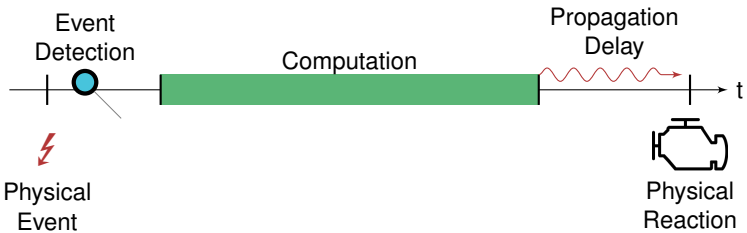
Static Response-Time Analysis of Complex Real-Time Systems: Time for Rethinking?

CASTOR Workshop on Dependable and Secure Systems
09. December 2020

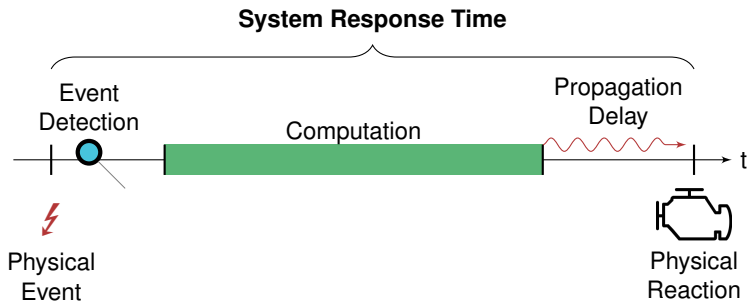
Peter Ulbrich, Simon Schuster, Tim Rheinfels, et al.

technische universität
dortmund

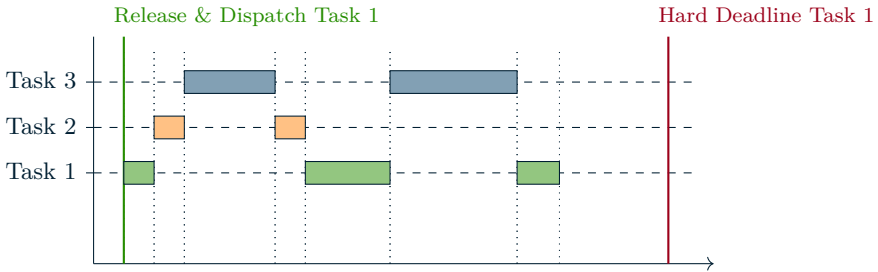
Response Time of Control Systems



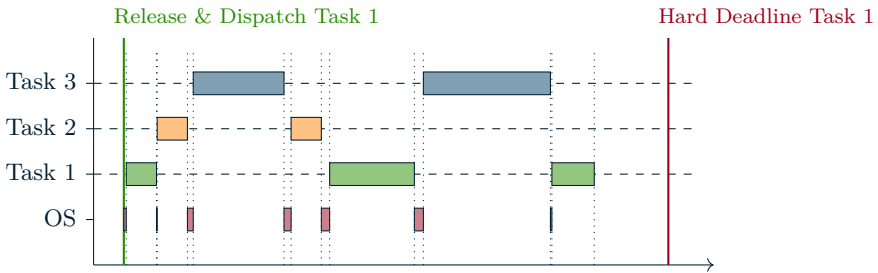
Response Time of Control Systems



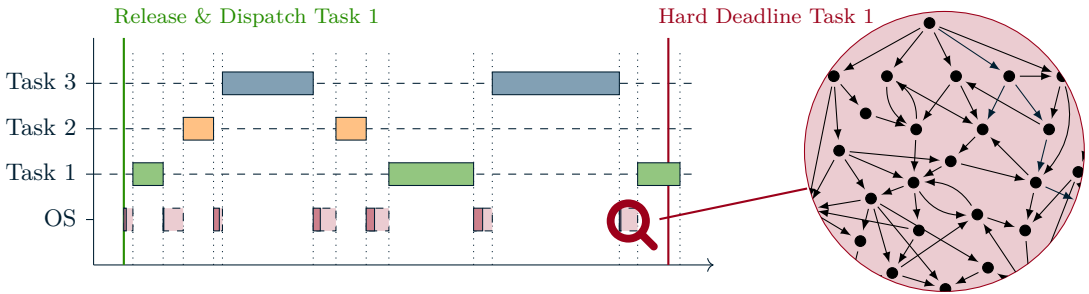
Scheduling View: Worst-Case Execution Times



Operating-System Overheads

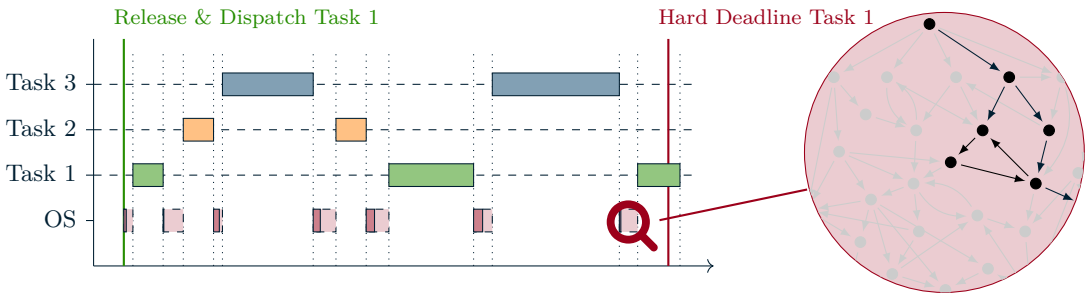


Operating-System Overheads



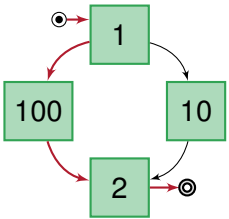
✗ **High level of pessimism** due to missing context information

Operating-System Overheads

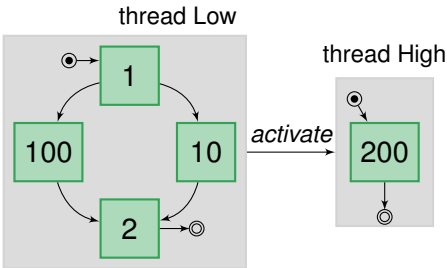


✗ **High level of pessimism** due to missing context information

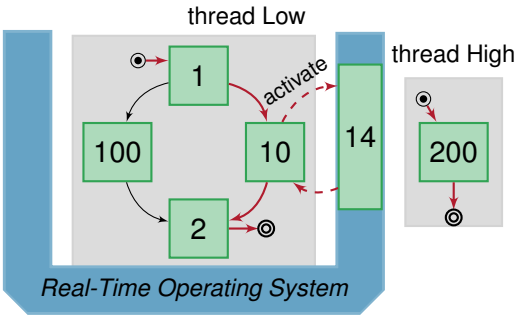
Whole-System Response-Time Analysis



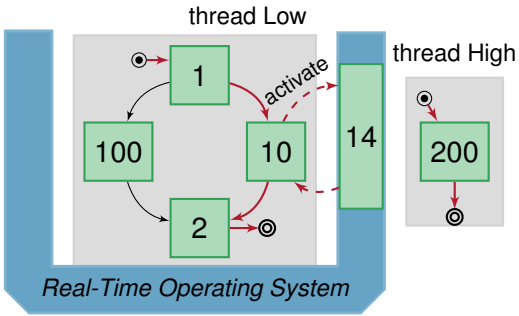
Worst-Case Response Time (WCRT): 103 cycles



Worst-Case Response Time (WCRT): 103 + 200 + t(RTOS) cycles?



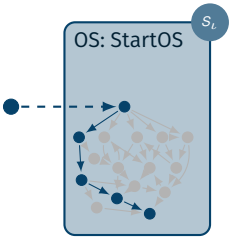
Worst-Case Response Time (WCRT): **331** cycles



Worst-Case Response Time (WCRT): **331** cycles

- Compositional approach on WCRT is **overly** pessimistic

→ Whole-system approach incorporating operating system semantics?

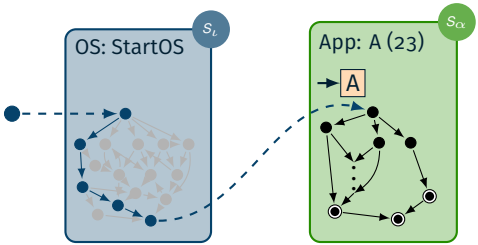


Status enumeration:

- Single-core system
- Fixed-priority scheduling
- OSEK-compatible

→ **Feasible system states**

Modeling Operating-System Semantics

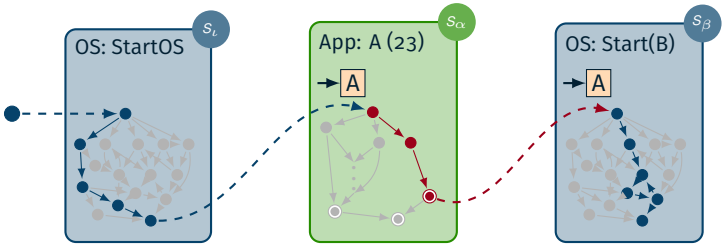


Status enumeration:

- Single-core system
- Fixed-priority scheduling
- OSEK-compatible

→ **Feasible system states**

Modeling Operating-System Semantics

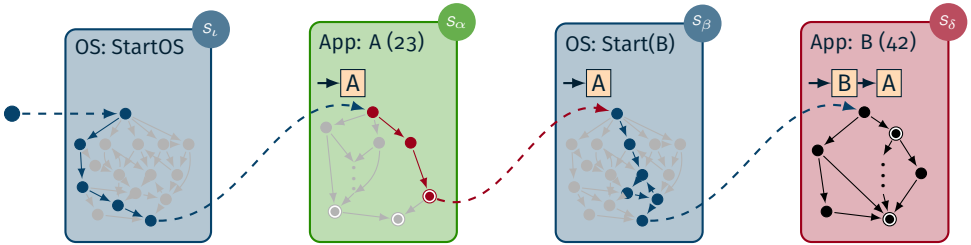


Status enumeration:

- Single-core system
- Fixed-priority scheduling
- OSEK-compatible

→ **Feasible system states**

Modeling Operating-System Semantics

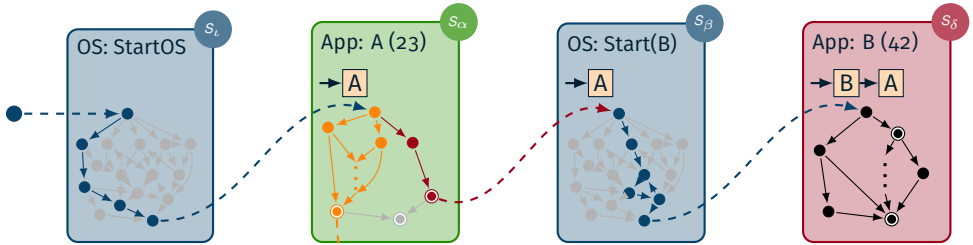


Status enumeration:

- Single-core system
- Fixed-priority scheduling
- OSEK-compatible

→ **Feasible system states**

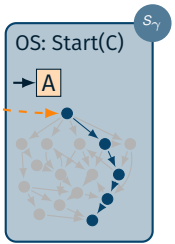
Modeling Operating-System Semantics



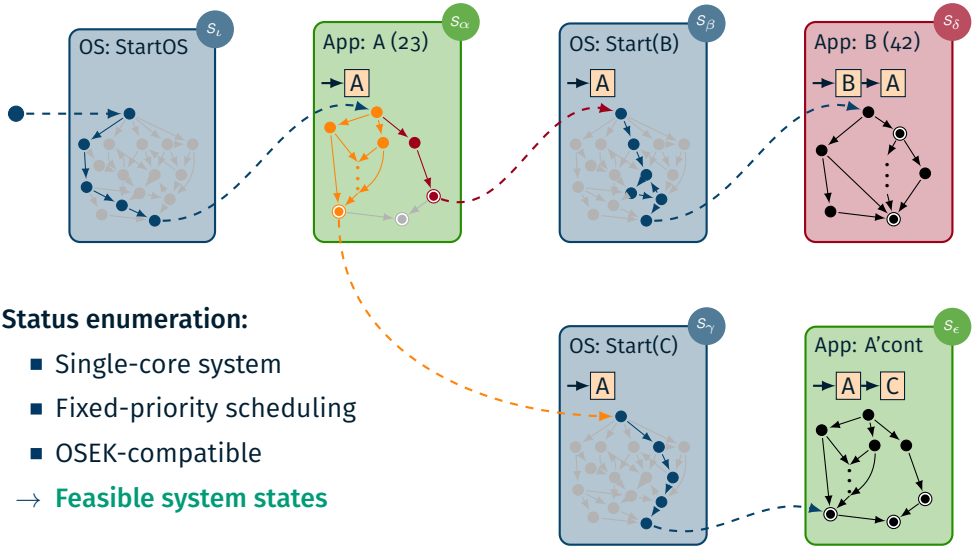
Status enumeration:

- Single-core system
- Fixed-priority scheduling
- OSEK-compatible

→ **Feasible system states**



Modeling Operating-System Semantics

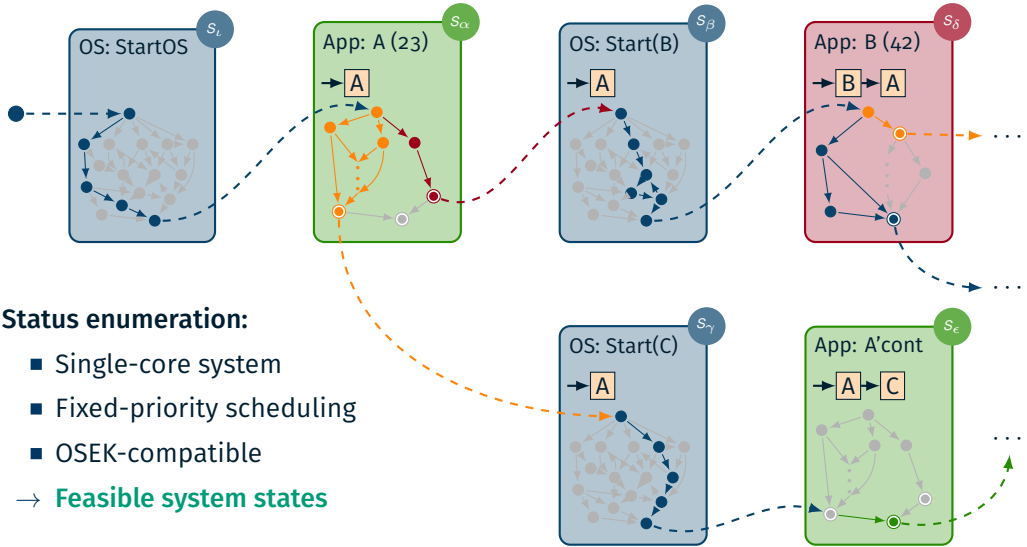


Status enumeration:

- Single-core system
- Fixed-priority scheduling
- OSEK-compatible

→ **Feasible system states**

Modeling Operating-System Semantics

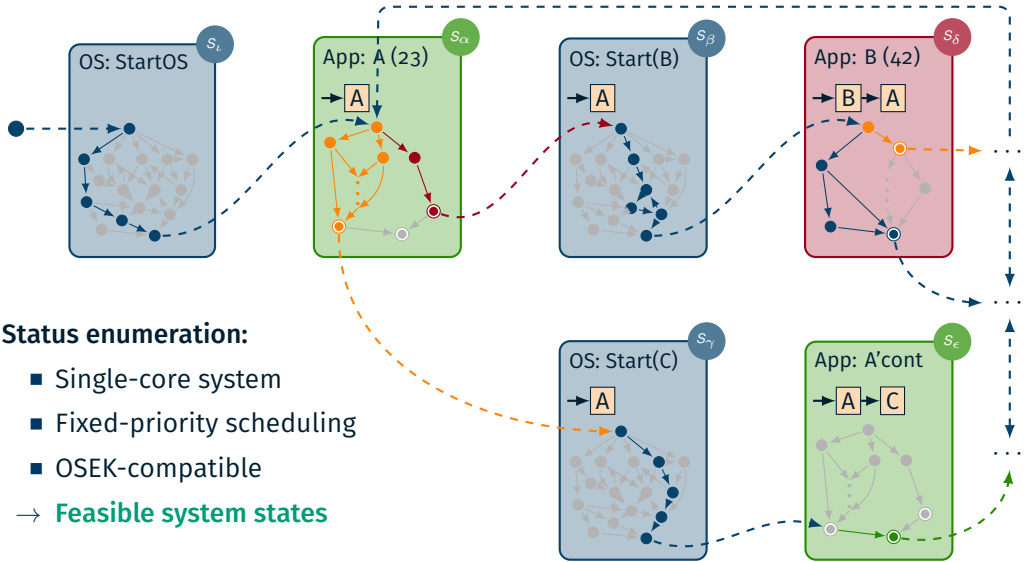


Status enumeration:

- Single-core system
- Fixed-priority scheduling
- OSEK-compatible

→ **Feasible system states**

Modeling Operating-System Semantics

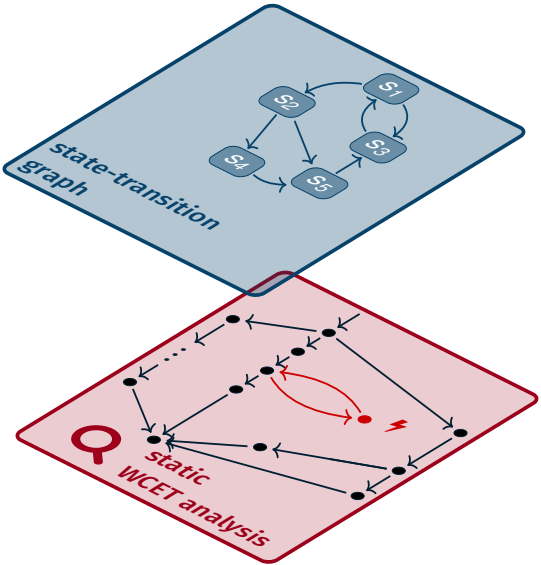


Status enumeration:

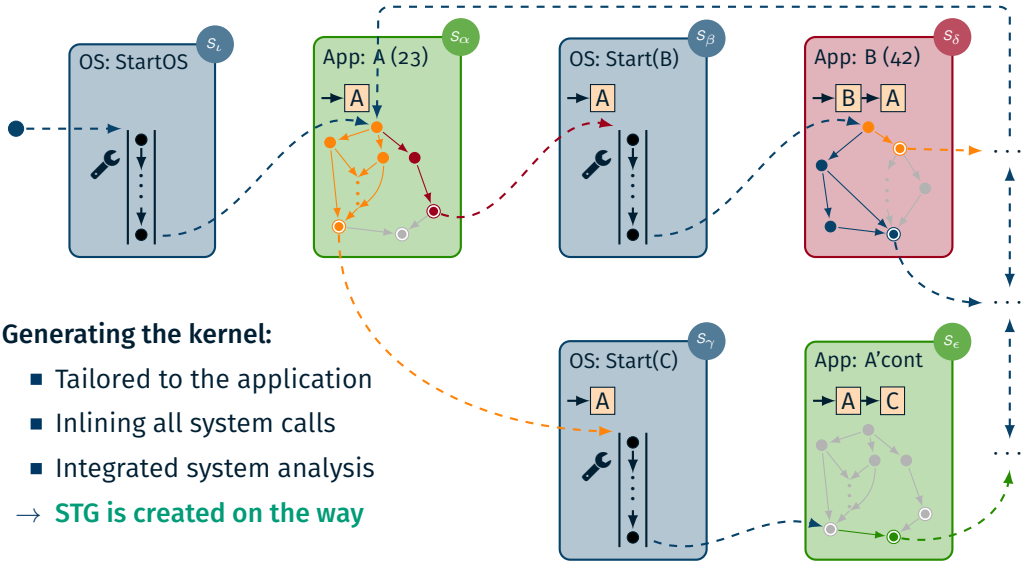
- Single-core system
- Fixed-priority scheduling
- OSEK-compatible

→ **Feasible system states**

Static Execution-Time Analysis of System Calls



Tailoring of System Calls

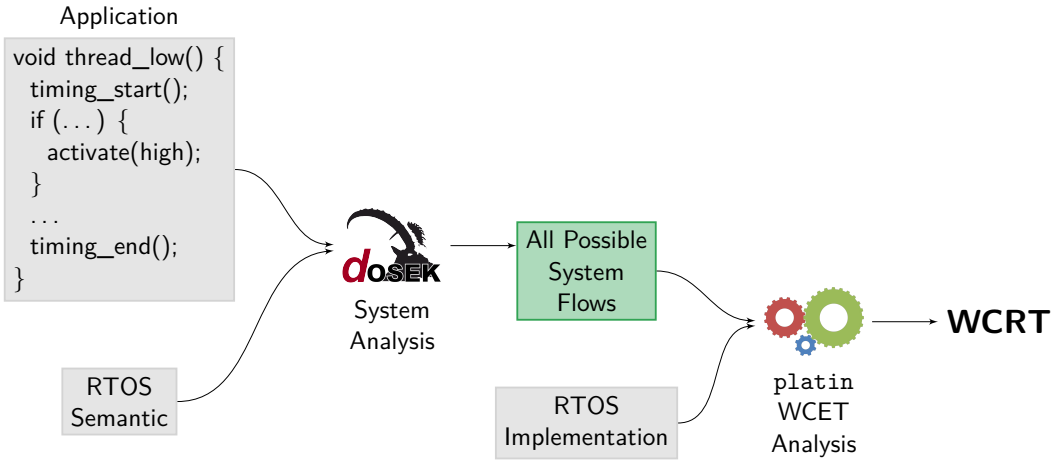


Generating the kernel:

- Tailored to the application
- Inlining all system calls
- Integrated system analysis

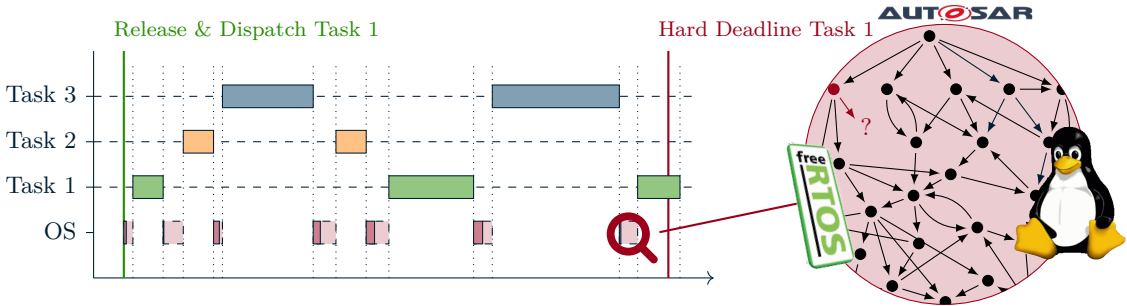
→ **STG is created on the way**

The SysWCET Approach – Overview



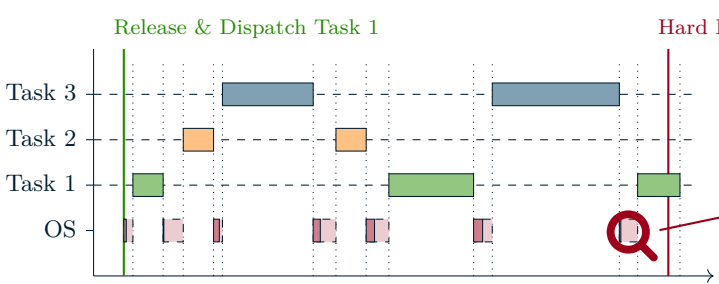
- Dietrich, Wägemann, Ulbrich, Lohmann.
SysWCET: Whole-System Response-Time Analysis for Fixed-Priority Real-Time Systems.
23rd Real-Time and Embedded Technology and Applications Symposium (*RTAS'17*)

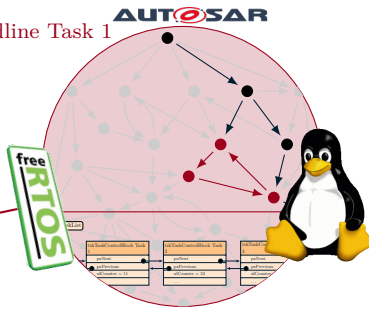
Operating-System Overheads (cont.)



- ✗ **High level of pessimism** due to missing context information
- ✗ **Control-flow reconstruction** difficult in some cases

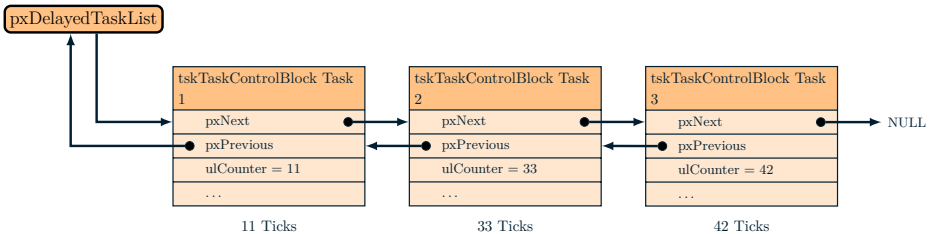
Operating-System Overheads (cont.)



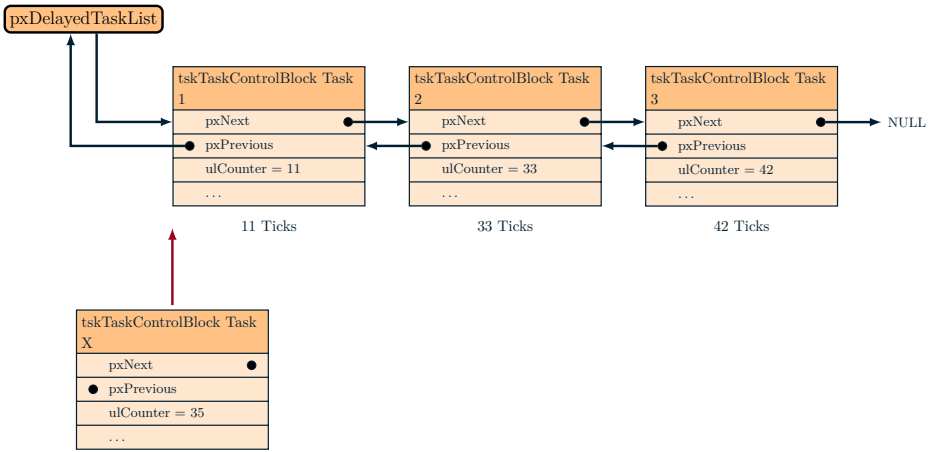


- ✗ **High level of pessimism** due to missing context information
- ✗ **Control-flow reconstruction** difficult in some cases
- ✗ **Indeterminable upper bound** due to application-dependent data structures

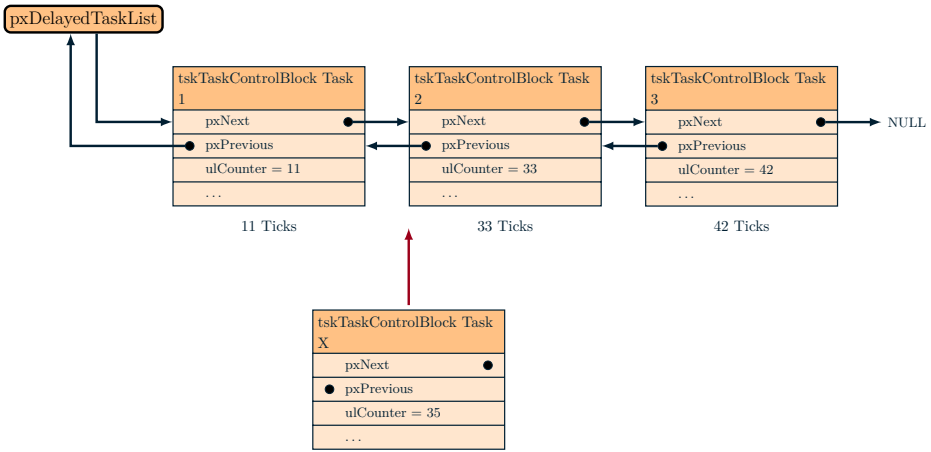
Operating System's Dynamic Data Structures



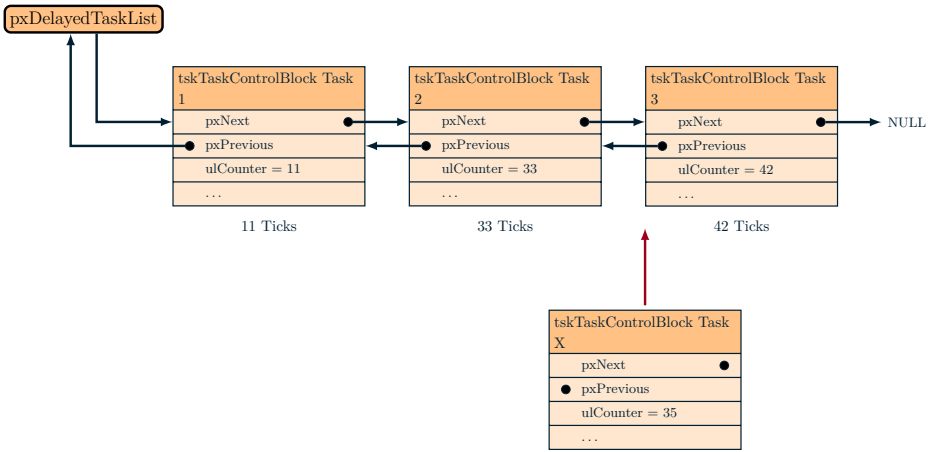
Operating System's Dynamic Data Structures



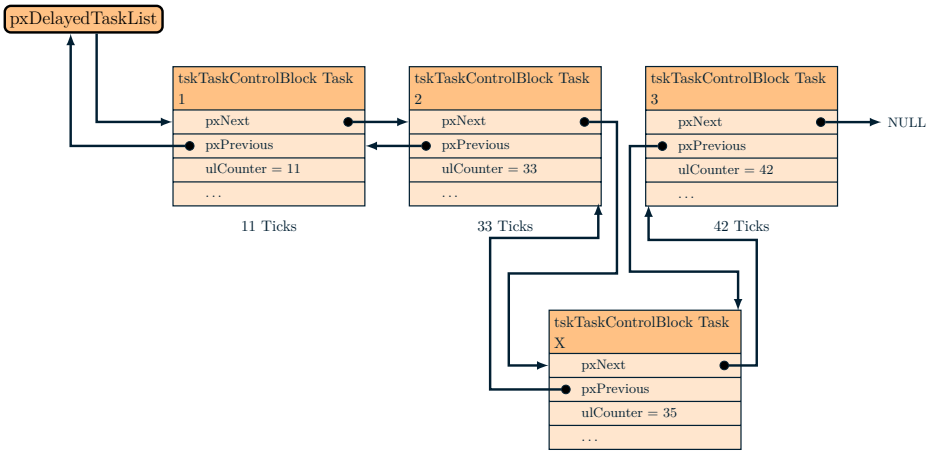
Operating System's Dynamic Data Structures



Operating System's Dynamic Data Structures

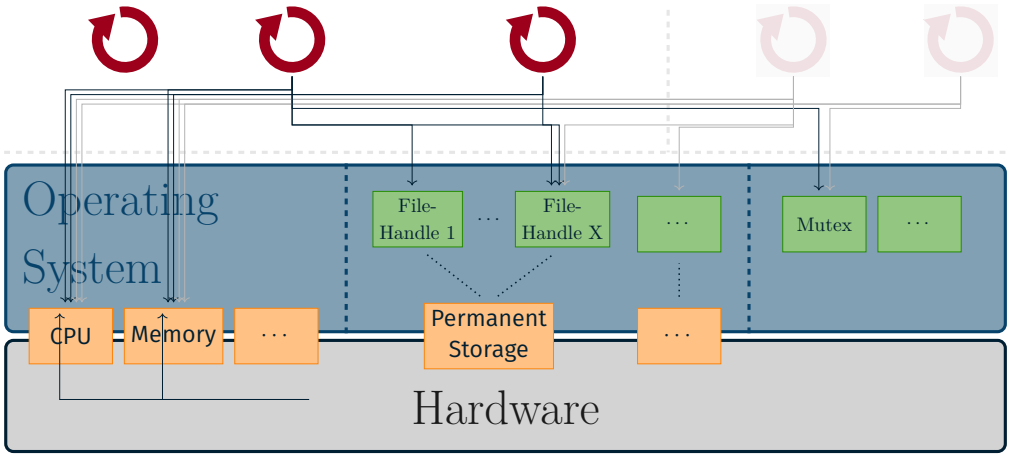


Operating System's Dynamic Data Structures

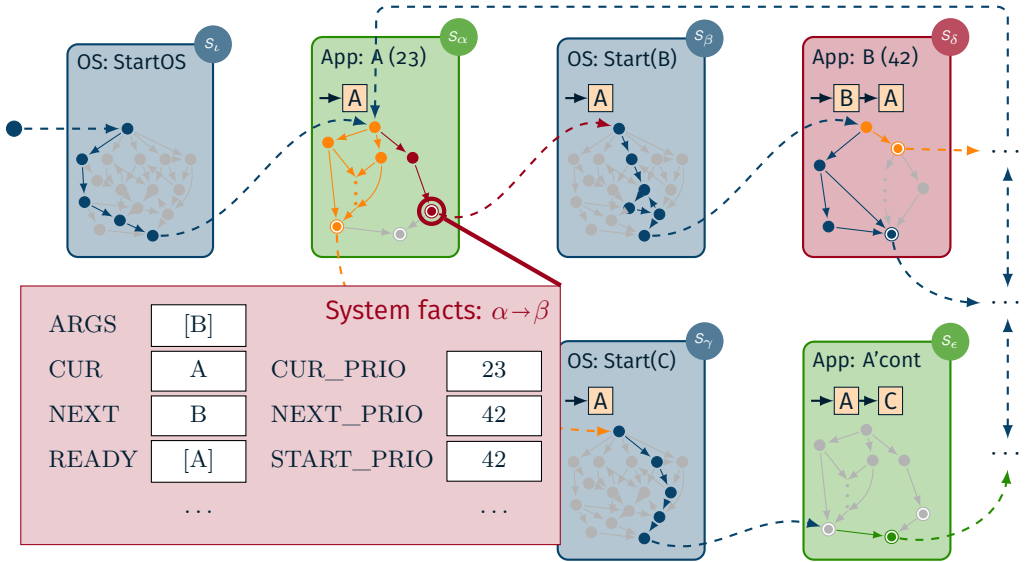


Operating System's Dynamic Data Structures

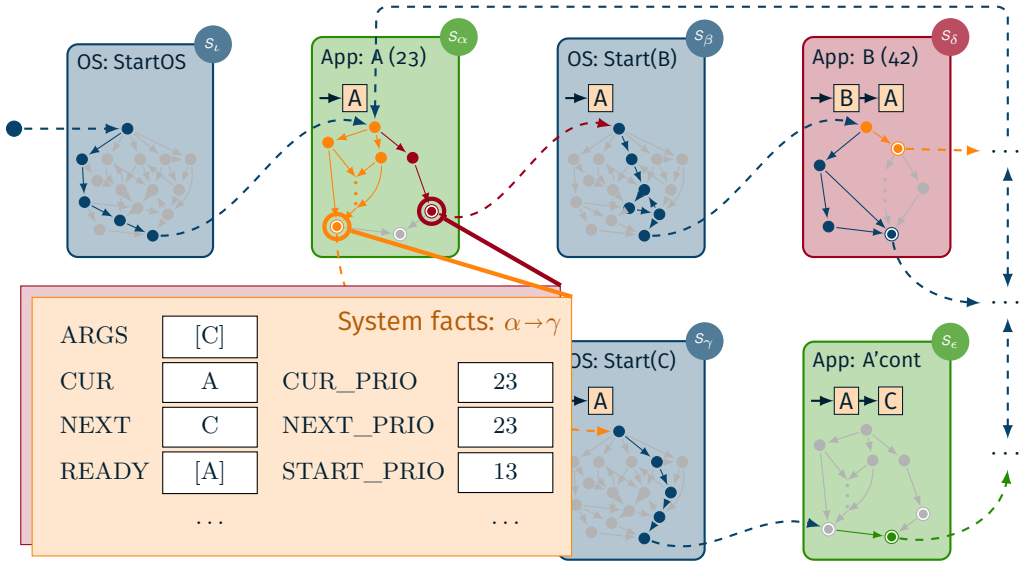
There are plenty of them!



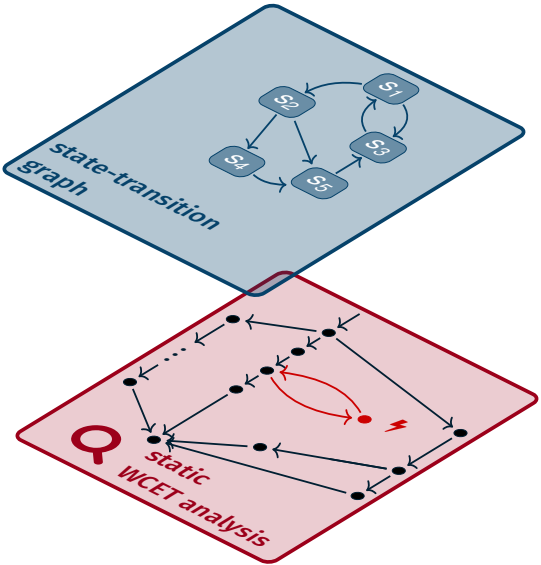
Deriving System Facts From the State Graph



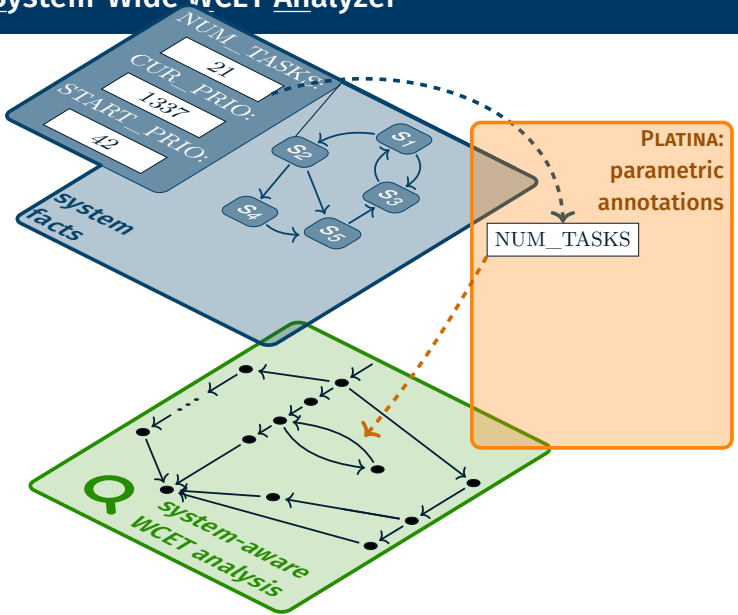
Deriving System Facts From the State Graph



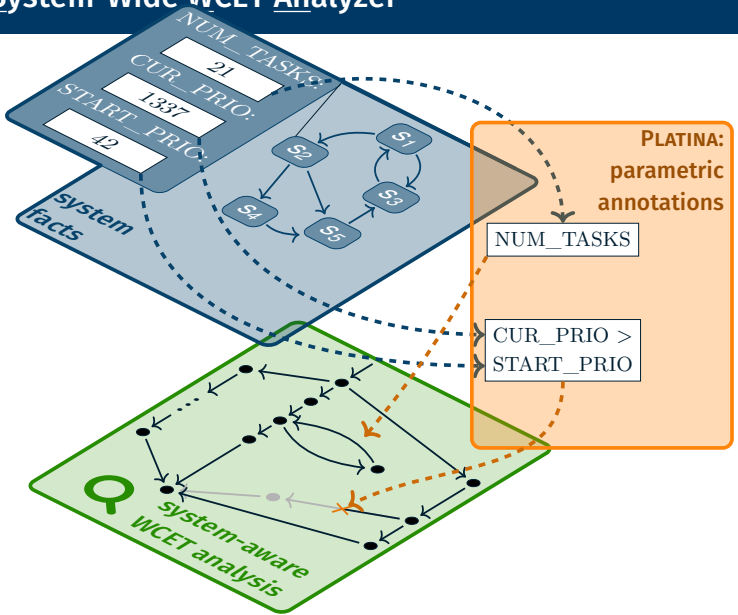
Static Execution-Time Analysis of System Calls



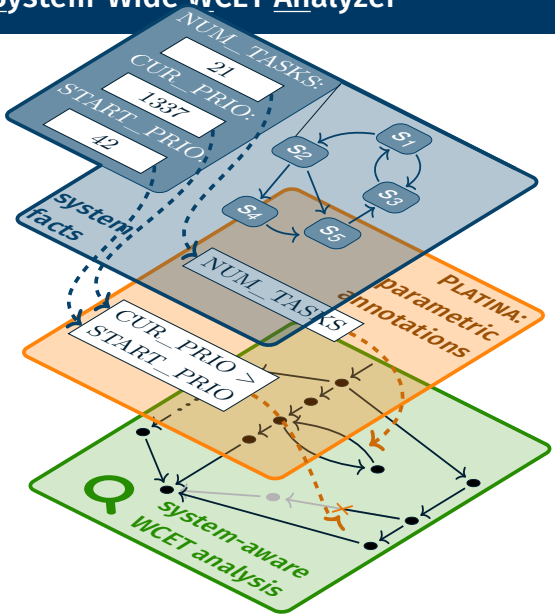
SWAN: System-Wide WCET Analyzer



SWAN: System-Wide WCET Analyzer

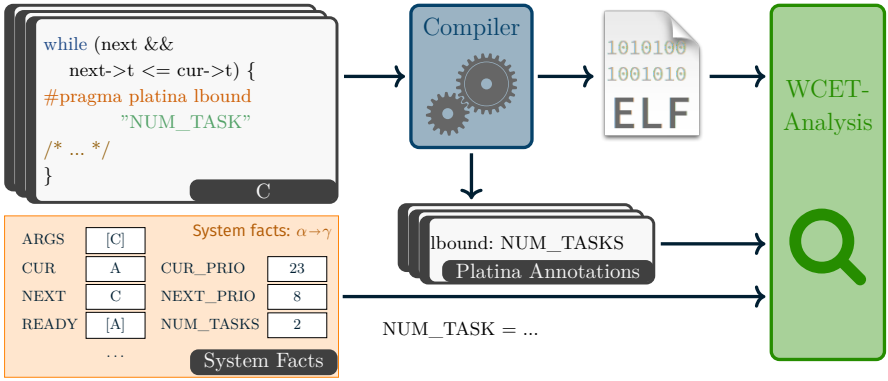


SWAN: System-Wide WCET Analyzer



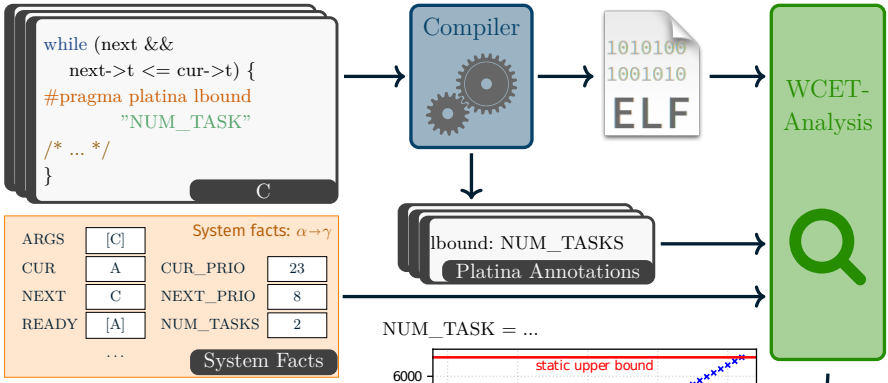
Information transport

The SWAN Toolchain



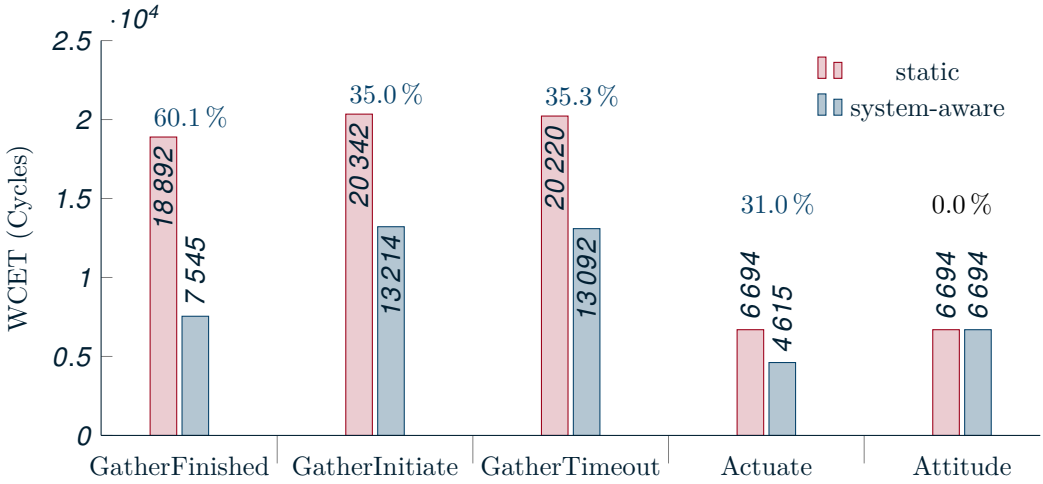
▶ Schuster, Wägemann, Ulbrich, Schröder-Preikschat.
Proving Real-Time Capability of Generic Operating Systems by System-Aware Timing Analysis.
25th Real-Time and Embedded Technology and Applications Symposium (*RTAS'19*)

The SWAN Toolchain

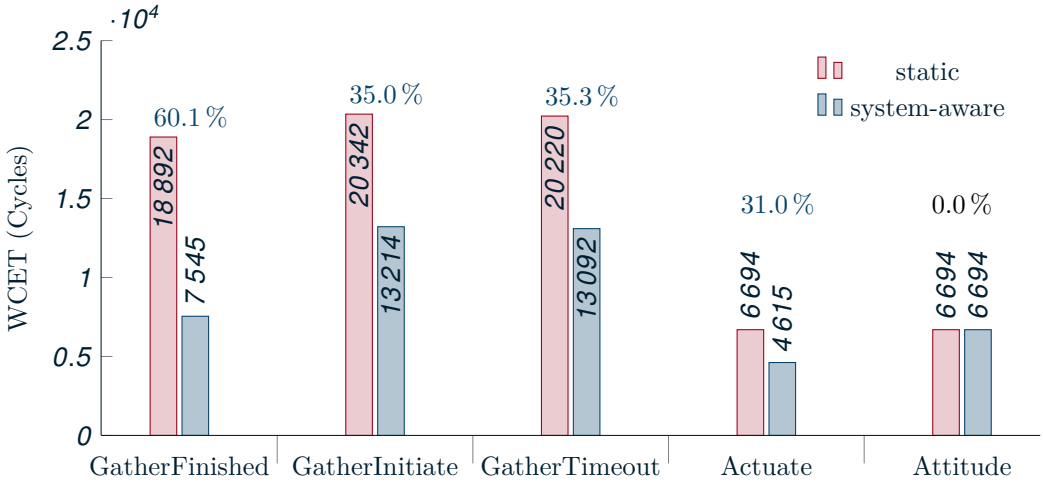


▶ Schuster, Wägemann, Ulbrich, Schröder-Preikschat.
Proving Real-Time Capability of Generic Operating Systems by System-Aware Timing Analysis.
25th Real-Time and Embedded Technology and Applications Symposium (*RTAS'19*)

Evaluation Results



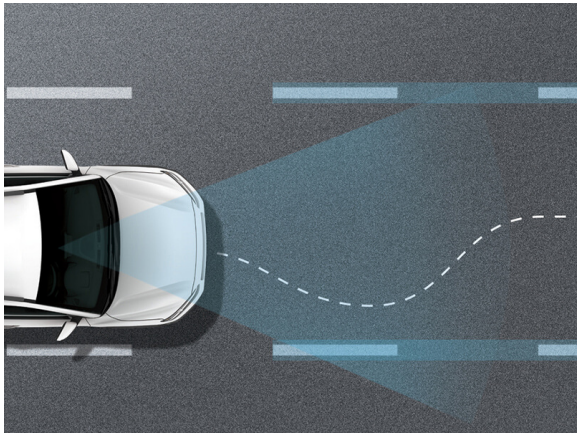
Evaluation Results



⇒ Reduction in WCRT by up to ***40.7***%

Runtime Adaptivity and Application-Centric Abstractions

Motivating Example: Human Drivers



Goal: Safe driving

- Keep lane adequately
- Omit obstacles

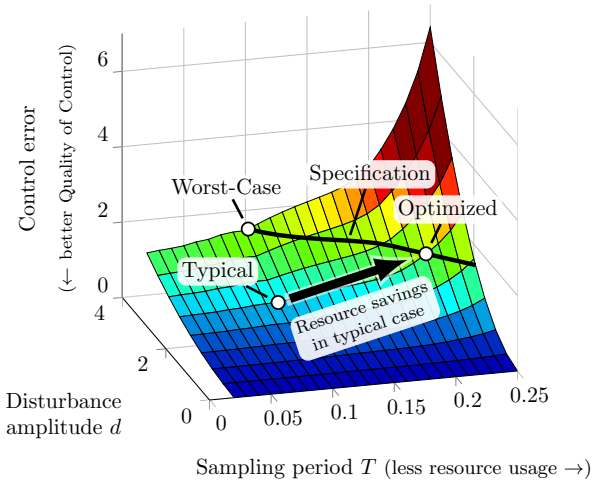
→ **Even in the worst case**

Situational awareness

- Reduced attention in normal traffic
- Focused in emergency situations

→ **Efficient use of resources**

Application's Work and OS's Contribution



Quality of control (QoC)

- Characterized by control error
- Degraded by environmental disturbance
- Implies resource demand

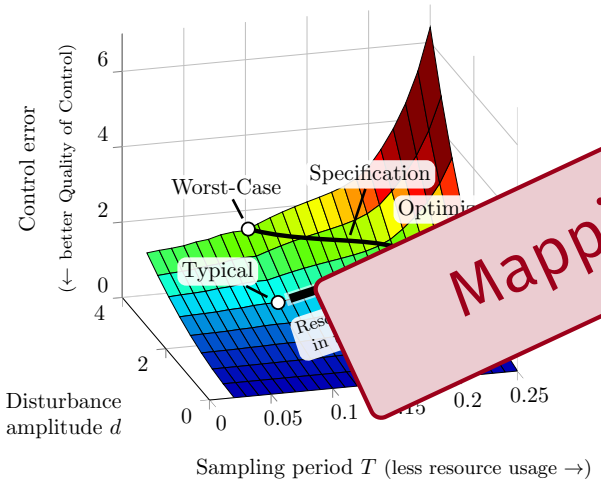
→ **Application-level constraints**

Quality of service (QoS)

- Resource allocation (periodicity)
- Deadline obedience

→ **OS-level timing constraints**

Application's Work and OS's Contribution



Mapping?

Quality of control (QoC)

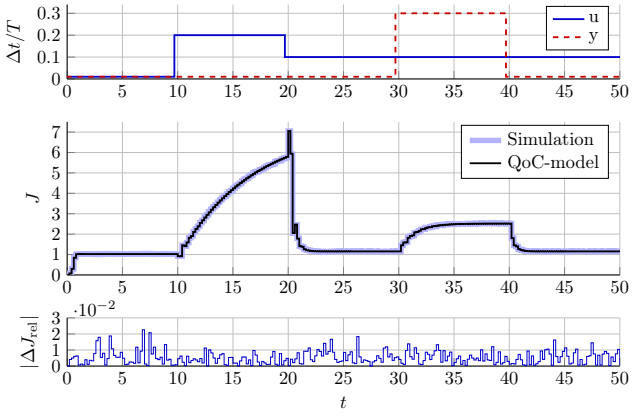
- Ch erized by control error
- by environmental
- e
- esource demand
- pplication-level constraints

Quality of service (QoS)

- Resource allocation (periodicity)
- Deadline obedience

→ **OS-level timing constraints**

Linking Qualities of Control and Service



Job-level runtime adaptivity

- Application constraints $\neq$ controller stability
- QoC is state-dependent

$\rightarrow$ **QoC prediction non-trivial**

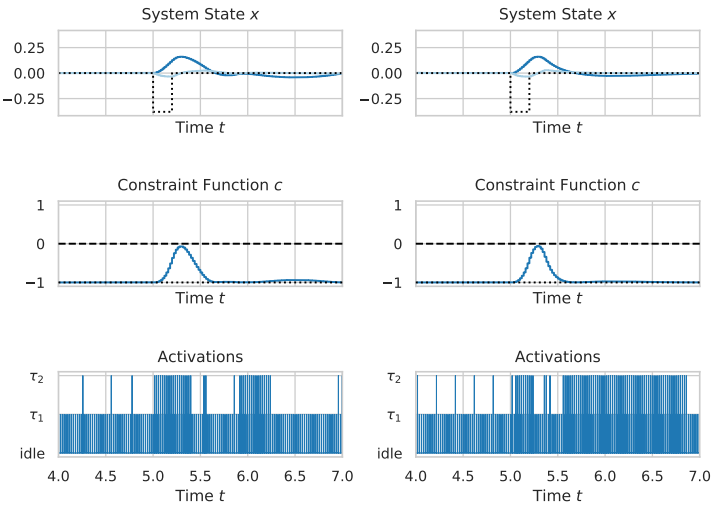
QRONOS: Quality-Aware Real-Time Control Systems

- QoC model (stochastic or deterministic)
- QoS mapping (deadlines)

$\rightarrow$ **Efficient yet reliable**

- Gaukler, Michalka, Ulbrich, Klaus.
  A New Perspective on Quality Evaluation for Control Systems with Stochastic Timing.
  21st ACM International Conference on Hybrid Systems: Computation and Control (*HSCC '18*)

Experimental Results (Work in Progress)



Good QoC obedience

- ≈100 % observed experimentally
- 71.0 % drop rate (NN)
  79.2 % drop rate (MPC)

Low overheads

- x86: MPC 93 ms
  vs. NN 1 ms (worst-case)
- Cortex-M4: NN 50 µs

→ **Feasible for scheduling**

Conclusion

Conclusion

Whole-System Response-Time Analysis

- Based on semantic analysis of operating system states

→ **SysWCET: Code tailoring of system calls**

Conclusion

Whole-System Response-Time Analysis

- Based on semantic analysis of operating system states

→ **SysWCET: Code tailoring of system calls**

Tackling Universal Operating Systems

- Code annotation and context-aware analysis

→ **SWAN: Tailoring the analysis**

Conclusion

Whole-System Response-Time Analysis

- Based on semantic analysis of operating system states

→ **SysWCET: Code tailoring of system calls**

Tackling Universal Operating Systems

- Code annotation and context-aware analysis

→ **SWAN: Tailoring the analysis**

Future Real-Time Design

- Are we hitting the pessimism wall?
- Deadlines are (often) inept!

→ **QRONOS: Application-level constraints as 1st class citizen**