A Comprehensive Survey of Industry Practice in Real-time Systems

Benny Akesson
ESI (TNO) & University of Amsterdam
the Netherlands
benny.akesson@tno.nl

Mitra Nasri
Electronic Systems Group
Eindhoven University of Technology
Eindhoven, the Netherlands
m.nasri@tue.nl

Geoffrey Nelissen
Dept. of Mathematics and Computer Science
Eindhoven University of Technology
Eindhoven, the Netherlands
g.r.r.j.p.nelissen@tue.nl

Sebastian Altmeyer
Dept. of Computer Science
University of Augsburg
Augsburg, Germany
sebastian.altmeyer@informatik.uni-augsburg.de

Robert I. Davis
Real-Time Systems Research Group
University of York
York, United Kingdom
rob.davis@york.ac.uk

***Abstract*—This paper presents results and observations from a survey of 120 industry practitioners in the field of real-time embedded systems. The survey provides insights into the characteristics of the systems being developed today and identifies important trends for the future. It extends the results from the survey data to the broader population that it is representative of, and discusses significant differences between application domains. The survey aims to inform both academics and practitioners, helping to avoid divergence between industry practice and fundamental academic research.**

***Index Terms*—Real-time systems, Survey, Industry Practice.**

EXTENDED VERSION

This paper builds upon and extends the RTSS 2020 paper *"An Empirical Survey-based Study into Industry Practice in Real-time Systems"* [1] by adding:

(i) A discussion of potential threats to survey validity and the steps taken to mitigate them, in Section III.
(ii) An analysis (using standard statistical tools) and discussion of the results in the context of five objectives of the survey, in Section V.

I. INTRODUCTION

The real-time embedded systems field covers a broad range of systems from simple control loops on micro-controllers to complex interconnected distributed systems. These systems span many different application domains, including avionics, automotive, consumer electronics, industrial automation, and medical systems, each with its own requirements, standards, and practices. This diversity makes industrial real-time systems and their associated design methods difficult to characterize.

Some fields, such as software engineering [2]–[4] and systems engineering [5], have a history of systematically researching industry practice using surveys, interviews, and literature reviews [6]. This provides a view of the perceived relevance, benefits, and drawbacks of different technologies and methods; identifies trends and opportunities for future research; and tracks the adoption of existing research results. By contrast, there is no tradition of empirical studies into industry practice in the real-time systems field. This omission contributes to a gap, and potentially a divergence, between industry practice and academic research. This paper addresses that gap via an empirical survey-based study into industry practice. The five objectives of the study were to:

O1 Establish whether timing predictability is of concern to the real-time embedded systems industry,
O2 Identify relevant industrial problem contexts, including hardware platforms, middleware, and software,
O3 Determine which methods and tools are used to achieve timing predictability,
O4 Establish which techniques and tools are used to satisfy real-time requirements,
O5 Determine trends for future real-time systems.

A survey targeting industry practitioners in the area of real-time embedded systems was developed and distributed. The survey comprised 32 questions related to the five objectives. Based on the survey data, we formulated a number of propositions about the characteristics of real-time embedded systems and on current practice in industry.

The three main contributions of this survey are:

1) Insights into the characteristics of real-time systems based on responses from 120 industry practitioners from a variety of organizations, countries, and application domains.
2) Discovery of statistically significant differences between the three largest application domains: avionics, automotive, and consumer electronics.
3) Extension of the results from the survey data to the broader population that it is representative of, via the use of standard statistical tools.

The remainder of the paper is organized as follows: Section II outlines related work. Section III describes the methodology used, threats to validity including steps to mitigate them, and the design of the survey. The survey questions and results are elaborated in Section IV, along with key observations and a discussion of statistically significant differences between domains. In Section V, we revisit the objectives, providing a number of propositions based on an extension of the sample results using statistical interference. Section VI concludes.

II. RELATED WORK

Research methods used for understanding industry practice can be broadly divided into three categories:

1) *Survey-based research* targeting industry practitioners in one or more application domains [7]–[14]. Surveys have the advantage that they can often reach more than 100 practitioners, but the drawback that they only invest 10-15 minutes answering 20-30 predetermined questions.
2) *Interviews* with industry practitioners, either open or based on a framework of questions, with the answers subsequently analyzed [15,16]. This approach has the disadvantage that it typically reaches far fewer practitioners, as interviews require more effort to conduct and analyze; however, it benefits from a more dynamic and interactive structure and hence richer responses [5].
3) *Literature surveys reviewing case studies* with the goal of categorizing industry experiences [6].

Some works combine both survey-based research and interviews, exploiting their complementary nature to improve overall quality [9,12]. There are also replication studies investigating how results generalize to other populations [2,13].

In contrast to the fields of software and systems engineering, there has been little if any research undertaken into industry practice in the real-time systems field. Instead, the academic community tends to look inwards, surveying and classifying its own work rather than studying industry practice, contexts, and needs. Examples of well-known literature review surveys include those on uniprocessor scheduling [17,18], multiprocessor scheduling [19], limited preemptive scheduling [20], mixed-criticality scheduling [21], resource allocation and mapping [22], timing analysis [23], and multi-core timing analysis [24]. Recent literature surveys [24]–[26] and other initiatives [27,28] take this one step further and include diagrams illustrating how the number of publications on different research topics has varied over time. This allows hot-topic areas to be identified. While these works may be useful to identify trends in academic real-time systems research, these trends may not be reflected in industry. In conclusion, there is no existing work that systematically surveys industry practice in the real-time systems field. The aim of this paper is to address that omission and to help close the gap between industry practice and academic research.

III. METHODOLOGY

The study described in this paper has five objectives **O1** to **O5** (listed in Section I) that focus on industry practice To meet these objectives, we chose, as the research method, a *survey* asking industry practitioners a set of predetermined questions. As noted in Section II, we found no existing surveys in the relevant area and thus could not reuse any design and assessment of validity and reliability. It was therefore necessary to develop a new survey instrument. In designing the survey, we first considered appropriate validity criteria and their relevant threats (see Section III-A), before making design choices to mitigate those threats (see Section III-B). Note, in the description below, we follow the structure, classifications, and terminology proposed by Kitchenham et al. [4] for survey-based research.

TABLE I
THREATS TO VALIDITY

Threats to Construct Validity

Threat 1	*Improper measurement instrument and/or process*: not measuring the characteristics of interest in the right way
Threat 2	*Improper measured attributes*: not selecting the right attributes to represent the characteristics of interest.

Threats to Internal Validity

Threat 3	*Self-exclusion bias*: participants who are not interested in, or are not allowed, to answer the questionnaire do not fill it in.
Threat 4	*Inclusion of "foreign units"*: questions are answered by respondents who do not have the necessary expertise.
Threat 5	*Selecting duplicate units*: two different teammates who work on the same system answer a question differently.
Threat 6	*Relying on personal experience*: respondents with different levels of experience may answer questions differently.
Threat 7	*Personal bias*: respondents who fill in the survey may have a certain bias with respect to the questions.

Threats to External Validity

Threat 8	*Sampling bias*: the probability that different members of the intended population from part of the sample is not uniform.
Threat 9	*Geographical sampling bias*: geographical location could impact the chance of receiving an invitation.

Threats to Conclusion Validity

Threat 10	Low statistical power or improper use of statistical tests and analyses.

A. Validity Criteria and Threats

Four categories of validity can be identified [3]:

(i) *Construct Validity* refers to the degree to which a question measures what it claims to measure.
(ii) *Internal Validity* reflects whether all causal relations are studied or if unknown factors could affect the results. The main threats to internal validity come from coverage issues [7], as detailed in Table I.
(iii) *External Validity* is the extent to which the research results can be generalized to the world at large [4].
(iv) *Conclusion Validity* is concerned with the ability to draw correct conclusions from the study methods.

B. Survey Design, Instrumentation, and Process

Survey design: The survey was designed as a *cross-sectional study*, i.e. a snapshot taken over a particular period of time, December 2019, to April 2020 in this case. We used a *self-administered questionnaire* on SurveyMonkey [29] that could be answered without the need for intervention, and without providing respondent or company identification. Further, we did not automatically collect any data relating to respondents identities (IP addresses etc.), thus preserving anonymity. The aim here was to reduce the risk of self-exclusion (Threat 3) by enabling those who work on confidential projects to still answer the survey questions. As an additional guarantee of anonymity, we promised respondents to only release summarized and aggregated results [30].

In designing the survey, we were cognizant of the trade-off between having a more comprehensive set of questions and increasing the time needed to complete the survey, which could reduce the completion rate. We began with more questions, but converged on approximately 30, inline with recommendations for scientific surveys, to avoid problems of abandonment (Threat 3). The questions were designed to capture information pertinent to the five objectives of the survey.

Survey questions: We focused on *closed questions*, where respondents are asked to select one or more answers from a list of predefined options. Closed questions are typically faster to answer and easier to analyze than open ones, thus they reduce the likelihood that a participant abandons the survey before reaching the end (to avoid Threat 3). The drawback of closed questions is that they limit the range of possible responses.

Phrasings of the questions, the predefined options, and the scales used to code responses can all impact construct validity. To mitigate Threat 1, the questions were carefully formulated to be neutrally worded, as precise as possible, and avoid unnecessary jargon. Key terms were explained where necessary, helping to ensure that the questions and the list of predefined options were as unambiguous and easy to comprehend as possible. We did not, however, manage to completely eliminate ambiguity caused by jargon. For example, the term "distributed system" used in Questions 10 and 11 may have been interpreted in two different ways. (This is discussed further in Section IV in the observations on Question 11).

With closed questions, care was taken to ensure that the predefined options were unbiased, mutually exclusive, and as exhaustive as possible (to avoid Threat 1). We included "Other", where appropriate, to give respondents the opportunity to go beyond the predefined options when necessary. Where appropriate, we also allowed multiple options to be selected to prevent arbitrary choices between equally valid answers. This was particularly important in the context of real-time systems comprising different sub-systems to which different answers could apply. Both of these techniques are listed as best-practices [4]. The category "I do not know" was added to a number of questions where we did not expect all respondents to be able to give an answer, despite belonging to the target population. This is common practice, despite there being some disagreement about it in the social science community [4]. The "I do not know" category has the benefit of making the lack of knowledge explicit, as distinct from skipped questions and arbitrary answers.

One of the main challenges to construct validity is striking a balance between usefulness and interpretability for the data gathered. Specifically, we had a choice of asking respondents to consider either one or several real-time systems that were being developed in their organization. We chose the former, since although this approach gathers data about fewer systems, it enables conclusions to be drawn about individual systems, and answers to different questions to be related to one another.

Instructions for participants: The welcome page of the survey was written in a neutral tone, explained the purpose of the survey, defined the target population, and suggested that it would take 10-15 minutes to answer the 32 questions. It also explained that the survey was anonymous and that the output of the survey would be an academic paper. The incentive for the respondents was the opportunity to shape future research in the area of real-time systems and align it with industry practice and needs.

Survey validation: To mitigate Threat 2, a draft of the survey was validated by a test group comprising 13 domain experts with extensive industry experience. Their independent (and concurrent) feedback on both questions and possible answers was used to improve the survey and to ensure that it was fit for purpose.

Sampling method: Since there is no list that identifies all industry practitioners who work on real-time embedded systems, it was not possible to perform a *random sampling* to invite the participants. Further, the target population is highly specific and has limited availability, which prevents the use of probabilistic sampling methods. Instead, we used a combination of *convenience sampling* and *snowball sampling*. Convenience sampling means that we reached out to the target population via the authors' combined networks using emails and personal messages on LinkedIn [31]. We sent them a personalized invitation, written in a neutral tone, followed by a reminder a few weeks later, as suggested in [4].

To increase the reach of the survey beyond the authors' networks, we applied snowball sampling in two different ways. First, by encouraging those who we invited to take the survey to forward it to other practitioners. However, we instructed them to only forward the invitation to people working on different real-time systems to avoid Threat 4 and Threat 5. Second, we used snowball sampling to mitigate a geographical bias towards contacts from the region where most of the authors' networks reside (to avoid Threat 8 and Threat 9). We asked 20 academics, primarily based in other regions to forward the invitation to members of the target population in their networks. In convenience sampling, we selected known industry contacts with substantial real-time systems experience, who we expected would be able to provide concrete answers to the questions, representative of the systems in their company's portfolio. We anticipated that these contacts would understand the utility of the survey, and thus be interested in diligently completing it, and in seeing the results. We were careful to avoid selecting contacts working in the same departments, and limited selections within in any one company. Overall, the authors directly invited 114 industry contacts, which, including snowball sampling, resulted in 90 respondents starting the survey. Invitations via the 20 academic contacts resulted in a further 30 respondents starting the survey. Of the 120 respondents starting the survey, 97 completed it. Due to snowball sampling, we do not know the exact response rate, i.e. the total number completing the survey divided by the total number invited to take it.

C. Further Discussions on Threats to Validity

In this subsection, we focus on threats to internal, external, and conclusion validity (construct validity having been covered in the previous subsection).

Internal validity: To reduce the effects of personal bias (Threat 7), we did not ask questions where the answers depend on personal opinion. We also formulated our propositions in Section V based on quantitative measures, i.e, statistical analysis, rather than qualitative ones that could be influenced by personal opinion. To reduce the risk of including foreign units (Threat 4), we did not advertise the survey via public channels. Furthermore, the welcome page explicitly asked only those who consider themselves part of the real-time embedded systems industry to complete the survey. (Threat 3). To reduce the risk of reliance of personal experience (Threat 6), we only invited (via convenience sampling) participants who were considered to have competent to answer the survey questions; however, those subsequently invited via snowball sampling may not necessarily have fulfilled that criteria.

External validity. The survey suffers from sampling bias as a result of practical limitations in finding industry practitioners through the author's networks and the networks of their close academic contacts. This impacts generalization and statistical inference, since they are only valid for a population that the sample data is representative of. This population is undoubtedly not "the real-time embedded systems industry as a whole" but rather some portion of it that can be described as follows:

> ***Effective population**: industry practitioners actively developing real-time embedded systems who have first- or second-order links to academic real-time systems researchers.*

As evidenced by the responses to Question 30 in Section IV, over 80% of respondents interact with the real-time research community (by reading articles, participating in conferences and research projects, and reviewing papers). Further, the responses to Question 1 indicate that only 15% of respondents are from small companies ($<$ 100 employees). Similarly, there is a large variation in the number of respondents per application domain (see Question 4). Hence, the survey results may be more representative of the automotive, avionics and consumer electronics industries than of healthcare or space.

Even though the effective population does not cover the whole real-time embedded systems industry, it represents the potential first-hand industry clients for the work published by the real-time systems research community, as evidenced by the responses to Questions 28 and 30 (which ask about the number of published papers the respondents read and their type of interactions with the research community).

> *The survey results, observations, propositions, and statistical inferences are therefore useful in the context of understanding the state of practice, needs and trends in companies developing real-time embedded systems, who interact with the real-time research community and are interested in exploiting research results.*

Conclusion validity. To isolate and contain the potential impact of Threat 10, we separate the results that are based purely on observations about the sample data, in Section IV, from the propositions that we infer about the effective population using statistical analyses in Section V. Further, to reduce Threat 10 to the statistical analyses used, we have explained our use of statistical inference in Section V, taking into account issues regarding the misinterpretation of confidence intervals [32]. Finally, we sought to give a completely objective view of the results, rather than identifying any particular results as surprises, since such surprises would only be relative to the authors prior beliefs and perceptions.

IV. RESULTS

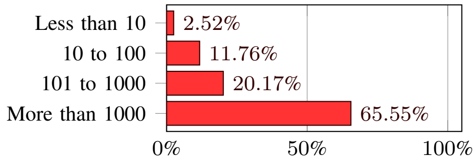
This section lists all of the survey questions, in the order in which they appeared, along with graphs of the results, and our observations. The survey was divided into a number of topics, which are separated by horizontal lines in the text below.

Where results are given as percentages, unless otherwise stated, these correspond to the proportion of respondents who selected that specific option out of all of the respondents who answered that particular question. The graphs presenting the results are color-coded. Red bars are used for questions with distinct alternatives, and hence the total sums to 100%. Blue bars are used where respondents were asked to "select all options that apply", and hence the percentages sum to more than 100%. Multi-colored bars (e.g. Question 6) indicate the percentage of respondents who selected the corresponding scores or rankings. Where the answers have an ordering (e.g. Question 1) then the results are presented in that order, otherwise they have been re-ordered with the most popular answer first for ease of reference, nevertheless, "I do not know" and "Other" are always placed last.

Our observations include a commentary on the results, and a more in-depth look at the data. In some cases, we comment on the results for sub-groups that have been identified via the answers to Question 4 (Avionics, Automotive, and Consumer Electronics) and Question 5 (safety-critical components, and no safety-critical components). We *only* comment on the difference in results between these sub-groups where these differences are statistically significant at the $p < 0.05$ level[1]. Finally, the number X of respondents answering each question is given as **(n=X)** at the right hand side of each question box.

Demographics: This part of the survey asked questions about the respondent's organization and professional experience.

> *Question 1:* How many employees does your organization have? **(n=120)**

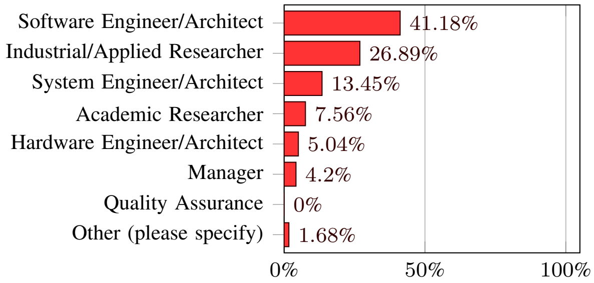


---

[1] According to the analysis provided by the SurveyMonkey toolset, see https://help.surveymonkey.com/articles/en_US/kb/Significant-Differences.

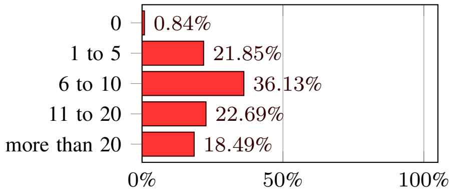
Observations: Approximately two thirds of the respondents were from large companies (> 1000 employees), with around one third from small and medium sized enterprises (SMEs).

Question 2: Which position best describes your current role in your organization? **(n=120)**



Observations: Approximately 60% of the respondents were directly involved in system development (software, system, or hardware), while approximately 27% were involved in industrial research. Note that the category "Academic Researcher" includes staff on secondment to industry, and staff who recently moved from industry to academia.

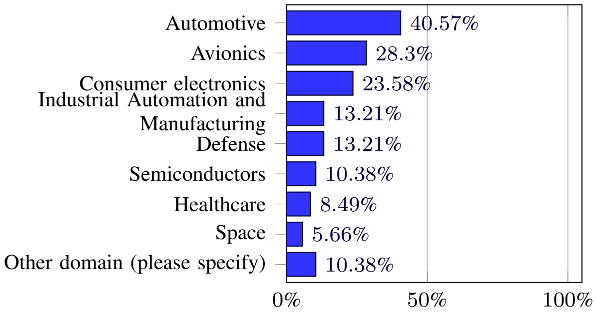
Question 3: How many years industrial experience do you have? **(n=120)**



Observations: The majority of respondents had many years of industrial experience, with 41% having more than 10 years, and only 23% having five years or less.

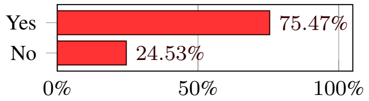
System Context: This part of the survey asked questions about hardware, software and the execution of the system. Respondents were asked to think about a particular system where they were familiar with these aspects, and to consider the same system for all questions to ensure consistent responses.

Question 4: To what domain(s) does the considered system belong? **(n=107)**



Observations: The survey has broad coverage of the different application domains. Note, that multiple domains could be selected. The largest overlaps were between: Avionics and Defense 9.4%, Automotive and Industrial Automation 6.6%, Automotive and Consumer Electronics 5.7%, Automotive and Avionics 5.7%, and Space and Defense 4.7%. Automotive alone was indicated by 65% of those selecting that domain, similarly, Avionics alone by 60% of those selecting that domain, and Consumer Electronics alone by 56% of those selecting that domain. Of the 11 respondents who indicated "Other domain", five specified "Telecomms." i.e. 4.7%.

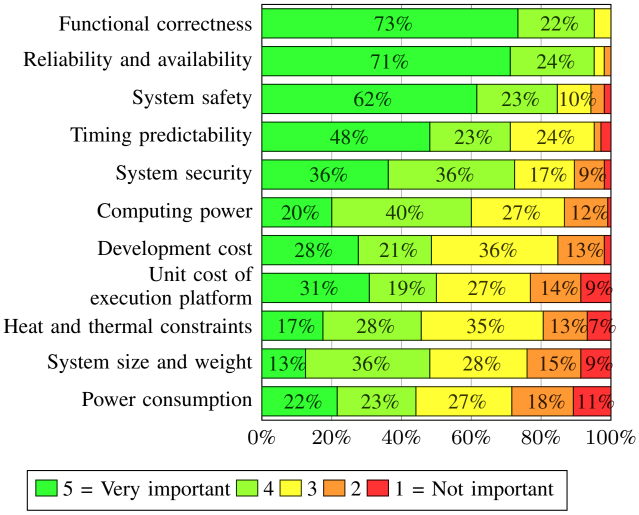
Question 5: Is (parts of) the considered system safety-critical? **(n=107)**



Observations: Even though the response to Question 4 indicates broad domain coverage, a large majority (75%) of the systems considered had some part that was safety critical.

Of those respondents who selected Avionics in Question 4, 100% answered "Yes" to this question, compared to 91% of those who selected Automotive, and just 52% of those who selected Consumer Electronics.

Question 6: Give a score to the importance of different system aspects for the considered system. **(n=107)**

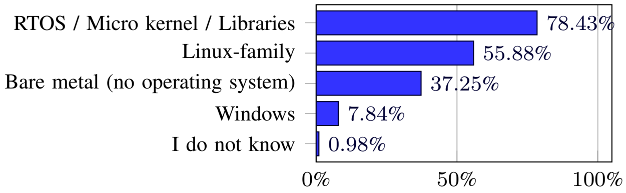


Observations: Timing predictability, although viewed as less important than functional correctness, reliability and safety, was seen as more important than security, computing power, cost, and thermal considerations. (This is perhaps unsurprising, since the survey targeted those working on real-time systems).

Of the respondents who selected Avionics in Question 4, 87% thought that timing predictability was very important, compared to 48% of those who selected Automotive, and just 26% of those who selected Consumer Electronics. In contrast, unit cost of the execution platform was rated as very important by 45% of those respondents who selected Automotive in Question 4, 32% of those who selected Consumer Electronics, and just 7% of those who selected Avionics.

Hardware Platform: This part of the survey asked questions about the hardware and software configurations of the considered system. Here, most questions allowed the selection of multiple options to capture the characteristics of complex systems with many components. Respondents were asked to select all options that apply to the system they were considering.

Question 7: What Operating Systems are running on the considered system? **(n=103)**

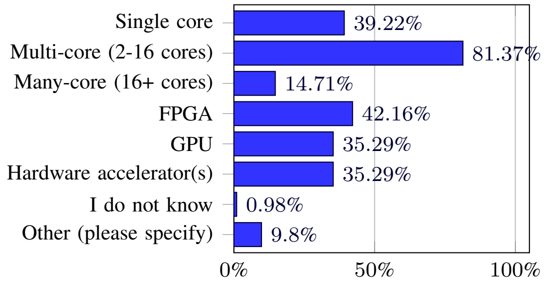


Observations: While 78% of respondents indicated that some parts of their system use an RTOS or Micro kernel, a significant minority (37%) had parts that use no Operating System (OS) at all. RTOS alone was selected by 22.5%, Linux alone by 7.8%, and Bare metal alone by 4.9%. None of the respondents used Windows alone. There were many systems that used more than one OS (62.7%). The largest overlaps were between RTOS and Linux (42.2%), Bare metal and RTOS (28.4%), and Bare metal and Linux (17.6%). The combination of Bare metal, RTOS, and Linux was used by 14.7% of respondents.

Of the respondents who indicated in Question 5 that their system contained some safety-critical components, 87% used an RTOS. This figure reduced to 50% of those who indicated no safety-critical components. By contrast, the corresponding figures for the use of Windows were 3% and 25%, respectively.

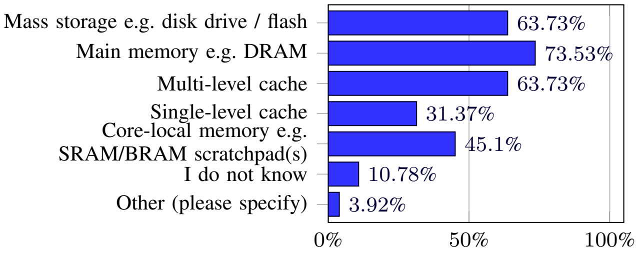
As an optional addition to this question, respondents were asked to name the operating systems that they were using. 32 respondents did so, many citing multiple operating systems. The following lists the operating systems named and the number of times they were cited: Autosar (8), QNX (8), VXWorks (4), OSEK (3), Redhat Linux (3), Free RTOS (2), Linux (2), PikeOS (2), Ubuntu (2), Yocto Linux (2), Arinc-653 (1), DEOS (1), EmbOS (1), Erika (1), Integrity (1), LynxOS (1), RTEMS (1), SafeRTOS (1), ThreadX (1), Windows (1), and Zephyr (1).

Question 8: Select the options that describe the processing hardware of the considered system. **(n=103)**



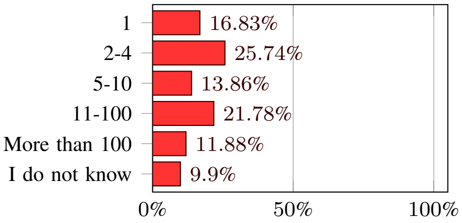
Observations: The majority of systems (81%) include multi-core components, while just under 40% include single core components. Similarly, 35% to 42% of systems include FPGAs, GPUs, and other hardware accelerators. Of the 10 respondents who indicated "Other", three specified DSP (i.e. 2.9%) and two specified System-on-Chip (i.e. 2.0%).

Question 9: Select the options that describe the memory hierarchy of the considered system. **(n=103)**



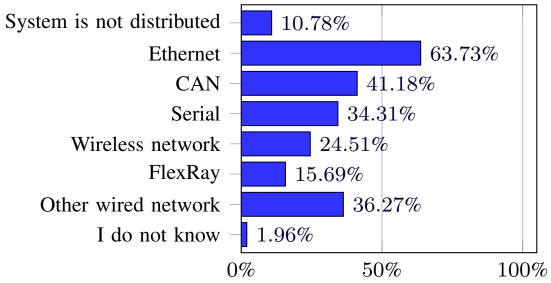
Observations: The majority of systems (over 63%) have elements of a complex memory hierarchy including mass storage devices, DRAM, and multiple levels of cache. Core local memory and single level caches are also prevalent.

Question 10: How many distributed nodes (e.g. ECUs) are there in the considered system? **(n=102)**



Observations: The majority of systems are distributed (73%), with only 17% identified as having a single node (ECU).

Question 11: Which of the following options describe the connectivity within the (distributed) system? **(n=103)**



Observations: Wireless networks were used in around 25% of systems, with Ethernet (64%) and CAN (41%) the most popular forms of wired network. Many systems (48%) used multiple types of wired network, with 34.3% using Ethernet and CAN, 27.5% Ethernet and Serial, 19.6% CAN and Serial, and 14.7% Ethernet, CAN, and Flexray. 9.8% of systems used Ethernet as the only wired network, while less than 3% used CAN, Flexray, or Serial alone. Wireless was used as the only network by 8.8% of respondents, about one third of those using that technology.

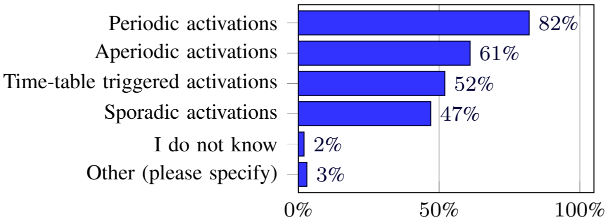
Of the respondents who selected Automotive in Question 4, 74% used CAN, and 34% used Flexray, this reduced to 21% and 3% for those who selected Avionics. Flexray was only used by one respondent (1%) who did not select Automotive.

There was some inconsistency in what was understood by one node or ECU in Question 10 (17%) and what was understood by "not distributed" (11%) in Question 11. This could be because respondents were considering "Nodes" or

"ECUs" in Question 10, and "connectivity" in Question 11. (A single node or ECU may contain multiple connected processing units).

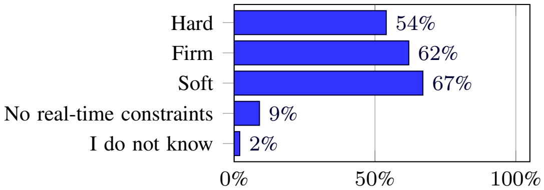
Timing Characteristics: This part of the survey asked questions about the timing characteristics of the considered system.

Question 12: Which of the following sentences are true about task activations in your system? **(n=101)**



Observations: While periodic activation is the most common at 82%, over 60% of systems included aperiodic activations. 22% of responses indicated highly predictable behaviors (utilizing either periodic or time triggered activation) with no sporadic or aperiodic tasks while 4% (and 2%) of respondents indicated to have purely sporadic (aperiodic) activations with no time-triggered or periodic tasks. Interestingly, 74% of respondents indicated at least two, and 25% all four types of activations.

Question 13: Which of the following timing constraints exist(s) in your system? **(n=101)**

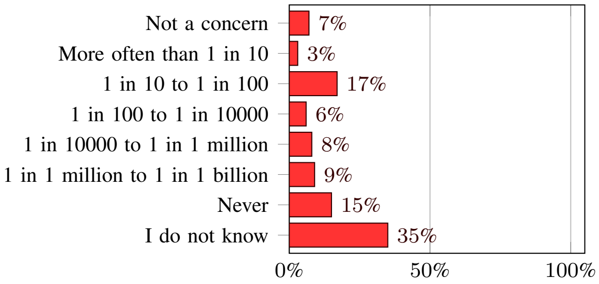


Note, a more detailed explanation of the terms used was provided in the survey[2].

Observations: Given the scope of the survey, it is unsurprising that just under 90% of respondents indicated that their system had some form of timing constraints. Many systems (62%) had a combination of two or more different constraints: Hard and Firm 38%, Hard and Soft 36%, Firm and Soft 42%, and all three 27%. In contrast, far fewer systems had only one type of timing constraint: Hard 5%, Firm 10%, and Soft 15%.

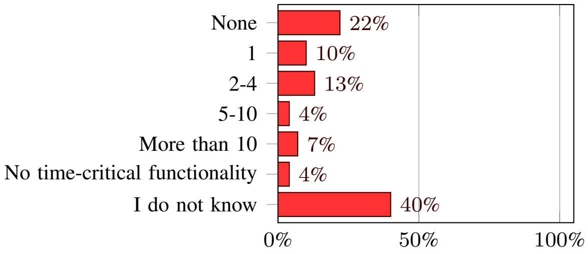
Of the respondents who selected Avionics in Question 4, 79% indicated Hard constraints, compared to 56% of those who selected Automotive, and only 27% of those who selected Consumer Electronics. Of the respondents who indicating in Question 5 that their system contained some safety-critical components, 64% indicated Hard constraints. This reduced to 21% of those who indicated no safety-critical components.

Question 14: For the most time-critical functions in the system, roughly how frequently can the deadline of a function be missed without causing a system failure. **(n=101)**



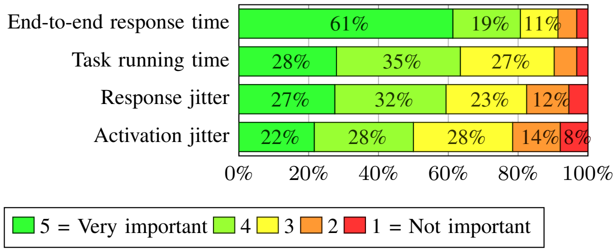
Observations: A substantial number of respondents (35%) were unable to give a specific answer to this question, and answered "I do not know". Only a small proportion (15%) of systems were considered strictly hard real-time, with deadlines that must never be missed. By contrast, 45% of respondents indicated that the most time critical functions in the system could miss some deadlines, and 20% indicated that deadline misses more often than 1 in 100 could be tolerated.

Question 15: What is the largest number of consecutive deadline misses that could be tolerated, assuming that such a blackout does not reoccur for a very long time. **(n=101)**



Observations: The responses to this question follow a similar pattern to those of Question 14, with 34% of respondents indicating that the system can tolerate black-out periods in the range of 1 to more than 10 deadline misses. Here, only about 60% of respondents were able to give a specific answer, with 40% answering "I do not know".

Question 16: What are relevant timing constraints to your system? **(n=99)**



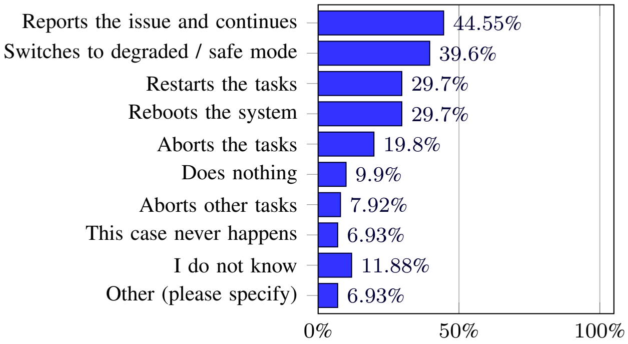
Observations: End-to-end response time was considered the most important timing constraint, with the largest percentage of "very important" scores and the highest average score of 4.3. However, task running time (3.78), response jitter (3.64) and activation jitter (3.42) also need to be considered. 72.7% of respondents rated end-to-end response time highest or equal highest. For task running time, response jitter, and activation

---

[2] Hard implies that violating the timing constraint is considered a failure of the system. Firm implies that violating the timing constraint is highly undesirable. Soft means that occasionally violating the timing constraint is acceptable, but negatively impacts the perceived quality of the system.

jitter, this was the case for 45.5%, 35.4%, and 32.3% of respondents, respectively.

Question 17: How does the considered system react if tasks miss deadlines? **(n=102)**

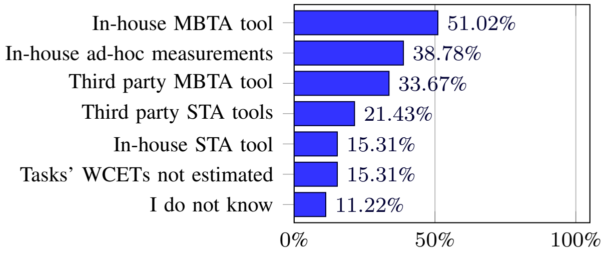


Observations: The most common (45%) reaction to a missed deadline is to report the issue and continue, with 10% of systems also doing nothing. Other systems take actions on a deadline miss, including 30% rebooting, and 30% restarting tasks. Further, although 15% said that a deadline miss may never occur (Question 14) only 7% trust their system enough to state that "This case never happens".

Of the respondents who indicated in Question 5 that their system contained some safety-critical components, 36% indicated "Reboots the system". This reduced to just 8% of those who indicated no safety-critical components. By contrast, the figures for "Does nothing" were 6% and 21%, respectively.

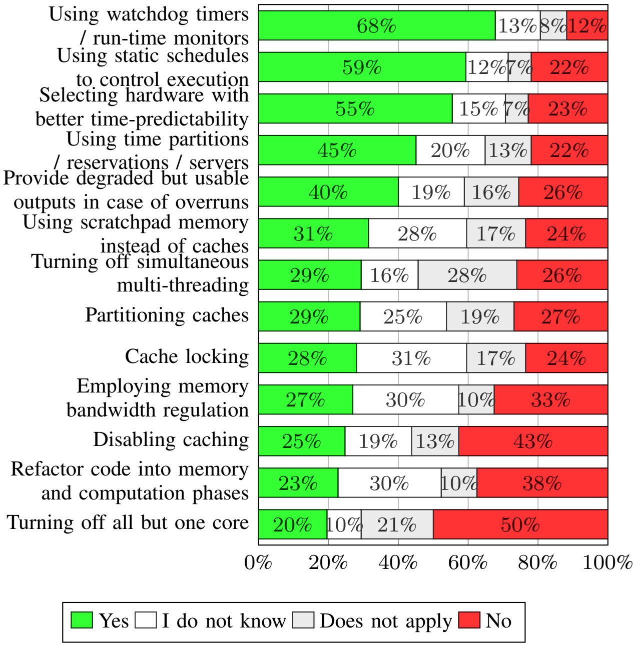
Managing Timing Behavior: This part of the survey asked questions about the methods used to analyze and influence the timing behavior of the system.

Question 18: Which methods are used for Worst-Case Execution Time (WCET) estimation in the considered system? **(n=99)**



Observations: Measurement-Based Timing Analysis (MBTA) tools are used by substantially more respondents than Static Timing Analysis (STA) tools, with more than 50% using in-house MBTA tools compared to 15% for in-house STA tools. This distinction in less stark when it comes to third party solutions with 34% using third party MBTA tools and 21% using third party STA tools. Overall, 67.4% of respondents used some form of MBTA tool, 33.7% used some form of STA tool, and 23.5% used both.

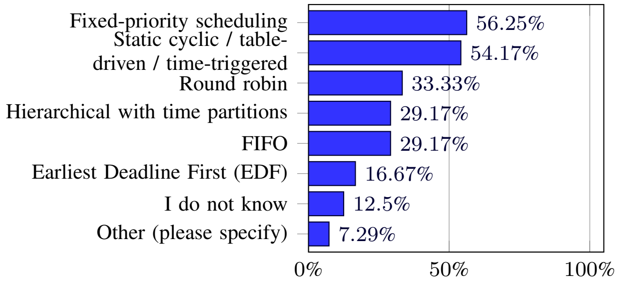
Question 19: What steps are taken to help increase timing predictability? **(n=97)**



Observations: While more than 50% of respondents use watchdog timers, static scheduling, and appropriate hardware selection, it is clear that there is no "silver bullet" to improving timing predictability. Each of the wide range of different techniques is used by at least 20% of respondents, and 46% of respondents answered "Yes" to at least 5 of the techniques listed. Some of the techniques that are least frequently employed are, however, those that have the largest impact on average-case execution times (e.g. disabling caching and turning off all but one core).

There was considerable uncertainty in answering parts of this question, reflected in approximately 30% of respondents answering "I don't know" with respect to the use of scratchpads, cache locking, memory bandwidth regulation, and refactoring code into memory and computation phases.

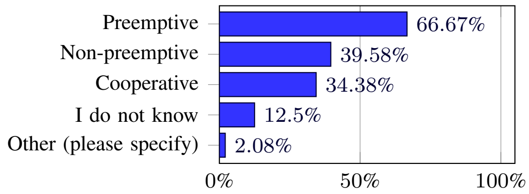
Question 20: Which task scheduling policy/policies are used in the considered system? **(n=97)**



Observations: The most popular scheduling policies were fixed priority and static cyclic / table driven with each used by more than half of the respondents. Round-robin and FIFO, which are not traditionally viewed as real-time policies, were employed in around 30% of systems, while EDF was employed in less than 17% of systems, less than one third as often as fixed priority scheduling.

Of the respondents who selected Automotive in Question 4, 27% used EDF scheduling, compared to just 3% of those who selected Avionics, and 11% of those who selected Consumer Electronics.

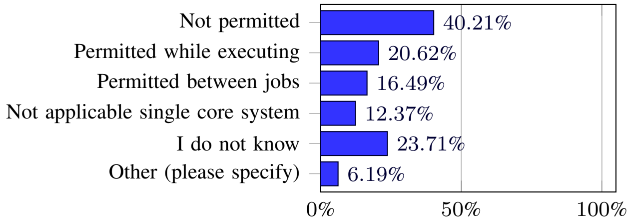
Question 21: Indicate the types of preemption that are supported in the considered system. **(n=97)**



Note, a more detailed explanation of the terms used was provided in the survey[3].

Observations: While preemptive scheduling is the most popular choice, used in two thirds of systems, both non-preemptive and co-operative scheduling are used in more than one third of systems.

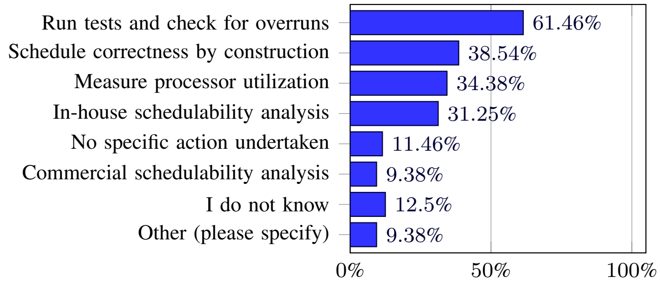
Question 22: Indicate how task migration can take place between different cores in the considered system. **(n=98)**



Observations: Although timing predictability is typically easier to achieve without task migration, the proportion of systems permitting migration (37%) is similar to the proportion that do not permit it (40%).

Of the respondents who selected Avionics in Question 4, only 7% indicated that task migration is permitted while the task is executing. By comparison, this figure was 27% for those who selected Automotive, and 30% for those who selected Consumer Electronics.

Question 23: How do you ensure that the functions in the considered system respect their deadlines? **(n=97)**



Observations: Less than 10% of respondents are using commercial schedulability analysis tools, while more than 30% use in-house solutions. The main off-line approach is schedule correctness by construction, using a static schedule and checking that execution time budgets hold. However, the most common approach overall is to run tests and check for overruns (61%), with a similar proportion to those that use watchdogs timers / run-time monitors (see Question 19).

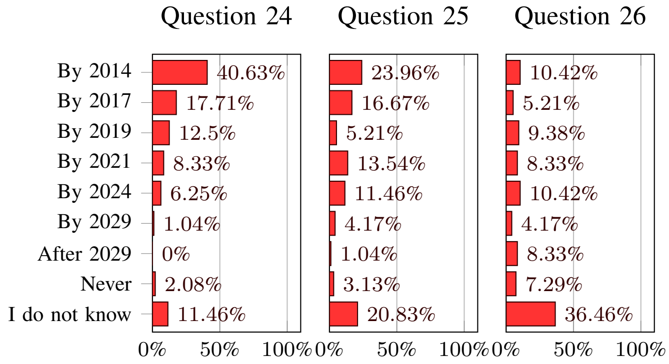
None of the respondents who selected Avionics in Question 4, indicated "no specific action undertaken", compared to 12% of those who selected Automotive, and 16% of those who selected Consumer Electronics. Of the respondents who indicating in Question 5 that their system contained some safety-critical components, 7% answered "no specific action undertaken". The corresponding figure was 29% of those who indicated no safety-critical components.

Timelines for Hardware Adoption: This part of the survey asked questions about timelines for hardware adoption.

Question 24: By which year did or do you expect development projects for real-time embedded systems in your department to begin using multi-core embedded processors (i.e. processors with 2 to 16 cores)? **(n=97)**

Question 25: By which year did or do you expect development projects for real-time embedded systems in your department to begin using heterogeneous multi-cores with different types of CPUs, GPUs, and other accelerators? **(n=97)**

Question 26: By which year did or do you expect development projects for real-time embedded systems in your department to begin using many-core embedded processors (i.e. processors with more than 16 cores)? **(n=97)**

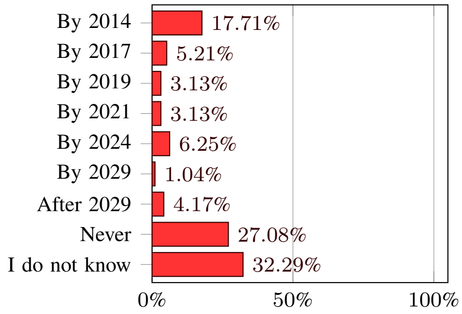


Observations: Multi-core systems (Question 24) are already widely used in current developments, with 80% of respondents indicating their use by 2021, and only 10% answering "I do not know". The uptake of heterogeneous multi-core systems (Question 25) lags behind simpler multi-core systems, but nevertheless just under 60% of respondents indicate their use by 2021, with 20% answering "I do not know". Finally, the uptake of many-core systems (Question 26) is less certain, with 36% of respondents answering "I do not know", 33% indicating take up by 2021, and 48% take up by by 2029.

---

[3] Preemptive implies that task execution can be preempted by other tasks at any time, non-preemptive implies that task execution cannot be preempted by other tasks before completion, and cooperative means that task execution can be preempted by other tasks, but only at predefined preemption points.

Question 27: By which year did or do you expect new development projects for real-time embedded systems in your department to stop using single-core embedded processors (i.e. processors with one core)? **(n=97)**

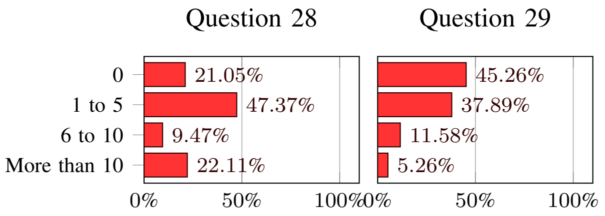


Observations: Although the proportion of respondents expecting to use single-core devices drops in future years, a substantial minority (31%) still expect to use single cores after 2029. Interestingly, this is the case for respondents who indicated each of the Automotive, Avionics, and Consumer Electronics domains in Question 4, with 30%, 34.5%, and 30%, respectively, expecting to use single-cores after 2029.

Familiarity with Real-time Systems Research: This part of the survey asked questions about familiarity with the real-time systems research community and its results.

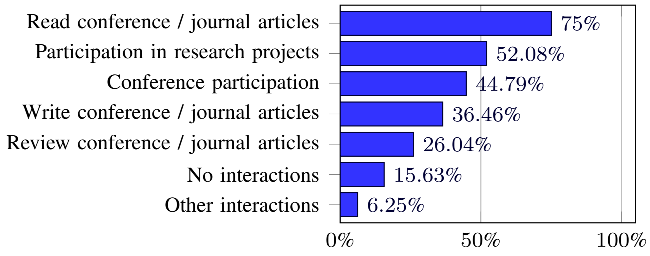
Question 28: How many research publications (e.g. conference or journal papers) in the real-time systems field have you read in the last year? **(n=96)**

Question 29: How many real-time systems research publications (e.g. conference or journal papers) have you published as a (co-)author in the last 5 years? **(n=96)**



Observations: Around 79% of respondents read at least one research publication in the past year, and around 55% contributed to research publications in the past 5 years.

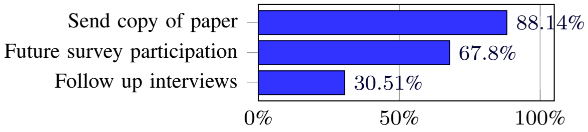
Question 30: How do you interact with the real-time research community? **(n=97)**



Observations: Only 16% of respondents have no interactions with the real-time research community. "Other interaction" (6%) included: research internships, co-supervisions, and interacting with researchers directly.

Follow Up: The final part of the survey asked questions about following up on this survey and general remarks.

Question 31: Indicate the purposes for which we may contact you again, if any. **(n=60)**



Observations: 48 respondents provided their email addresses for subsequent follow up.

Question 32: Enter feedback or remarks **(n=23)**

Observations: 23 respondents provided feedback. The most common comments were complementary remarks about the survey and a desire to see the results.

V. ANALYSIS AND DISCUSSION

In this section, we present the results of using a standard statistical tool, to generalize our findings by estimating parameters (e.g. the proportion using some feature F) for the population from which the sample data was taken. Using *statistical inference*, we derive the *confidence intervals* of our main findings, at a *confidence level* γ of 95%. Confidence intervals provide a useful estimate of population parameters, since their calculation tends to produce intervals that contain the true value of the parameter. There are a number of common misconceptions about confidence intervals [32]. For example, it is not correct to assume that there is a probability of γ (e.g. 95%) that the confidence interval will actually contain the true parameter value. Rather, confidence intervals are such that if the sampling process were repeated a large number of times, then the true value of the population parameter would be expected to fall within the confidence intervals computed for the samples γ (e.g. 95%) of the time [33]. The confidence interval represents a range of values for the population parameter for which the difference between the parameter and the estimate from the sample is not statistically significant at the $(1-\gamma)$ level [34]. Hence, if the true value does not fall within the confidence interval, then it means that a sampling event has occurred that had a probability of $(1-\gamma)$ (e.g. 5%) or less of happening by chance.

Below, we revisit the five objectives set out in Section I. For each objective, we use statistical inference to extend the results of the survey to the *effective population* (see Section III-C). For each objective, we list a set of *propositions*. Each proposition is expressed as a statement (in **bold**) that is derived from generalizations of the survey results that follow. Each generalization gives the data from the sample, followed by a confidence interval for the population parameter (e.g. "C (count) of S (sample size), CI [L% to U%] (confidence interval), use feature F"). The confidence intervals were calculated assuming a 95% confidence level using the StatKey online tool[4]. The confidence intervals were computed excluding "I do not know" responses, since we assumed that such responses to factual questions were genuine attempts by respondents to complete the survey to the best of their knowledge.

Objective O1 Establish whether timing predictability is of concern to the real-time embedded systems industry.

Proposition 1: **Although timing predictability is important, it is only one of many system design aspects.**
97 of 105, CI [87%, 97%], consider timing predictability to be no more important than functional correctness. 96 of 105, CI [86%, 96%], than reliability and availability; 90 of 105, CI [79%, 92%], than system safety; 69 of 105, CI [56%, 75%], than system security; 54 of 105, CI [42%, 61%], than development cost; 53 of 105, CI [40%, 60%], than unit cost; **(Question 6)**

Objective O2 Identify relevant industrial problem contexts, including hardware platforms, middleware, and software.

Proposition 2: **Hardware platforms are complex and distributed.**
91 of 101, CI [84%, 96%], use multi-cores or many-cores. 68 of 101, CI [57%, 77%], have FPGAs and/or GPUs and/or hardware accelerators. 48 of 91, CI [42%, 63%], include mass storage, main memory, and multi-level caches. 65 of 100, CI [56%, 74%], use two or more types of network. 48 of 91, CI [42%, 63%], include 5 or more distributed nodes. **(Questions 8, 9, 10, 11)**

Proposition 3: **Multiple different types of Operating System (OS) are used, often within the same system[5].**
80 of 101, CI [71%, 87%], use an RTOS; 38 of 101, CI [28%, 48%], use bare metal (i.e. no OS); 57 of 101, CI [46%, 67%], use Linux. 60 of 101, CI [49%, 69%], use at least two of: bare metal, RTOS, Linux in the same system. **(Question 7)**

Proposition 4: **Deadlines are not sacrosanct.**
44 of 66, CI [54%, 79%], consider that the most time critical functions in their systems can miss some deadlines. 20 of 66, CI [19%, 40%], can miss deadlines more often than 1 in 100. Further, 24 of 57, CI [29%, 55%], can tolerate two or more consecutive deadline misses. **(Questions 14, 15)**

Proposition 5: **Different types of timing constraints are present in the same system[5].**
62 of 98, CI [54%, 73%], work on systems with a mix of at least two different types of timing constraint (i.e. hard, firm, and soft). **(Question 13)**

Objective O3 Determine which methods and tools are used to achieve timing predictability.

Proposition 6: **Measurement-based timing analysis is more prevalent than static timing analysis, but both are used.**
66 of 87, CI [66%, 85%], use measurement-based timing analysis. 33 of 87, CI [27%, 49%], use static timing analysis, and 23 of 87, CI [17%, 36%], use both. **(Question 18)**

Proposition 7: **Both static and dynamic methods of improving timing predictability are widely used.**
76 of 95, CI [71%, 88%], use at least one static approach (e.g. static schedules, time partitions, turn off multi-threading, partitioned caches, cache locking, disable caching, refactor code into memory and computation phases, turn off all but one core). 74 of 95, CI [69%, 87%], use at least one dynamic mechanism (e.g. watchdog timers, degraded outputs on overrun, memory bandwidth regulation, and run-time monitors). **(Question 19)**

Proposition 8: **Systems often take mitigating actions in the event of timing violations.**
40 of 89, CI [34%, 56%], switch to degraded/safe mode; 44 of 89, CI [39%, 60%], abort or restart tasks; and 30 of 89, CI [23%, 44%], reboot the system to mitigate missed deadlines. **(Question 17)**

Proposition 9: **Some systems use only highly predictable task activation patterns.**
21 of 99, CI [13%, 30%], use exclusively periodic and/or time-triggered forms of task activation. **(Question 12)**

Objective O4 Establish which techniques and tools are used to satisfy real-time requirements.

Proposition 10: **Many different scheduling policies are used in the same system[5], some of which are not "real-time".**
54 of 84, CI [53%, 74%], use Fixed Priority scheduling, 52 of 84, CI [51%, 73%], use static cyclic scheduling, 16 of 84, CI [10%, 28%], use EDF. 32 of 84, CI [28%, 49%], use Round-robin, and 28 of 84, CI [23%, 44%], use FIFO. 56 of 84, CI [57%, 77%], use at least two of the above policies in the same system. **(Question 20)**

Proposition 11: **Many different preemption strategies are used in the same system[5].**
65 of 85, CI [67%, 85%], use preemptive scheduling. 56 of 85, CI [55%, 77%], use cooperative and/or non-preemptive scheduling. 36 of 85, CI [31%, 53%], use both preemptive and cooperative/non-preemptive scheduling in the same system[5]. 28 of 85, CI [23%, 43%], use exclusively preemptive scheduling. 19 of 85, CI [14%, 32%], use exclusively cooperative and non-preemptive scheduling. **(Question 21)**

Proposition 12: **Some systems permit task migration between cores.**
13 of 58, CI [12%, 33%], always permit task migration either between or during jobs. 17 of 58, CI [17%, 42%], do not permit any form of task migration. 30 of 58, CI [39%, 64%], permit migrations for some parts and do not permit task migrations for other parts. **(Question 22)**

Proposition 13: **The most common way to verify timing requirements is to run tests and check for overruns.**
37 of 84, CI [33%, 55%], use static schedules. 37 of 84, CI [33%, 55%], use schedulability analysis tools. 59 of 84, CI [60%, 80%], run tests and check for overruns. **(Q. 23)**

---

[4] http://www.lock5stat.com/StatKey/index.html, "CI for single proportion".

[5] Note most systems are distributed and have multiple nodes.

Objective O5 Determine trends for future real-time systems.

Proposition 14: **Multi-core and complex heterogeneous multi-core processors are being adopted, as are many-cores.** 77 of 86, CI [82%, 96%], expect to use multi-cores by 2021, 57 of 77, CI [63%, 84%], expect to use complex heterogeneous multi-cores by 2024, and 32 of 62, CI [38%, 65%], expect to use many-cores by 2029. **(Q. 24, 25, 26)**

Proposition 15: **Single-cores continue to be used.** 30 of 65, CI [33%, 59%], expect to still be using single-cores after 2029. **(Question 27)**

The results of the survey show that many respondents work for companies that are active in multiple application domains. This real-world complexity prevented a fully stratified analysis, comparing and contrasting the characteristics of different application domains.

VI. CONCLUSIONS

An absence of any systematic studies into industry practice increases the risk that academic research will diverge from areas that are crucial to the development of future industrial systems. This may lead to research that is less relevant, less likely to be adopted, and has a lower potential for impact. While empirical survey-based research is well-established in software and system engineering, there were previously no such studies in the real-time systems field. This paper addresses that omission by presenting the results of a survey, containing 32 questions related to methods, tools, and trends in industrial real-time systems development. The survey was completed by 120 industry practitioners from a variety of different organizations, countries, and application domains.

The survey results show that industry recognizes the importance of timing predictability, but that other design aspects are of equal or greater importance, such as functional correctness and reliability/availability. Hence, it is important for real-time systems research to be cognizant of its impact on these aspects.

Many real-time systems today are distributed systems that use multi-core processors, and have complex memory hierarchies. Further, multiple different operating systems and networking technologies are typically utilized within the same system, as are different types of timing constraints and task activation patterns. Many respondents did not consider timing constraints to be sacrosanct, with even the most time critical functions allowed to miss some deadlines.

There is no silver bullet to manage timing behavior in complex real-time systems. Instead, the survey reveals that a wide range of different tools, techniques, and policies are used for timing analysis, scheduling, and to increase timing predictability. There is no one size fits all solution.

The trends suggest that single-core systems are still widely used today, and are expected to remain relevant for new developments for at least the next ten years. However, more complex (heterogeneous) multi- and many-core systems are already prevalent and their adoption is expected to increase significantly during the 2020s.

Finally, we would like to end by encouraging members of the academic real-time systems research community to read the survey carefully, absorb and interpret the information presented in the context of their specific research topics, and reflect on how they can address the variety and complexity of future industrial real-time systems in their own research. With this purpose in mind, the primary data from this survey has been made available in aggregate form for others to use [30].

ACKNOWLEDGMENTS

The research described in this paper is funded, in part, by the Innovate UK HI-CLASS project and by the Netherlands Organisation for Applied Scientific Research TNO. The authors would like to thank all of the respondents who took the time and effort to complete the survey and also to provide feedback on it. Without them, this paper would not have been possible.

REFERENCES

[1] B. Akesson, M. Nasri, G. Nelissen, S. Altmeyer, and R. I. Davis, “An empirical survey-based study into industry practice in real-time systems,” in *IEEE Real-Time Systems Symposium (RTSS 2020)*, 2020.

[2] A. Höfer and W. F. Tichy, “Status of empirical research in software engineering,” in *Empirical Software Engineering Issues. Critical Assessment and Future Directions*. Springer, 2007, pp. 10–19.

[3] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, and A. Wesslén, *Experimentation in software engineering*. Springer Science & Business Media, 2012.

[4] B. A. Kitchenham and S. L. Pfleeger, “Personal opinion surveys,” in *Guide to advanced empirical software engineering*. Springer, 2008, pp. 63–92.

[5] G. Muller, “Systems engineering research methods,” *Procedia Computer Science*, vol. 16, pp. 1092–1101, 2013.

[6] P. Mohagheghi and V. Dehlen, “Where is the proof? – a review of experiences from applying MDE in industry,” *Lecture Notes in Computer Science*, vol. 5095, pp. 432–443, 2008.

[7] M. Torchiano, F. Tomassetti, F. Ricca, A. Tiso, and G. Reggio, “Relevance, benefits, and problems of software modelling and model driven techniques – a survey in the Italian industry,” *Journal of Systems and Software*, vol. 86, no. 8, pp. 2110–2126, 2013.

[8] G. Liebel, N. Marko, M. Tichy, A. Leitner, and J. Hansson, “Model-based engineering in the embedded systems domain: an industrial survey on the state-of-practice,” *Software & Systems Modeling*, vol. 17, no. 1, pp. 91–113, 2018.

[9] J. Hutchinson, J. Whittle, and M. Rouncefield, “Model-driven engineering practices in industry: Social, organizational and managerial factors that lead to success or failure,” *Science of Computer Programming*, vol. 89, pp. 144–161, 2014.

[10] A. Forward and T. C. Lethbridge, “Problems and opportunities for model-centric versus code-centric software development: a survey of software professionals,” in *Proceedings of the 2008 international workshop on Models in software engineering*. ACM, May 2008, pp. 27–32.

[11] M. Broy, S. Kirstan, H. Krcmar, B. Schätz, and J. Zimmermann, “What is the benefit of a model-based design of embedded software systems in the car industry?” *Emerging Technologies for the Evolution and Maintenance of Software Models*, pp. 343–369, 2012.

[12] J. Whittle, J. Hutchinson, and M. Rouncefield, “The state of practice in model-driven engineering,” *IEEE software*, vol. 31, no. 3, pp. 79–85, 2014.

[13] A. Vetro, W. Bohm, and M. Torchiano, “On the benefits and barriers when adopting software modelling and model driven techniques – an external, differentiated replication,” in *Empirical Software Engineering and Measurement (ESEM), 2015 ACM/IEEE International Symposium on*. IEEE, 2015, pp. 1–4.

[14] F. Hermans, M. Pinzger, and A. Van Deursen, “Domain-specific languages in practice: A user study on the success factors,” in *International Conference on Model Driven Engineering Languages and Systems*. Springer, 2009, pp. 423–437.

[15] J. Whittle, J. Hutchinson, M. Rouncefield, H. Burden, and R. Heldal, "A taxonomy of tool-related issues affecting the adoption of model-driven engineering," *Software & Systems Modeling*, vol. 16, no. 2, pp. 313–331, 2017.

[16] A. Kuhn, G. C. Murphy, and C. A. Thompson, "An exploratory study of forces and frictions affecting large-scale model-driven development," in *International Conference on Model Driven Engineering Languages and Systems*. Springer, 2012, pp. 352–367.

[17] N. C. Audsley, A. Burns, R. I. Davis, K. W. Tindell, and A. J. Wellings, "Fixed priority pre-emptive scheduling: An historical perspective," *Real-Time Syst.*, vol. 8, no. 23, p. 173198, Mar. 1995. [Online]. Available: https://doi.org/10.1007/BF01094342

[18] L. Sha, T. Abdelzaher, K.-E. Årzén, A. Cervin, T. Baker, A. Burns, G. Buttazzo, M. Caccamo, J. Lehoczky, and A. K. Mok, "Real time scheduling theory: A historical perspective," *Real-Time Syst.*, vol. 28, no. 23, p. 101155, Nov. 2004. [Online]. Available: https://doi.org/10.1023/B:TIME.0000045315.61234.1e

[19] R. I. Davis and A. Burns, "A survey of hard real-time scheduling for multiprocessor systems," *ACM Comput. Surv.*, vol. 43, no. 4, Oct. 2011. [Online]. Available: https://doi.org/10.1145/1978802.1978814

[20] G. C. Buttazzo, M. Bertogna, and G. Yao, "Limited preemptive scheduling for real-time systems. a survey," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 3–15, 2013.

[21] A. Burns and R. I. Davis, "A survey of research into mixed criticality systems," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, pp. 1–37, 2017.

[22] A. K. Singh, P. Dziurzanski, H. R. Mendis, and L. S. Indrusiak, "A survey and comparative study of hard and soft real-time dynamic resource allocation strategies for multi-/many-core systems," *ACM Comput. Surv.*, vol. 50, no. 2, Apr. 2017. [Online]. Available: https://doi.org/10.1145/3057267

[23] R. Wilhelm, J. Engblom, A. Ermedahl, N. Holsti, S. Thesing, D. Whalley, G. Bernat, C. Ferdinand, R. Heckmann, T. Mitra *et al.*, "The worst-case execution-time problemoverview of methods and survey of tools," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 7, no. 3, p. 36, 2008.

[24] C. Maiza, H. Rihani, J. M. Rivas, J. Goossens, S. Altmeyer, and R. I. Davis, "A survey of timing verification techniques for multi-core real-time systems," *ACM Computing Surveys (CSUR)*, vol. 52, no. 3, pp. 1–38, 2019.

[25] R. I. Davis and L. Cucu-Grosjean, "A survey of probabilistic schedulability analysis techniques for real-time systems," *LITES: Leibniz Transactions on Embedded Systems*, pp. 1–53, 2019.

[26] ——, "A survey of probabilistic timing analysis techniques for real-time systems," *Leibniz Transactions on Embedded Systems*, vol. 6, no. 1, pp. 03–1, 2019.

[27] [Online]. Available: https://github.com/ebni/classify-rt

[28] [Online]. Available: https://site.ieee.org/tcrts/classification-of-rtss-papers/

[29] [Online]. Available: https://www.surveymonkey.com

[30] B. Akesson, M. Nasri, G. Nelissen, S. Altmeyer, and R. I. Davis, "Data Set Real-Time Systems Survey," Oct. 2020. [Online]. Available: https://uvaauas.figshare.com/articles/dataset/Data_Set_Real-Time_Systems_Survey/13117709

[31] [Online]. Available: https://www.linkedin.com/

[32] S. Greenland, S. J. Senn, K. J. Rothman, J. B. Carlin, C. Poole, S. N. Goodman, and D. G. Altman, "Statistical tests, p values, confidence intervals, and power: A guide to misinterpretations," *European Journal of Epidemiology*, vol. 31, no. 4, pp. 337–350, 2016.

[33] J. Neyman and H. Jeffreys, "Outline of a theory of statistical estimation based on the classical theory of probability," *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, vol. 236, no. 767, pp. 333–380, 1937. [Online]. Available: https://royalsocietypublishing.org/doi/abs/10.1098/rsta.1937.0005

[34] D. R. Cox and D. V. Hinkley, *Theoretical Statistics*. Chapman and Hall, 1979. [Online]. Available: https://www.amazon.co.uk/Theoretical-Statistics-D-R-D-V-Hinkley/dp/0412161605