An Empirical Survey-based Study into Industry Practice in Real-time Systems

Benny Akesson, Mitra Nasri, Geoffrey Nelissen,
Sebastian Altmeyer, and Rob Davis

An initiative of industry, academia and TNO

ESI

Introduction

Real-time embedded systems is a broad field spanning multiple application domains

- Diversity makes systems and design methods **difficult to characterize**

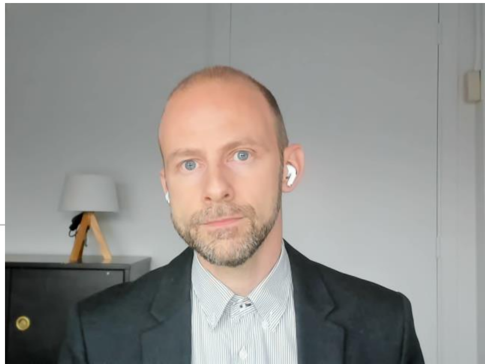
Systematically researching industry practice is common in software engineering

- Provides **views** on needs, technology adoption, trends, and innovation gaps

There is no tradition of empirical studies into industry practice in real-time systems

- Contributes to **gap** and **possible divergence** between industry practice and academic research

This paper addresses this problem through an empirical study



Study Objectives

1. Establish whether timing predictability is of concern to the real-time systems industry
2. Identify relevant industrial problem contexts, including hardware, middleware, and software
3. Determine which methods and tools are used to achieve timing predictability
4. Establish which techniques and tools are used to satisfy real-time requirements
5. Determine trends for future real-time systems

Contributions

A survey targeting industry practitioners was developed and distributed

- 32 questions related to the five objectives

The two main contributions of the survey are:

1. **Insights into characteristics of real-time systems based on responses from 120 practitioners**
2. **Discovery of statistically significant differences between the three largest domains**

Presentation Outline

Introduction

Methodology

Results

Conclusions

Methodology

A survey is chosen as the research method to meet the five research objectives

- A new survey instrument was created as there was **no existing survey** available for this purpose

Survey design

- The survey was created using **SurveyMonkey**
- Survey comprised **32 questions**, which would take some **15 minutes to answer**
- Focus on **closed questions**, which are **faster to answer** and **easier to analyze**
- Survey validated by a **test group** of 13 people from industry, research institutes, and universities
- Survey is **anonymous** and we promised only to release aggregated data

Sampling Method

A combination of convenience sampling and snowball sampling was used

- Members of the target population were invited using **e-mails** and **private LinkedIn messages**
- Invitees were encouraged to **forward invitations** to others working on **different systems**
- 20 academics from **across the world** forwarded the invitation in their industrial networks

The survey was open from December 2019 to April 2020

- 120 participants started the survey and 97 made it to the end

Please read details about methodology and threats to validity before using these results

Presentation Outline

Introduction

Methodology

Results

Conclusions

Introduction to Results

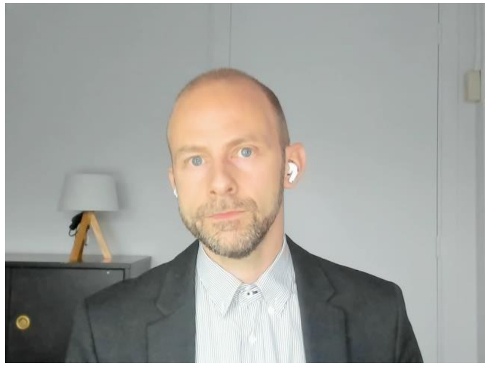
This section will present the results of the survey and highlight relevant observations

We comment on statistically significant differences between domains at $p < 0.05$

- Larger p value identifies more significant differences, but also adds false positives

Color legend

- Red bars are for mutually exclusive answers, so percentages sum up to 100%
- Blue bars are for multiple choice questions, so percentages sum up to >=100%



Demographics

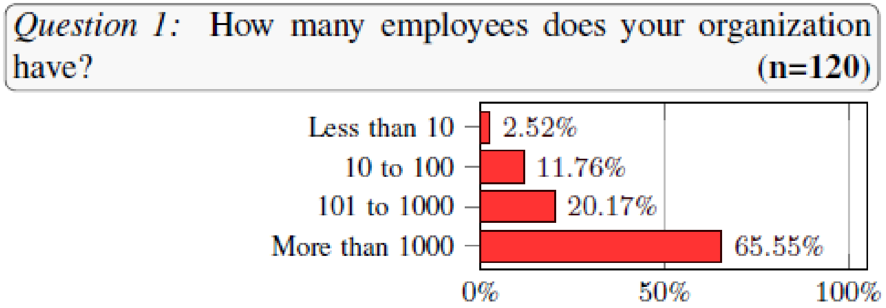
Most respondents (66%) work for large companies

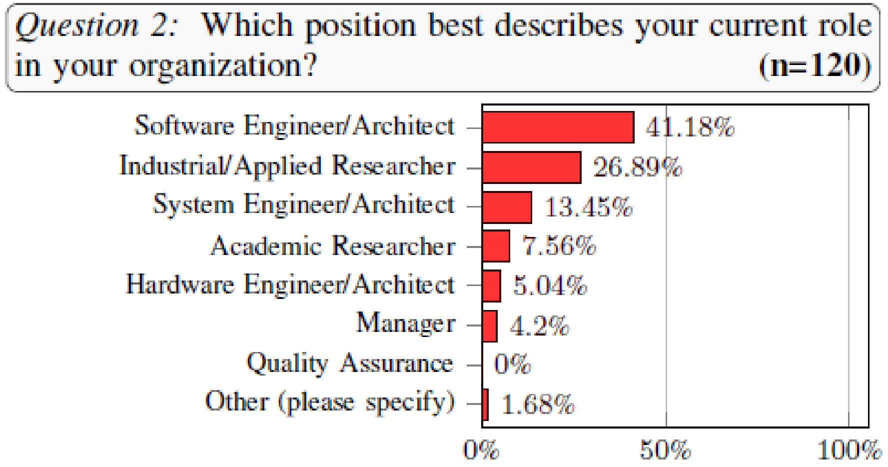
- Remaining 34% for SMEs

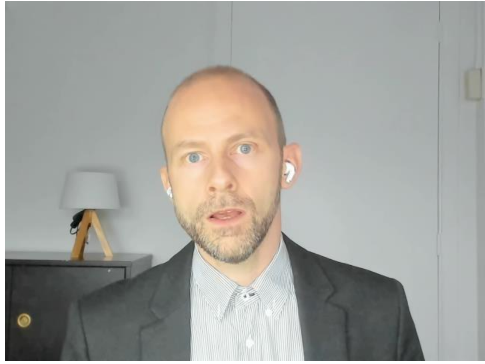
Roles in the organization

- 60% directly involved in system development
- 27% involved in industrial research
- 8% academic research, e.g. seconded staff

The majority of respondents had many years of experience







System Domains

Participants had to pick a system to focus on during survey

Most common application domains

- Automotive, avionics, and consumer electronics

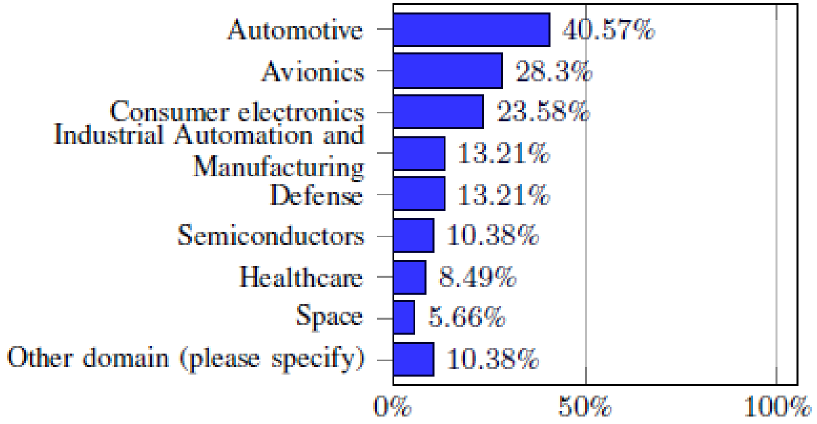
Many systems belong to several domains

- Largest overlap between avionics and defense (9%)

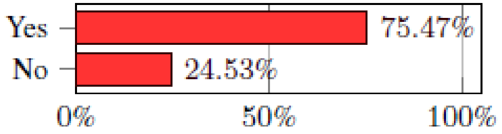
75% of considered systems had safety-critical parts

- 100% avionics, 91% automotive, and 52% consumer electronics

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



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



Results for Objective 1

Establish whether timing predictability is of concern to the real-time systems industry

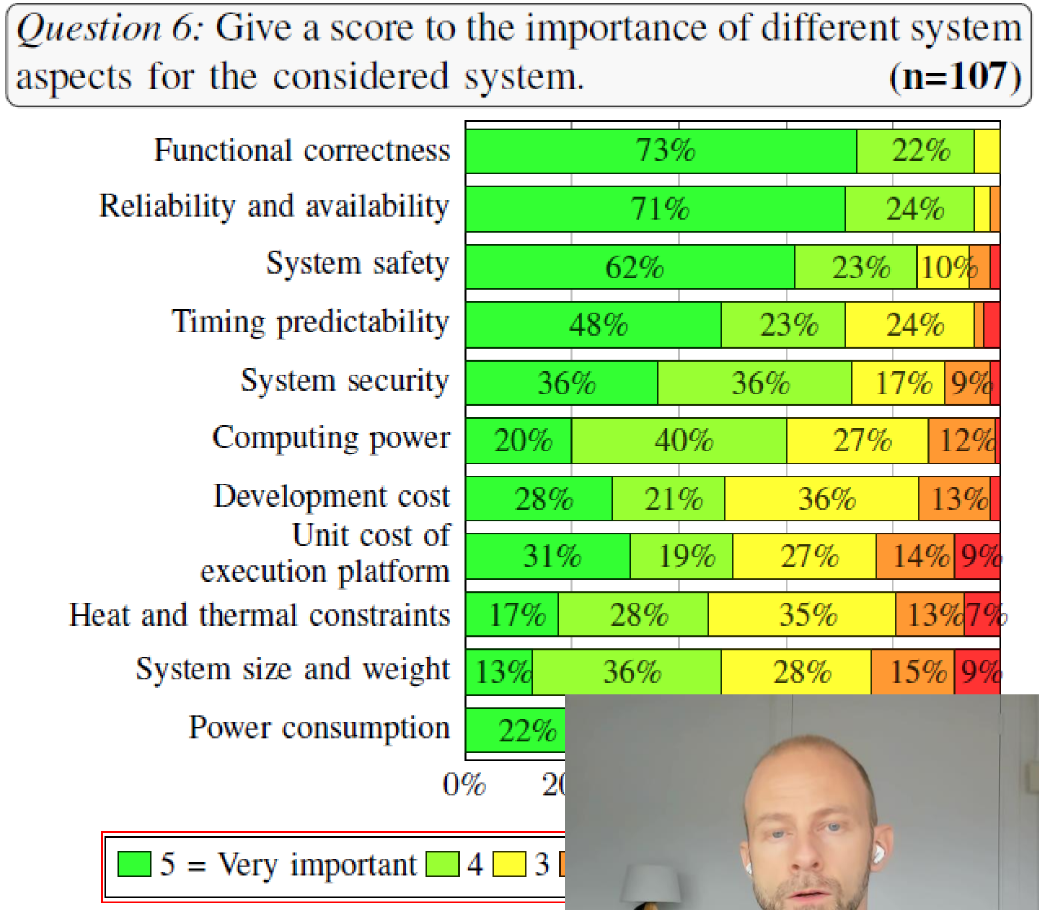
Importance of Timing Predictability

Although timing predictability is important, it is only one of many system design aspects.

- Most see timing predictability as (very) important
- Less important than functional correctness, reliability/availability, and safety, across domains

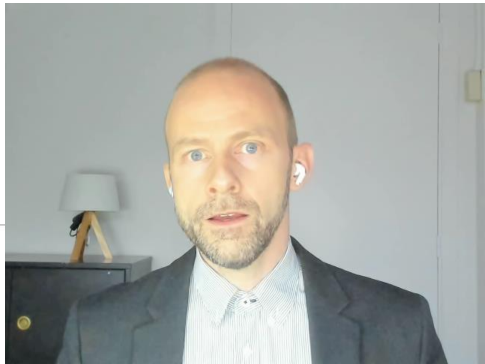
Participants indicating timing predictability / unit cost as "very important" aspects per domain

- 87% / 7% in avionics
- 48% / 45% in automotive
- 26% / 32% in consumer electronics



Results for Objective 2

Identify relevant industrial problem contexts, including hardware, middleware, and software



Hardware Platforms are Complex

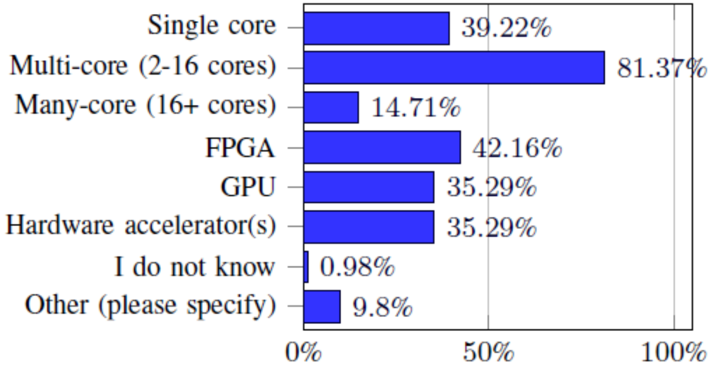
A majority of systems (>81%) include multi-core components, while a minority (<40%) contain single-core components

Little over 1/3 systems include FPGA, GPU, and hardware accelerators

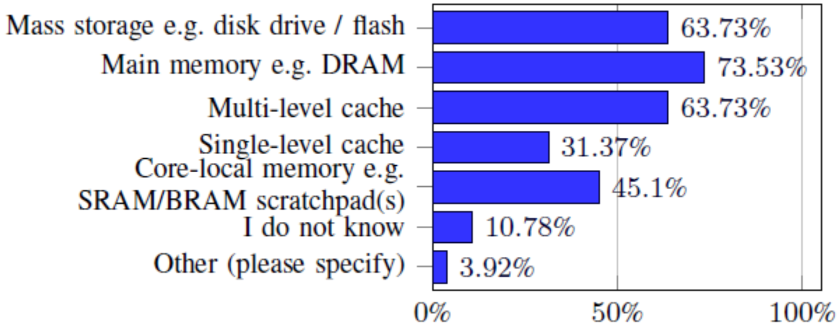
A majority of systems (>63%) have elements of a complex memory hierarchy

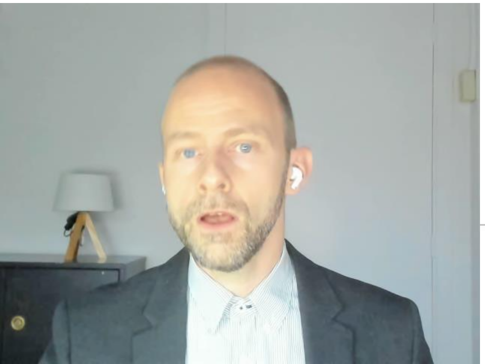
- Mass storage devices, DRAM, and multiple levels of cache

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



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





Hardware Platforms are Distributed

A majority of systems (73%) are considered distributed, while less than 17% only contain a single node

Wireless networks used in 25% of systems

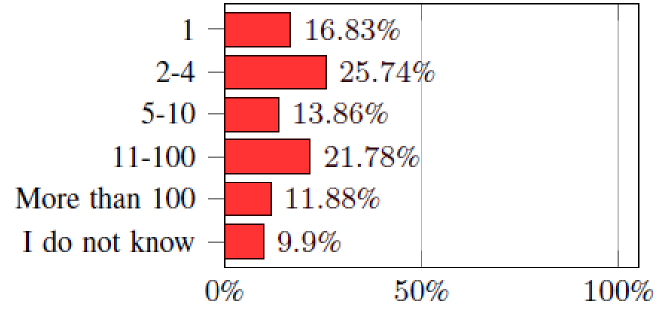
- Wireless the only network in just 9% of systems

Wired connectivity more common than wireless

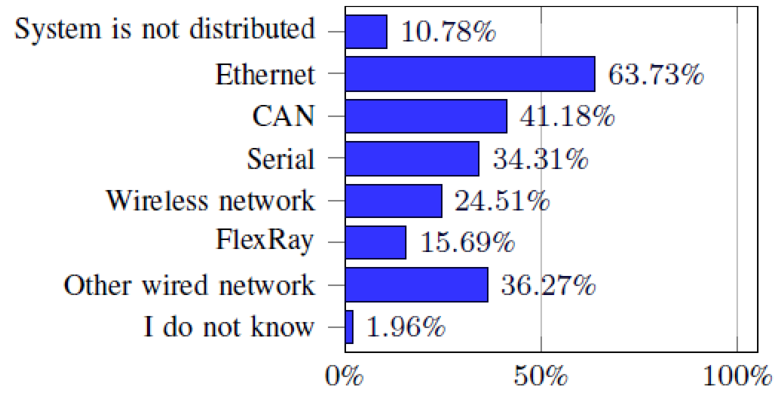
- Ethernet, CAN, and Serial most popular
- 48% of systems use multiple types of wired networks

Most automotive systems (74%) include CAN, while 34% include FlexRay

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



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



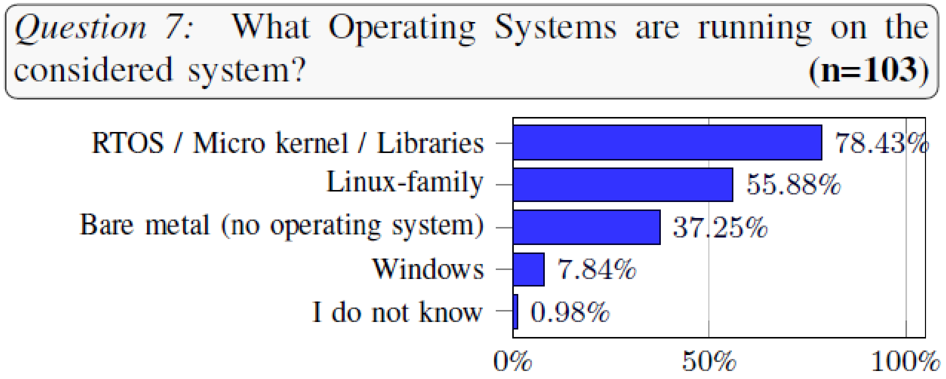
Multiple Different Operating Systems

Multiple different types of operating systems (OS) are used, often within the same system

63% of systems use more than one OS

- Most commonly combination between RTOS and Linux or bare metal

RTOS more prevalent in systems with safety-critical components, while the opposite holds for Windows



Different Types of Timing Constraints in a System

Most systems (90%) have some type of timing constraints

- Unsurprising since the survey targeted practitioners in area of real-time systems

Many systems (62%) combine two more types of constraints

- 27% have all three types!

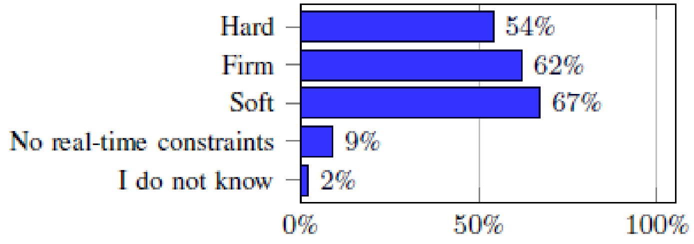
Only a few systems have one type of constraints

- Hard 5%, Firm 10%, and Soft 15%

Hard constraints most common in avionics domain

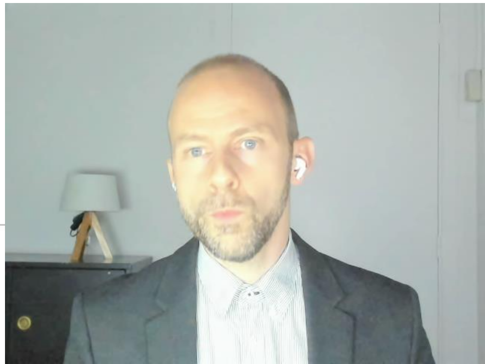
- Avionics 79%, automotive 56%, and 27% consumer electronics

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



Results for Objective 3

Determine which methods and tools are used to achieve timing predictability



WCET Estimation

Measurement-based timing analysis (MBTA) is more prevalent than static timing analysis (STA), but both are used

- 67% of responses use MBTA , 34% use STA, and 24% use both

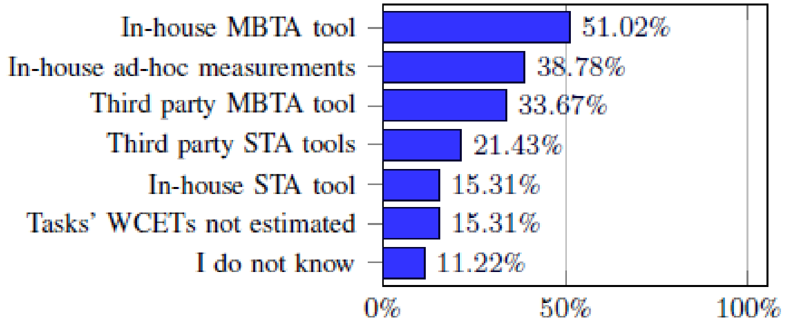
Difference is bigger with in-house tools

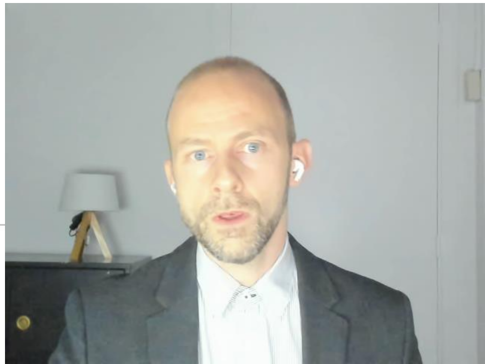
- > 50% use in-house MBTA compared to 15% STA

Distinction less stark for third-party tools

- 34% third-party MBTA vs. 21% STA

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





Improving Timing Predictability

Both static and dynamic methods of improving timing predictability are widely used

- >50% use watchdog timers, static schedule, and hardware selection to improve timing predictability

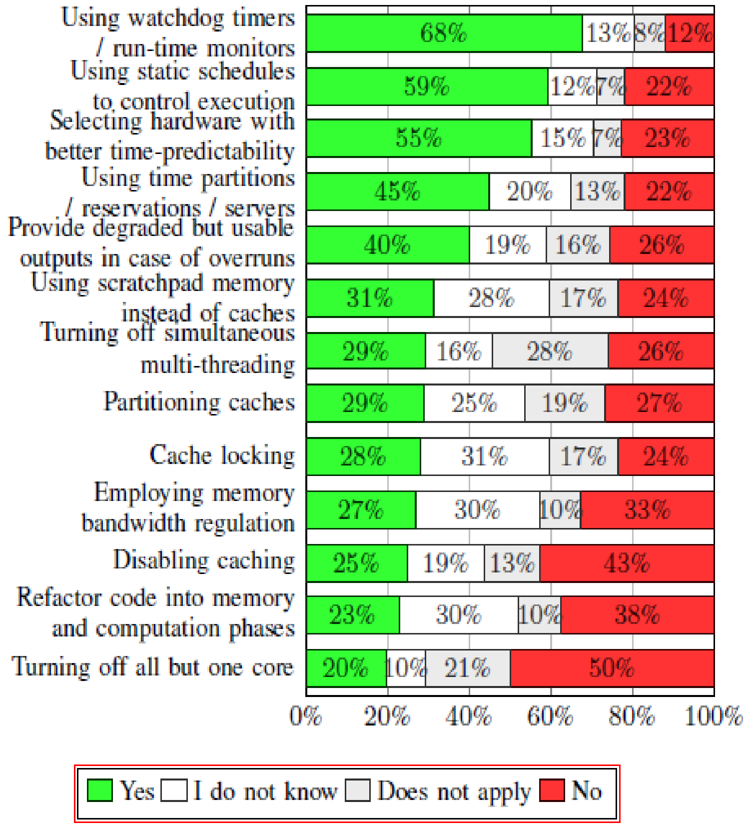
There is no silver bullet to improving timing predictability

- Each listed technique used by at least 20% of respondents
- 46% answered ‘yes’ to at least 5 techniques

Substantial uncertainty about some techniques

- > 20% “I do not know” for 6 techniques

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



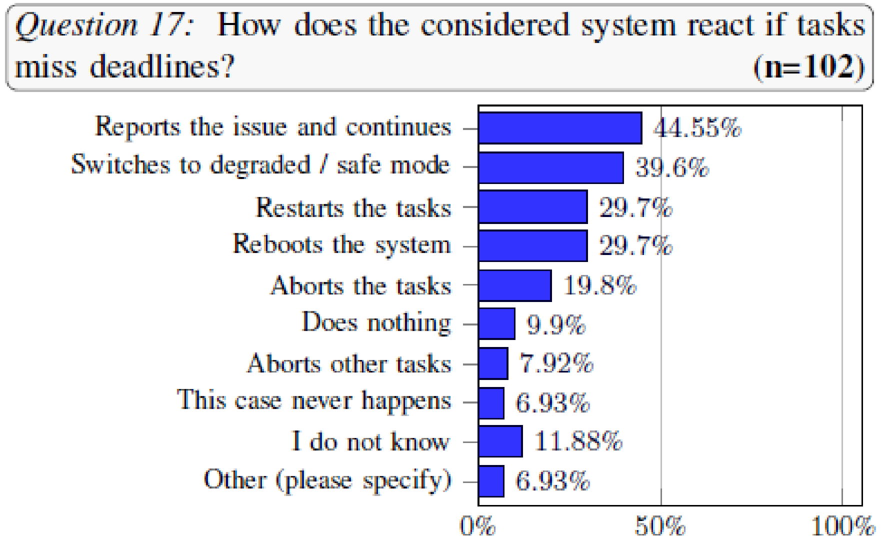
Reaction to Deadline Misses

Systems often take mitigating actions in the event of timing violations

- Most common (45%) is to report the issue and continue
- Many systems (40%) switches to safe mode
- 30% reboots the system, and 30% restarts the task

Safety-critical systems more likely to take action and reboot

- 36% of critical system reboot vs. 8% for other systems
- Only 6% of critical systems do nothing vs. 21% for others



Task Activations

All four distinguished types of activations are relatively common

- Periodic and aperiodic activations are most common

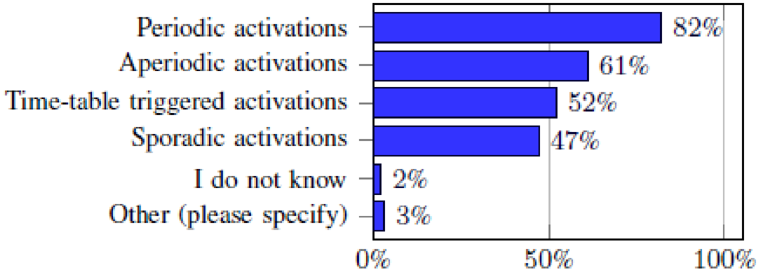
Some systems use only highly predictable task activation patterns

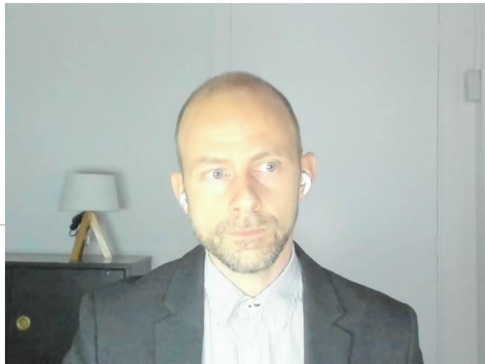
- 22% of systems have only periodic or time-triggered activations
- In contrast, only 4% and 2% had only sporadic or aperiodic activations, respectively

Most systems have more than one type of activations

- 74% indicated have least two types, and 25% have all four types!

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





Results for Objective 4

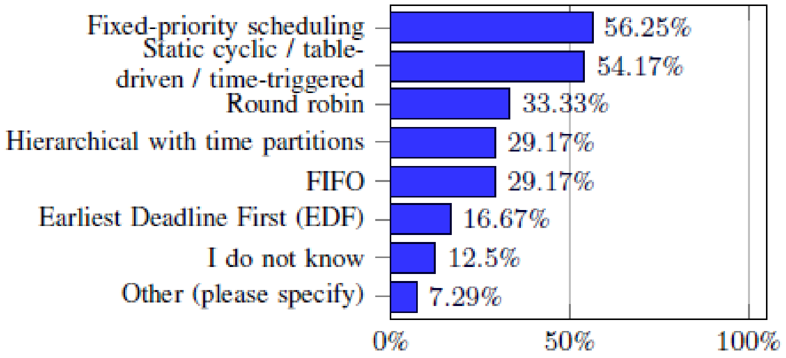
Establish which techniques and tools are used to satisfy real-time requirements

Scheduling Policies

Many different scheduling policies are used, some of which are not “real-time”

- Fixed-priority scheduling and static cyclic scheduling most common
- Fixed-priority scheduling 3x more common than EDF
- Round-robin and FIFO scheduling in approximately 1/3 of systems, despite not being “real-time”

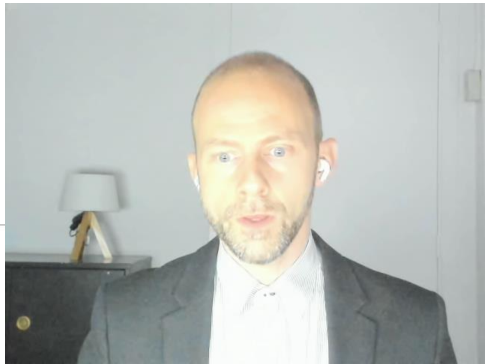
Question 20: Which task scheduling policy/policies are used in the considered system? **(n=97)**



EDF scheduling most frequent in automotive domain

- 27% in automotive, 11% consumer electronics, and 3% avionics

Most systems use two or more scheduling policies

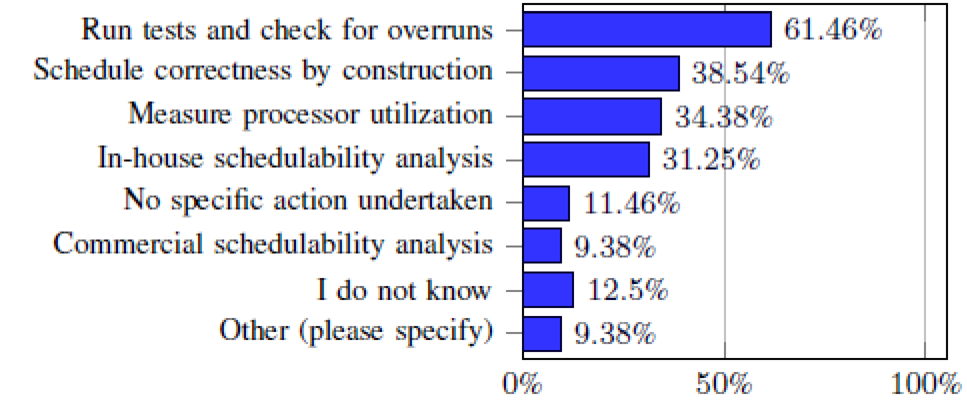


Timing Verification

The most common way to verify timing requirements is to run tests and check for overruns (61%)

- Most common static approach is to use schedule correctness by construction (39%)

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

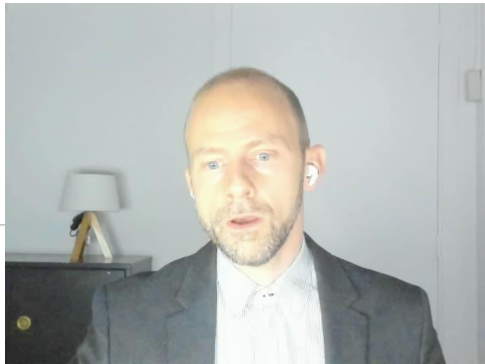


Less than 10% of respondents use commercial schedulability analysis tools

- More than 30% use in-house schedulability analysis

Consumer electronics most likely not to take action

- Consumer electronics (16%), automotive (12%), and avionics (0%)



Results for Objective 5

Determine trends for future real-time systems

Hardware Adoption Trends

Multi-core systems are widely used in development

- 80% indicate their use by 2021, and 10% ‘I do not know’

Adoption of heterogenous multi-core lags behind a bit

- 60% indicate their use by 2021, and 20% ‘I do not know’

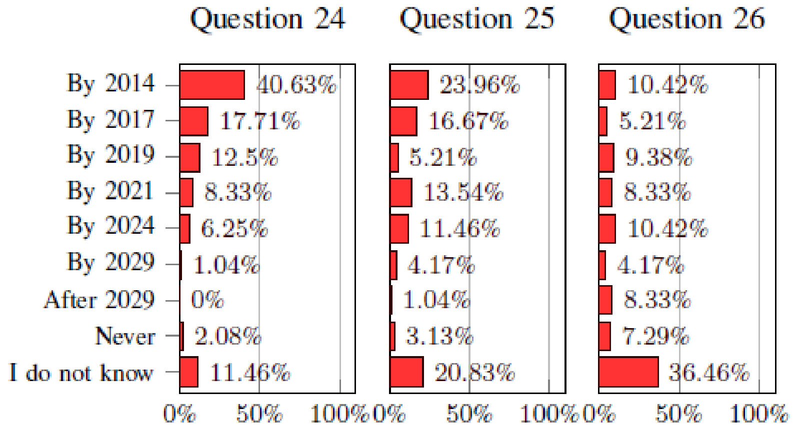
Many-core adoption is less certain

- 33% indicate use by 2021, and 36% ‘I do not know’

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)**



Single-core Trends

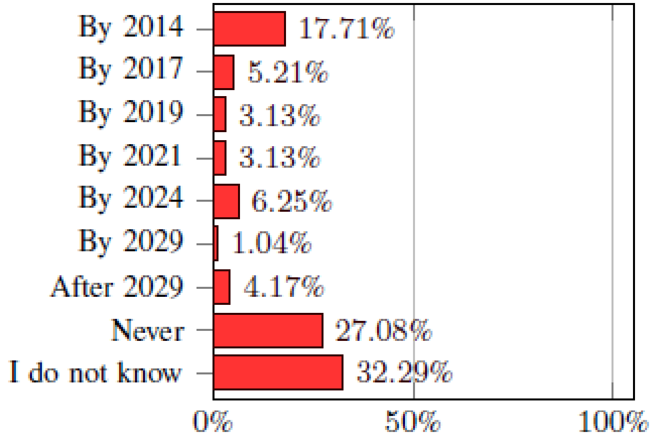
New projects with single-core processors are declining

- 28% of respondents expect new development projects to stop using single-cores by 2021

Single-core systems expected to remain relevant

- A substantial minority (31%) expect to use single-cores after 2029
- Expectation is similar for automotive (30%), avionics (35%), and consumer electronics (30%)

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)**



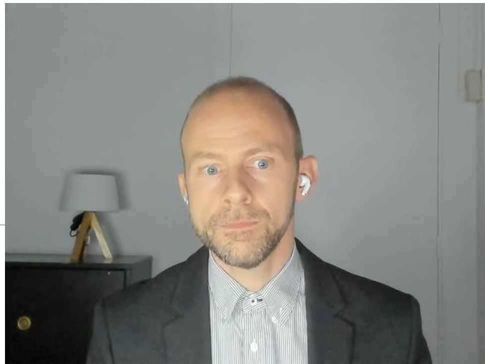
Presentation Outline

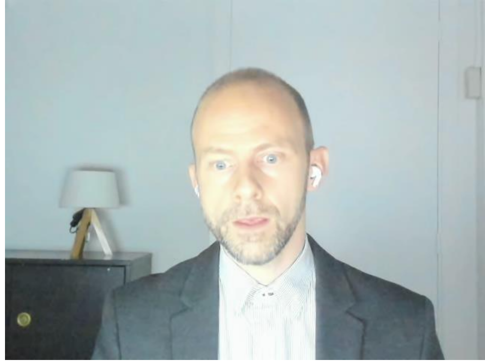
Introduction

Methodology

Results

Conclusions





Conclusions

The real-time community does not systematically research industry practice

- Can lead to a **divergence** between academic research and industry needs

This work addresses this problem through an empirical survey-based study

- 32 questions about methods, tools, and trends were asked to **120 industry practitioners**

Key results

- Timing predictability is important, but it is only one of many important system aspects
- Many systems are **distributed** and have hardware with **multiple** cores, complex memory hierarchy, **multiple** types of connectivity, operating systems, real-time requirements, etc.
- There is **no silver bullet**, but a wide range of techniques are used to increase timing predictability
- (Heterogeneous) multi-core and many-core systems are **increasingly adopted**, but single-core processors are expected to **stay relevant** on longer term