



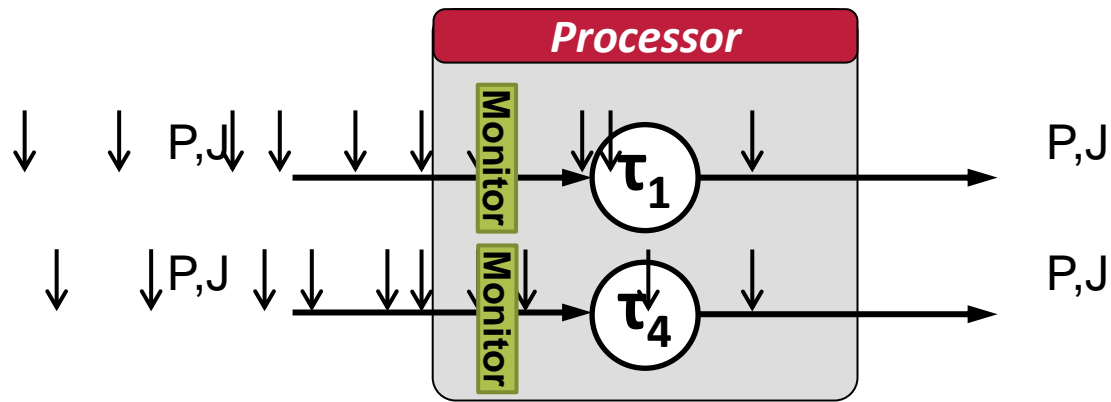
Technische  
Universität  
Braunschweig



## Monitoring of Complex Activation Patterns in Real-Time Systems

Moritz Neukirchner, *Tobias Michaels*, Philip Axer, Sophie Quinton,  
Rolf Ernst

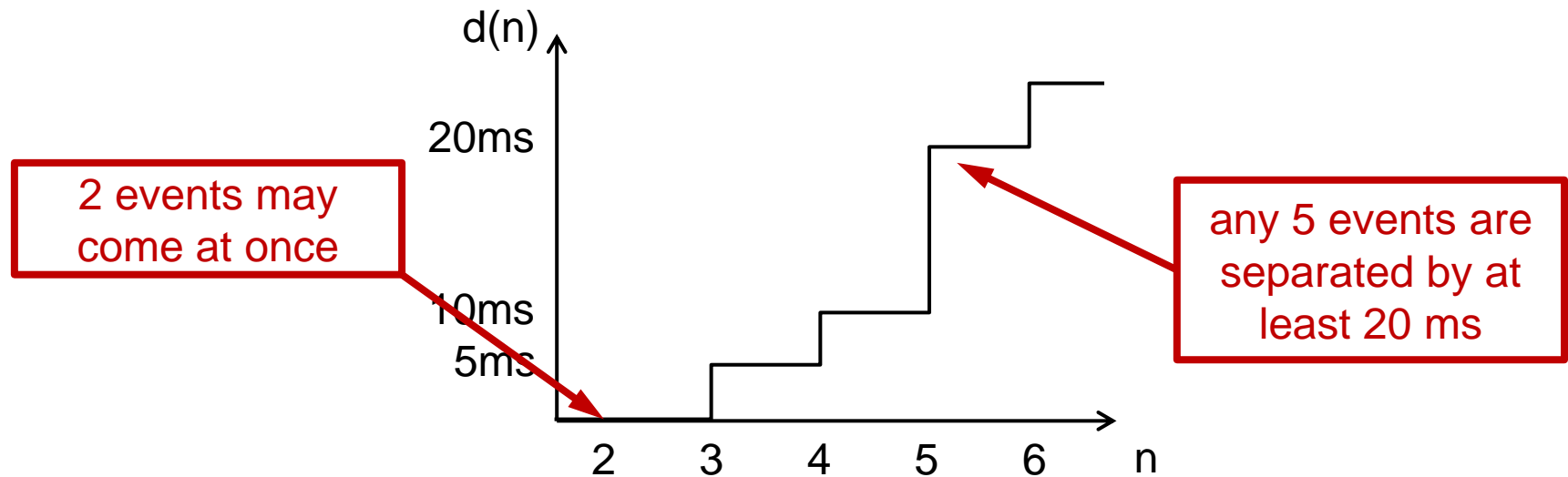
# Task Activations and Their Modelling



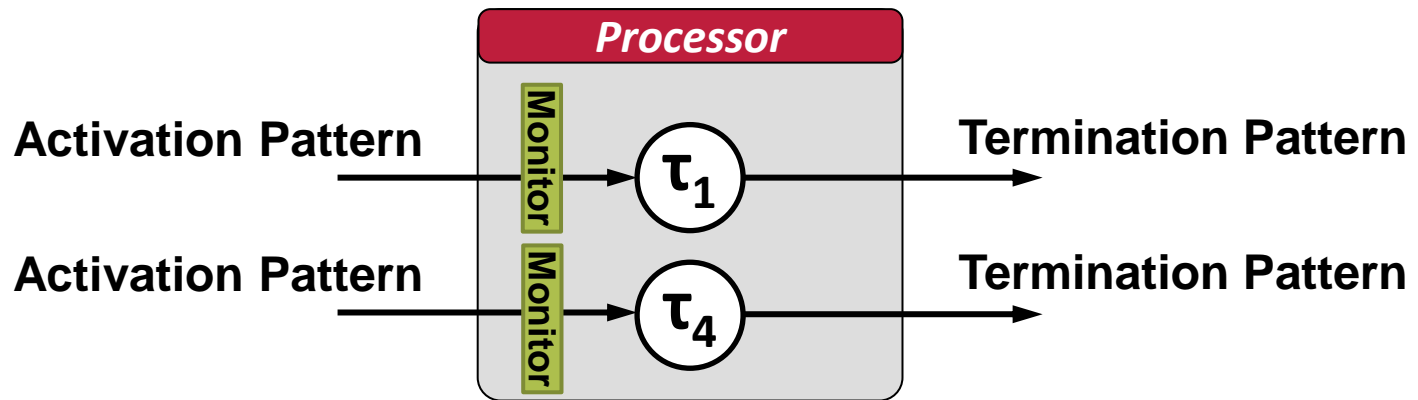
- **Specific** task **activations/terminations** are captured in **traces**
- **Monitoring** determines whether **trace adheres** to description of **verification**
- For analysis activations/terminations are **abstracted** in **event models**
- Common event models, e.g.
  - Periodic
  - Periodic with Jitter

# Complex Activation Patterns

- **Variety** of **activation patterns** used in practice  
*e.g. periodic + spontaneous, dual cyclic, on change*
- Timing **verification** can consider them through use of **minimum distance functions**  
*i.e. specification of the minimum distance between any  $n$  consecutive events*



# The Monitoring Gap



- **Monitoring** is typically performed with

- watchdog timers
- heartbeat signals
- timed monitoring tasks

→ **Implies periodic activation**

→ **Currently no monitoring of complex activation patterns**

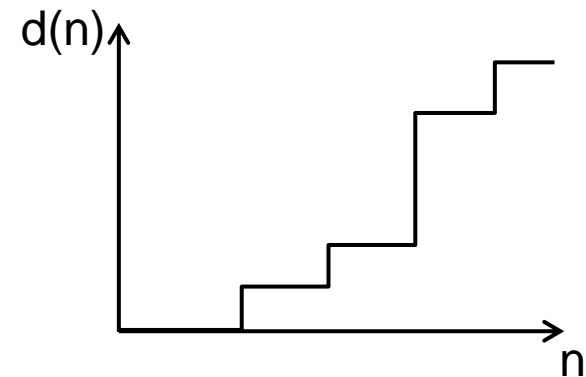
# The Monitoring Gap

**Why is monitoring currently not performed according to complex event models?**

*Recall:*

*Specification of the minimum distance between any  $n$  consecutive events*

- Minimum distance functions have **infinite domain**
- Checking activation patterns requires to **store** and **check entire trace**
  - Monitoring **overhead increases with system uptime**
  - Monitoring **overhead is unbounded**



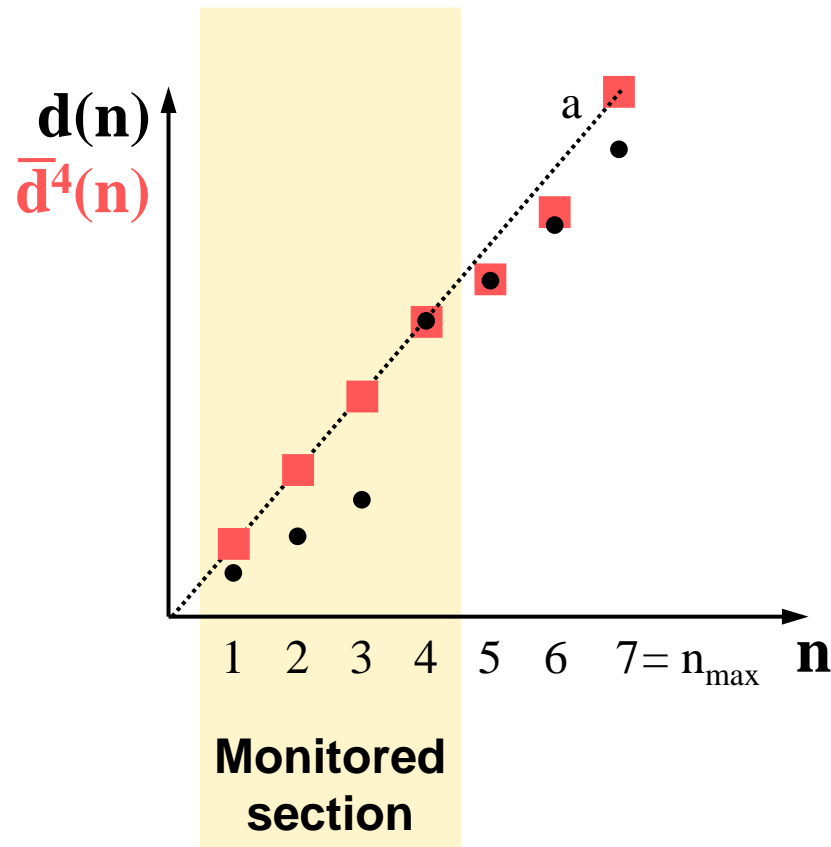
# Closing the Monitoring Gap

We can show that

- for a **certain class** of minimum distance functions („I-repetitive“ ) it is **sufficient to log only I events** and perform **I comparisons**  
*I-repetitive functions are relevant in practice, e.g. periodic bursts*
- one can **construct such** an I-repetitive minimum distance **function** from an arbitrary minimum distance function to monitor conservatively  
*i.e. monitoring may yield false positives but overhead is limited*
- **conservatism** can be **explicitly calculated**

**We can construct a monitor with user-specified constant overhead (runtime & memory) for arbitrary activation patterns.**

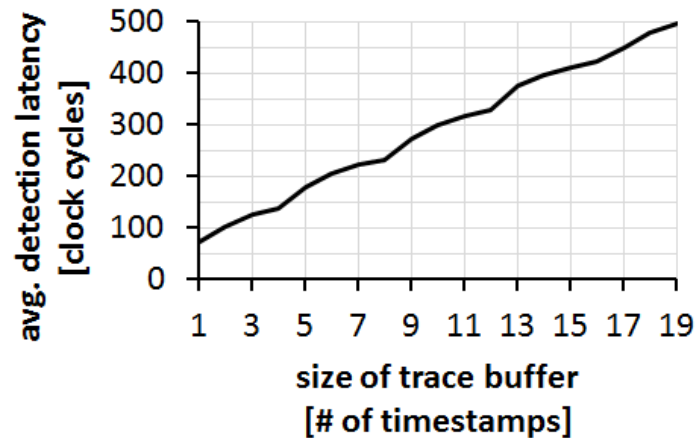
# Construction of Conservative Monitor



- Provide **relevant interval** of minimum distance function *obtained through WCRT analysis*
- Determine **maximum tangent**
- **Decrease** explicitly defined points **as far as possible**
- All **other** points **implicitly defined** through l-repetitiveness
- **Only** required to **monitor** explicitly **defined section**

# Overhead

- Prototype implemented on ARM Cortex-M3
- Trace buffer requires  $l$  elements for  $l$ -repetitive function
- **Runtime:**



- **Memory:**

$20+l*8$  byte

(4 bytes for tracing / 4 bytes for min dist funct.)



# Conclusion

- **Current monitoring** schemes support **limited activation patterns**  
*e.g. periodic activations*
- **Current verification** algorithms **utilize arbitrary activation patterns**  
*i.e. minimum distance functions*
- **To ensure verified behavior** in-system, **monitoring of arbitrary**  
activation patterns required

We can provide

- **Monitoring of arbitrary activation patterns**
- Monitoring at **designer-specifiable constant overhead**  
*i.e. trade-off between monitoring accuracy and runtime & memory*