’ cognitive load. It constitutes several additional

variables that constrain the optimization problem; critically, it poses a discovery and estimation challenge for both known and unknown “unknowns”. Whilst our focus in this paper was on the real-time performance aspects of cyber-physical systems, that is merely illustrative, the broader challenge encompasses not only this but also includes throughput, communication, monitoring, security, and safety.

We identify a few directions that research must pursue to solving these challenges.

- **Estimation methodology.** With so many unknowns that affect the problem and the infamous lack of reliable legacy design documentation, a comprehensive methodology for estimating required parameters is required. Practitioners cannot rely on *ad hoc* approaches; know-how from the disciplines of diagnostics and reverse engineering must be formalized within the context of industrial informatics.
- **Interdisciplinary pollination.** Whilst the need for an optimization approach has long been clear in traditional industrial engineering, its need within industrial informatics has not been so clear. We need to make known more optimization methods available to practitioners of industrial informatics (particularly those engaged with legacy systems)—at the very least, broad adoption of linear programming methods in industrial informatics.
- **Theoretical framework.** The previous directions must be unified within a comprehensive, theoretical framework of modernization on retrofitting intelligence within industrial informatics. Whilst some work has begun this process (Zakoldaev et al., 2019), we are far from a comprehensive theory that informs the discipline. Algorithms, heuristics, best practices, and formalisms must comprise the theoretical framework; this paper offers a small contribution in that direction.
- **Associated tooling.** All these developments mean little if there is no tool support to implement them in practice. In parallel

with establishing a theoretical framework, we must develop hardware and software tools that aid in retrofitting legacy systems if the next industrial revolution is to be fulfilled.

## Data availability statement

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found at: <https://github.com/paulo-chula/Industrial-cyber-physical-simulation>.

## Author contributions

PG: conceptualization, methodology, and writing—original draft. WS: conceptualization, methodology, writing—original draft, and writing—review and editing.

## Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors, and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

## References

- Baruah, S. K., and Goossens, J. (2003). Rate-monotonic scheduling on uniform multiprocessors. *IEEE Trans. Comput.* 52, 966–971. doi:10.1109/tc.2003.1214344
- Ben-Guang, R., Fang-Yu, H., Kraslawski, A., and Nyström, L. (2000). Study on the methodology for retrofitting chemical processes. *Chem. Eng. Technol.* 23, 479–484. doi:10.1002/1521-4125(200006)23:6<479::aid-ccat479>3.0.co;2-5
- Bi, Z. (2017). Embracing internet of things (iot) and big data for industrial informatics. *Enterp. Inf. Syst.* 11, 949–951. doi:10.1080/17517575.2016.1258734
- Blagodurov, S., Zhuravlev, S., and Fedorova, A. (2010). Contention-aware scheduling on multicore systems. *ACM Trans. Comput. Syst. (TOCS)* 28, 1–45. doi:10.1145/1880018.1880019
- Buse, D. P., and Wu, Q. H. (2004). Mobile agents for remote control of distributed systems. *IEEE Trans. Industrial Electron.* 51, 1142–1149. doi:10.1109/tie.2004.837881
- Butun, I., Almgren, M., Gulisano, V., and Papatrifiantafilou, M. (2020). *Industrial IoT*. Springer.
- Carley, T. W., Ba, M. A., Barua, R., and Stewart, D. B. (2003). “Contention-free periodic message scheduler medium access control in wireless sensor/actuator networks,” in RTSS 2003. 24th IEEE Real-Time Systems Symposium, 2003, Cancun, Mexico, 05-05 December 2003 (IEEE), 298–307.
- Fryer, J., and Garcia, P. (2020). “Towards a programming paradigm for reconfigurable computing: asynchronous graph programming,” in 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Vienna, Austria, 08-11 September 2020 (IEEE), 1721–1728.
- Gao, Z., Cecati, C., and Ding, S. X. (2015). A survey of fault diagnosis and fault-tolerant techniques—part i: fault diagnosis with model-based and signal-based approaches. *IEEE Trans. industrial Electron.* 62, 3757–3767. doi:10.1109/tie.2015.2417501
- Gaudio, M. T., Coppola, G., Zangari, L., Curcio, S., Greco, S., and Chakraborty, S. (2021). Artificial intelligence-based optimization of industrial membrane processes. *Earth Syst. Environ.* 5, 385–398. doi:10.1007/s41748-021-00220-x
- Ghobakhloo, M. (2020). Industry 4.0, digitization, and opportunities for sustainability. *J. Clean. Prod.* 252, 119869. doi:10.1016/j.jclepro.2019.119869
- Gomes, T., Pinto, S., Garcia, P., and Tavares, A. (2015). “Rt-shadows: real-time system hardware for agnostic and deterministic oses within softcore,” in 2015 IEEE 20th Conference on Emerging Technologies and Factory Automation (ETFA), Luxembourg, Luxembourg, 08-11 September 2015 (IEEE), 1–4.
- Guerreiro, B. V., Lins, R. G., Sun, J., and Schmitt, R. (2018). “Definition of smart retrofitting: first steps for a company to deploy aspects of industry 4.0,” in *Advances in manufacturing* (Springer), 161–170.
- Hills, T., Leeson, D., Florin, N., and Fennell, P. (2016). Carbon capture in the cement industry: technologies, progress, and retrofitting. *Environ. Sci. Technol.* 50, 368–377. doi:10.1021/acs.est.5b03508
- Johnson, F. R., Stoica, R., Ailamaki, A., and Mowry, T. C. (2010). Decoupling contention management from scheduling. *ACM SIGARCH Comput. Archit. News* 38, 117–128. doi:10.1145/1735970.1736035
- Lasi, H., Fetteke, P., Kemper, H. G., Feld, T., and Hoffmann, M. (2014). Industry 4.0. *Bus. Inf. Syst. Eng.* 6, 239–242. doi:10.1007/s12599-014-0334-4
- Lee, J., Bagheri, B., and Kao, H. A. (2014). “Recent advances and trends of cyber-physical systems and big data analytics in industrial informatics,” in *International*

*proceeding of int conference on industrial informatics (INDIN)* (Porto Alegre, Brazil: Citeseer), 1–6.

Lins, R. G., Guerreiro, B., Schmitt, R., Sun, J., Corazzim, M., and Silva, F. R. (2017). “A novel methodology for retrofitting cnc machines based on the context of industry 4.0,” in 2017 IEEE international systems engineering symposium (ISSE), Vienna, Austria, 11–13 October 2017 (IEEE), 1–6.

Lins, T., and Oliveira, R. A. R. (2020). Cyber-physical production systems retrofitting in context of industry 4.0. *Comput. industrial Eng.* 139, 106193. doi:10.1016/j.cie.2019.106193

Parri, J., Patara, F., Sampietro, S., and Vicario, E. (2021). A framework for model-driven engineering of resilient software-controlled systems. *Computing* 103, 589–612. doi:10.1007/s00607-020-00841-6

Sinnen, O., and Sousa, L. A. (2005). Communication contention in task scheduling. *IEEE Trans. parallel distributed Syst.* 16, 503–515. doi:10.1109/tpds.2005.64

Tavares, A., Carvalho, A., Rodrigues, P., Garcia, P., Gomes, T., Cabral, J., et al. (2012). “A customizable and arinc 653 quasi-compliant hypervisor,” in 2012 IEEE International Conference on Industrial Technology, Athens, Greece, 19–21 March 2012 (IEEE).

Tran, T. A., Ruppert, T., Eigner, G., and Abonyi, J. (2022). Retrofitting-based development of brownfield industry 4.0 and industry 5.0 solutions. *IEEE Access* 10, 64348–64374. doi:10.1109/access.2022.3182491

Zakoldaev, D., Shukalov, A., and Zharinov, I. (2019). From industry 3.0 to industry 4.0: production modernization and creation of innovative digital companies. *IOP Conf. Ser. Mater. Sci. Eng.* 560, 012206. doi:10.1088/1757-899x/560/1/012206

Zhou, X., and Peng, T. (2020). Application of multi-sensor fuzzy information fusion algorithm in industrial safety monitoring system. *Saf. Sci.* 122, 104531. doi:10.1016/j.ssci.2019.104531