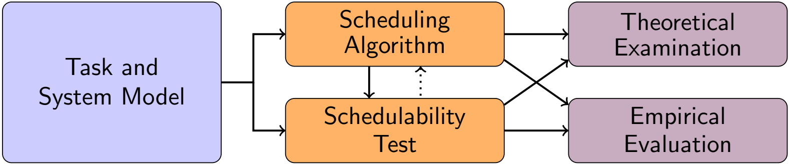
Realistic Scheduling Models and Analyses for Advanced Real-Time Embedded Systems

Georg von der Brüggen

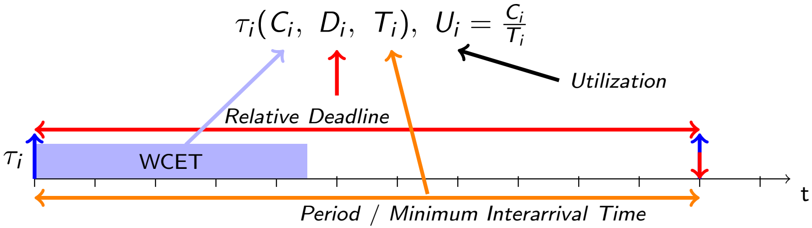
Department of Computer Science, Chair 12
TU Dortmund, Germany

20 November 2020

Modelling and Analysis of Real-Time Systems



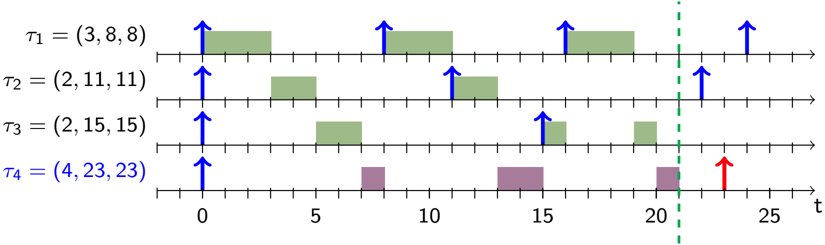
Periodic or Sporadic Task Model



- Set of recurrent tasks $\mathrm{T} = \{\tau_1, ..., \tau_n\}$
- Uniprocessor systems
- Implicit deadline: relative deadline = period

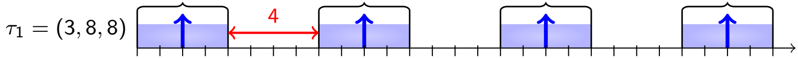
Static-Priority Preemptive Scheduling

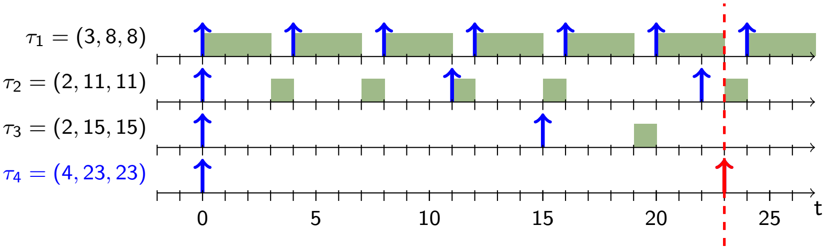
Worst-Case Response Time Analysis



- Periodic or sporadic tasks
- τ_1 highest priority
- τ_4 lowest priority

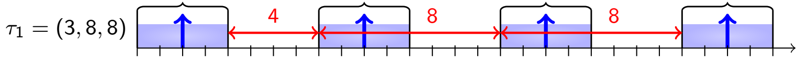
Static-Priority Preemptive Scheduling



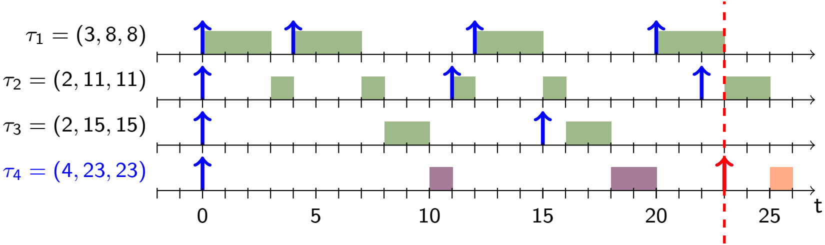


- τ_1 with release jitter $[-2; +2]$
- τ_1 highest priority
- τ_4 lowest priority

Static-Priority Preemptive Scheduling

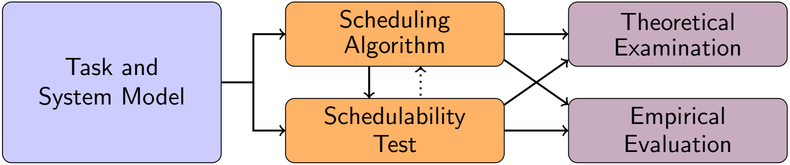


Worst-Case Response Time Analysis



- τ_1 with release jitter $[-2; +2]$
- τ_1 highest priority
- τ_4 lowest priority

Modelling and Analysis of Real-Time Systems



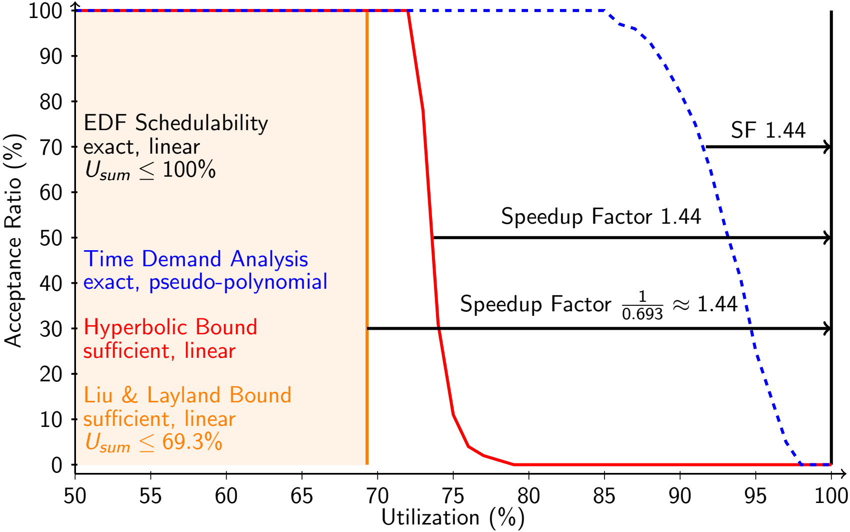
Hypothesis

Realistic scheduling models and analyses are essential for guaranteeing timing correctness in advanced real-time systems while ensuring that the system resources necessary to provide these guarantees are not over-provisioned.

Speedup Factors and Parametric Utilization Bounds

- Non-Preemptive Scheduling ECRTS 2015
- Exact Speedup Factors in Linear-Time IPL 177 (2017)
- Pitfalls of Resource Augmentation Factors and Utilization Bounds in Real-Time Scheduling ECRTS 2017
- Parametric Utilization Bounds for Implicit-Deadline Automotive Task Systems RTNS 2017

Evaluating Schedulability - Rate-Monotonic

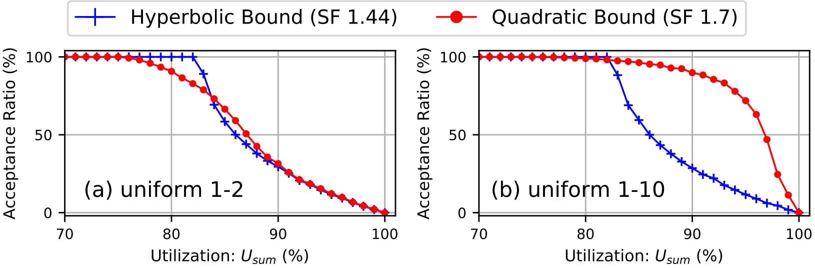


Exact Speedup Factors of DM vs EDF

Preemptive Bounds			
Constraints	lower	upper (linear)	upper (exp.)
implicit	≈ 1.44	≈ 1.44	≈ 1.44
constrained	≈ 1.76	≈ 1.76	≈ 1.76
arbitrary	2	2	2
Non-Preemptive Bounds			
Constraints	lower	upper (linear)	upper (exp.)
implicit	≈ 1.76	≈ 1.76	≈ 1.76
constrained	≈ 1.76	≈ 1.76	≈ 1.76
arbitrary	2	2	2

- No discrimination between the performance of different scheduling algorithms and schedulability tests
- Should only be considered for their negative implications
- Optimal speedup factors do not imply good performance

Non-Dominance Based on Speedup Factors

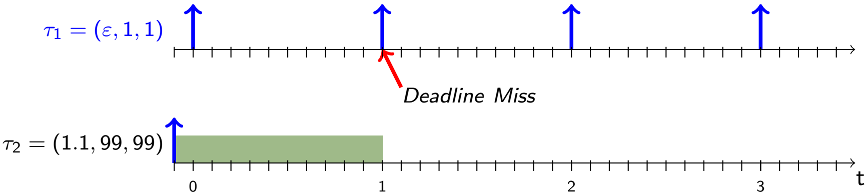


- Performance in evaluation may contradict speedup factors
- Identify regions of dominance

Parametric Augmentation Functions

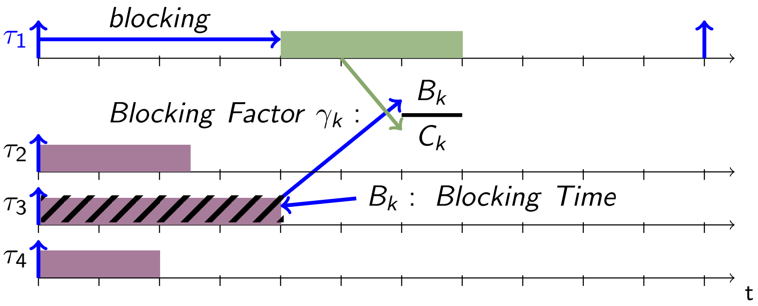
- Vector $\vec{x}$ with parameters of interest
- Parametric augmentation function $\mathcal{A}(\vec{x})$ based on the parameters in $\vec{x}$
- Detailed information about the actual performance across a wide range of parameter values
- Avoid singularities and unrealistic corner cases

Parametric Utilization Bounds for RM-NP



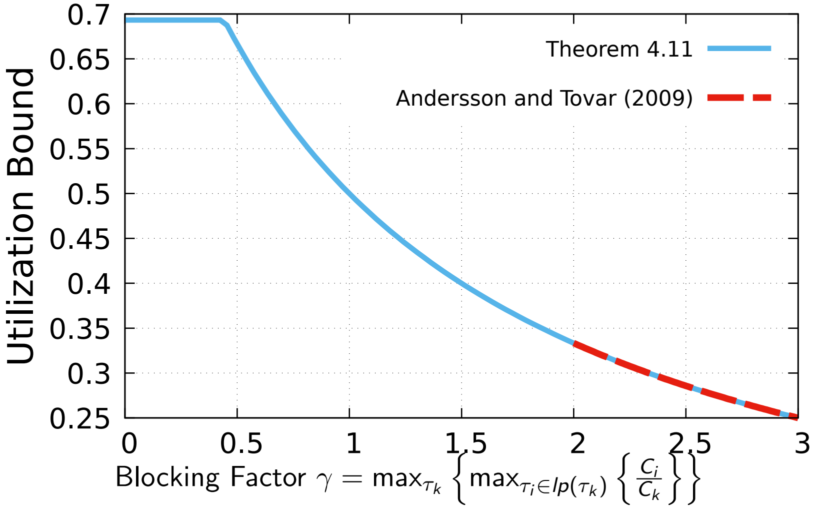
- Utilization Bound: 0
  - $\varepsilon \rightarrow 0$
  - $T_2 \rightarrow \infty$

Parametric Utilization Bounds for RM-NP



$$\textit{Task Set Blocking Factor } \gamma : \max_{\tau_k \in \tau} \{\gamma_k\}$$

Parametric Utilization Bounds for RM-NP

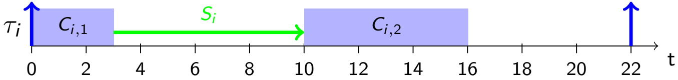


Self-Suspension

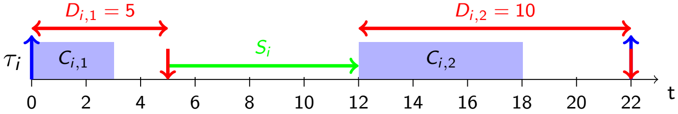
- Shortest Execution Interval First Deadline Assignment (SEIFDA) for Segmented Self-Suspension RTNS 2016
- Release Enforcement in Resource-Oriented Partitioned Scheduling for Multiprocessor Synchronization RTNS 2017
- Hybrid Self-Suspension Models RTCSA 2017

Fixed-Relative-Deadline Scheduling Strategies

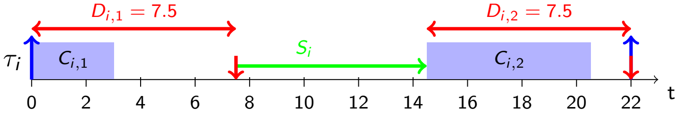
$$C_{i,1} = 3,\ C_{i,2} = 6,\ S_i = 7,\ T_i - S_i = 15$$



Proportional Deadline Assignment



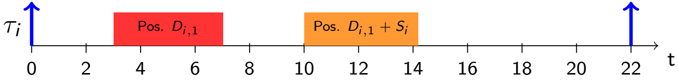
Equal Deadline Assignment



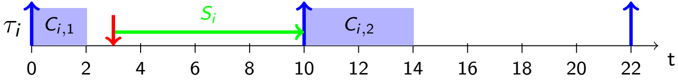
Shortest Execution Interval First Deadline Assignment

- Consider workload of previously assigned tasks

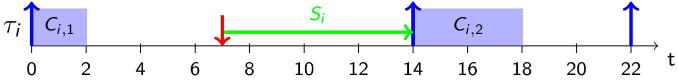
$C_{i,1} = 2,\ C_{i,2} = 4,\ S_i = 7,\ T_i - S_i = 15;$ Possible $D_{i,1}$: [3;7]



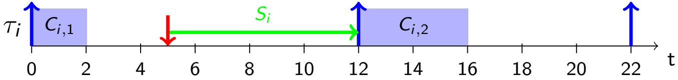
SEIFDA-minD:



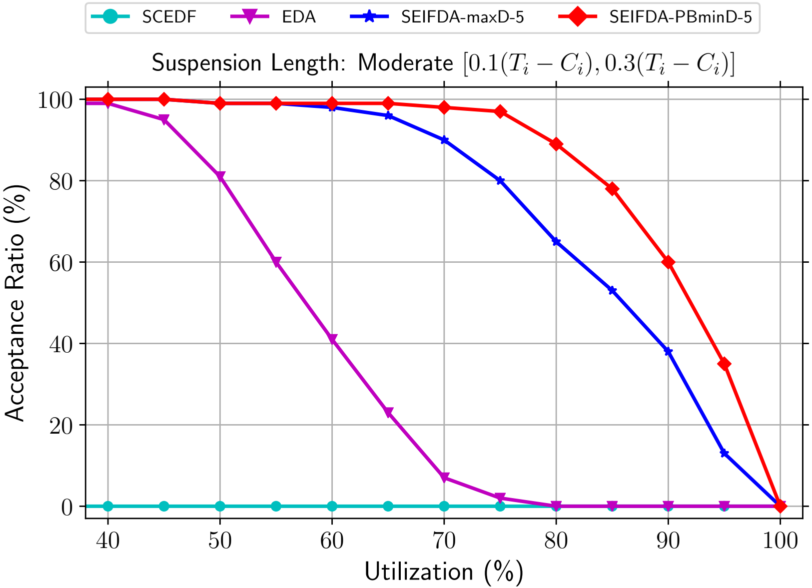
SEIFDA-maxD:



SEIFDA-PBminD:

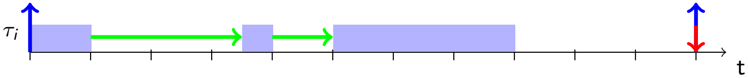


Comparison of SEIFDA and other Algorithms



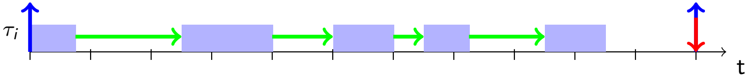
Segmented vs. Dynamic

Segmented self-suspension model



- Fixed execution / suspension pattern
  - Very accurate
  - Over-restrictive

Dynamic self-suspension model

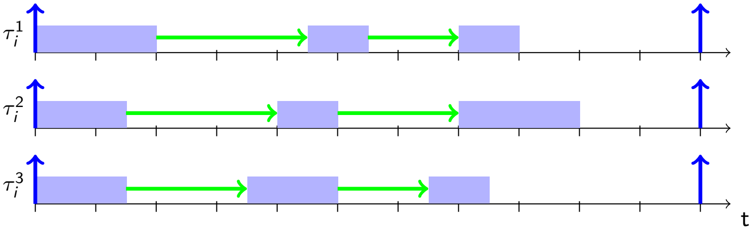


- Only maximum S_i as additional information
  - Over-flexible
  - Inaccurate

⇒ Huge gap regarding flexibility and accuracy

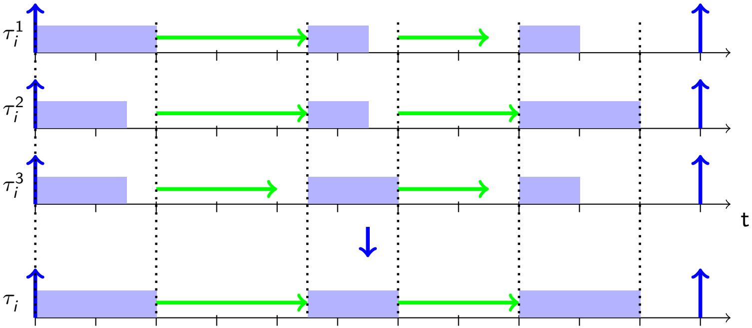
⇒ Hybrid models with different flexibility / accuracy tradeoffs

Hybrid Self-Suspension Models



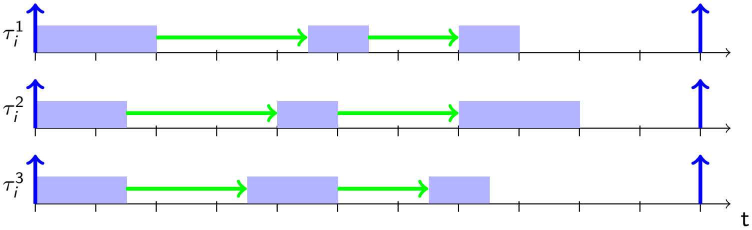
- Task: set of execution / suspension patterns
- Hybrid Models
  - Given number of suspension intervals
  - more information ⇒ more precision

Hybrid Self-Suspension Models



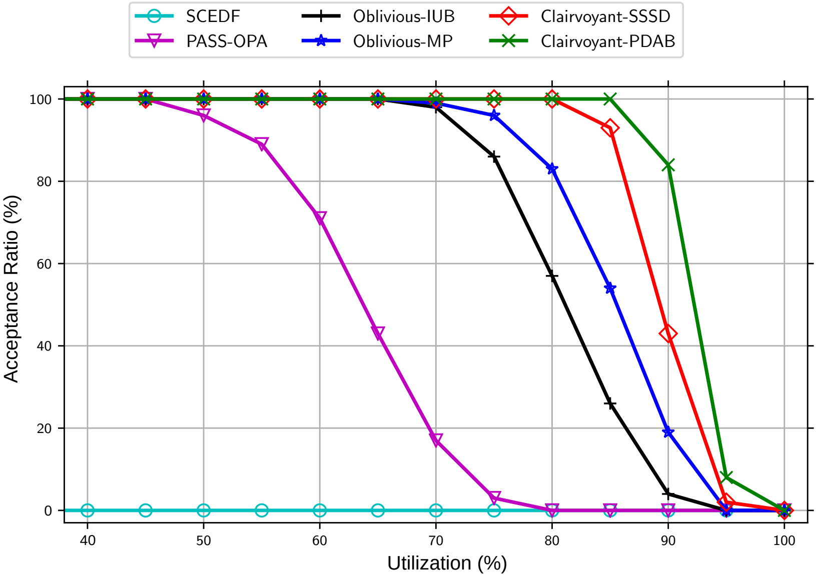
Model		Information
Dynamic		maximum suspension time
Hybrid	Pattern Obli. Upper Bounds	pattern upper bounds
Segmented		one precise pattern

Hybrid Self-Suspension Models

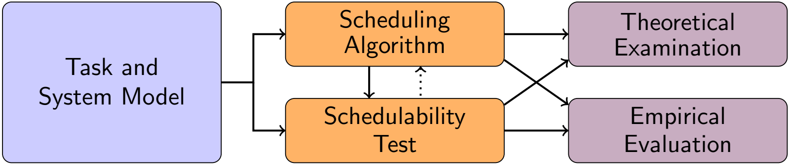


<table>
<tr><th colspan="2">Model</th><th>Information</th></tr>
<tr><td colspan="2">Dynamic</td><td>maximum suspension time</td></tr>
<tr><td rowspan="3">Hybrid</td><td>Pattern Obli. Upper Bounds</td><td>pattern upper bounds</td></tr>
<tr><td>Pattern Obli. Multiple Paths</td><td>precise patterns offline</td></tr>
<tr><td>Pattern Clairvoyant</td><td>precise patterns offline & online</td></tr>
<tr><td colspan="2">Segmented</td><td>one precise pattern</td></tr>
</table>

Evaluation - Moderate Suspension in [0.1,0.3]



Conclusion



Hypothesis

Realistic scheduling models and analyses are essential for guaranteeing timing correctness in advanced real-time systems while ensuring that the system resources necessary to provide these guarantees are not over-provisioned.

Hybrid Self-Suspension Models

Shortest Execution Interval First Deadline Assignment

Parametric Augmentation Functions