OSEK-Like Kernel Support for Engine Control Applications Under EDF Scheduling

Vincenzo Apuzzo, **Alessandro Biondi** and Giorgio Buttazzo

ReTiS Laboratory

Scuola Superiore Sant'Anna, Pisa, Italy

Towards the use of EDF in real-world engine control applications

Not only periodic tasks!
Engine control applications also include adaptive variable-rate tasks

Benefits in terms of schedulability have been observed (in theory) under **EDF** scheduling

This Work

- **OSEK-like** RTOS support for **EDF** scheduling of engine control applications
- Simulation Framework

INSTITUTE OF COMMUNICATION, INFORMATION AND PERCEPTION TECHNOLOGIES

Scuola Superiore Sant'Anna

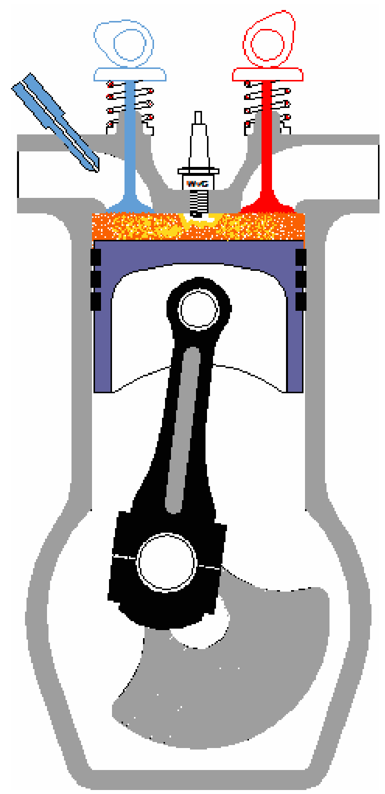
Retis
Real-Time Systems Laboratory

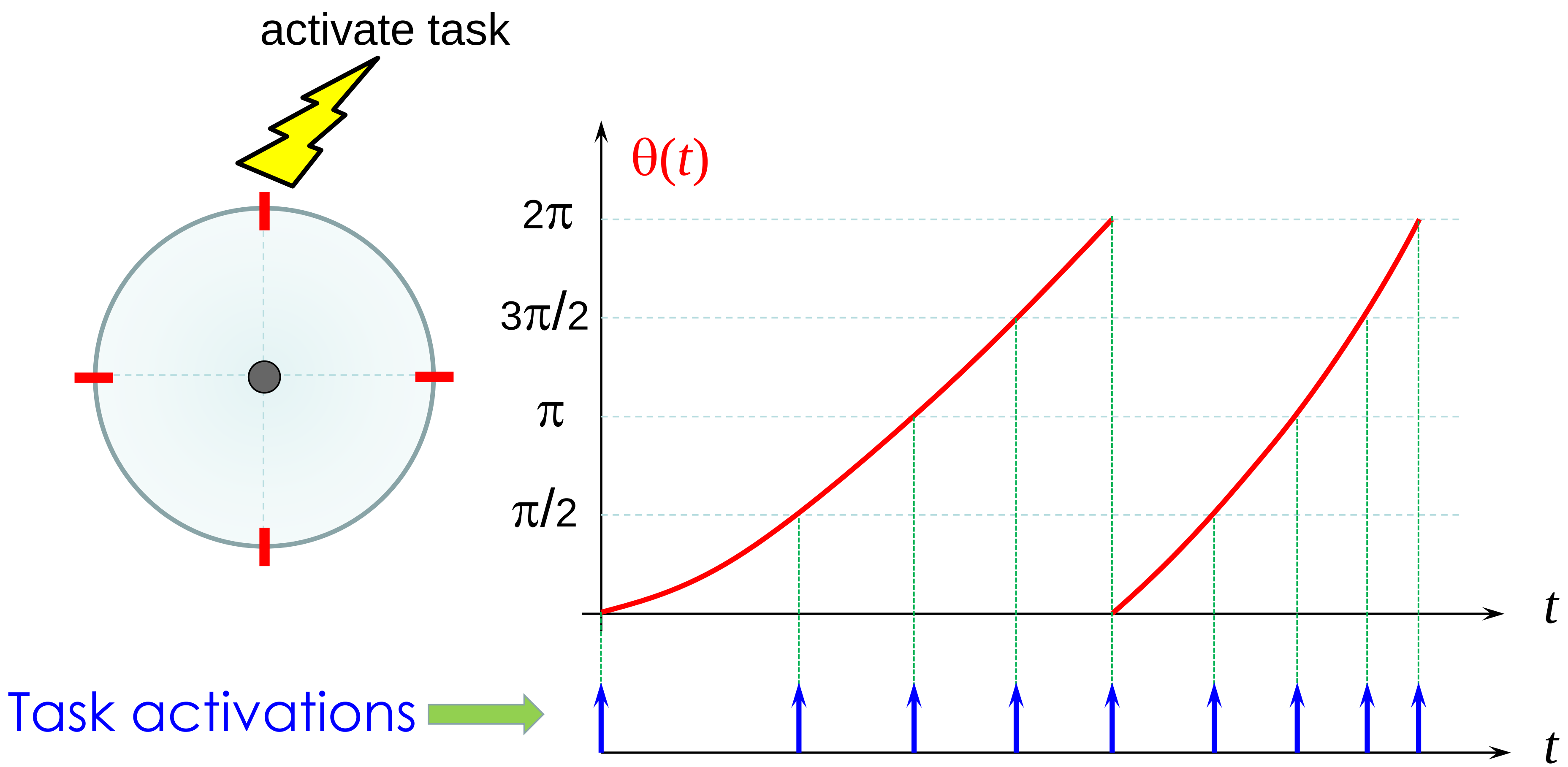
A LOOK INTO ENGINE CONTROL APPLICATIONS

Introduction

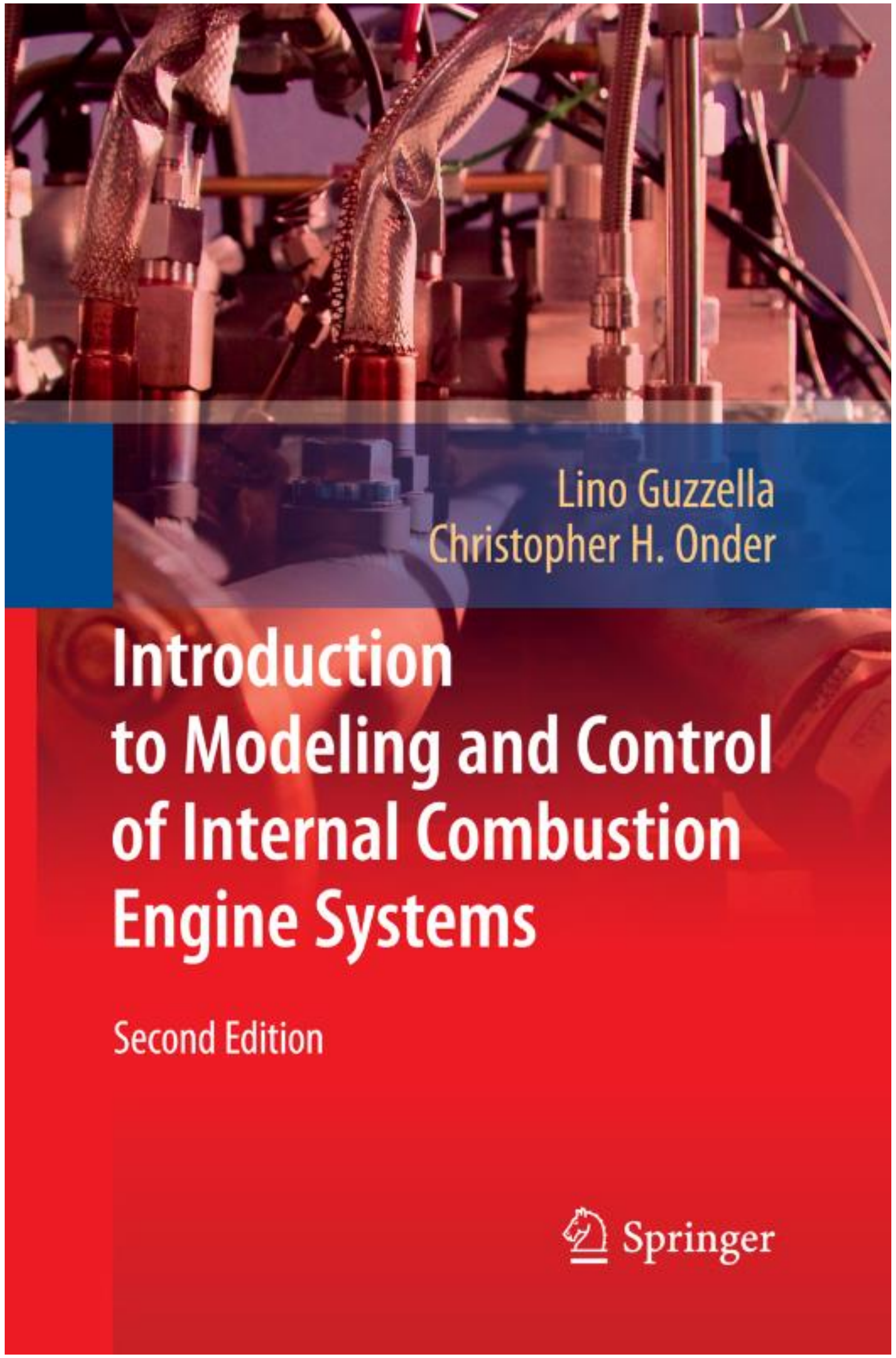
Engine control applications include

- **Periodic tasks** with fixed periods: 1 - 500 ms
- **Angular tasks**, linked to the rotation of the crankshaft



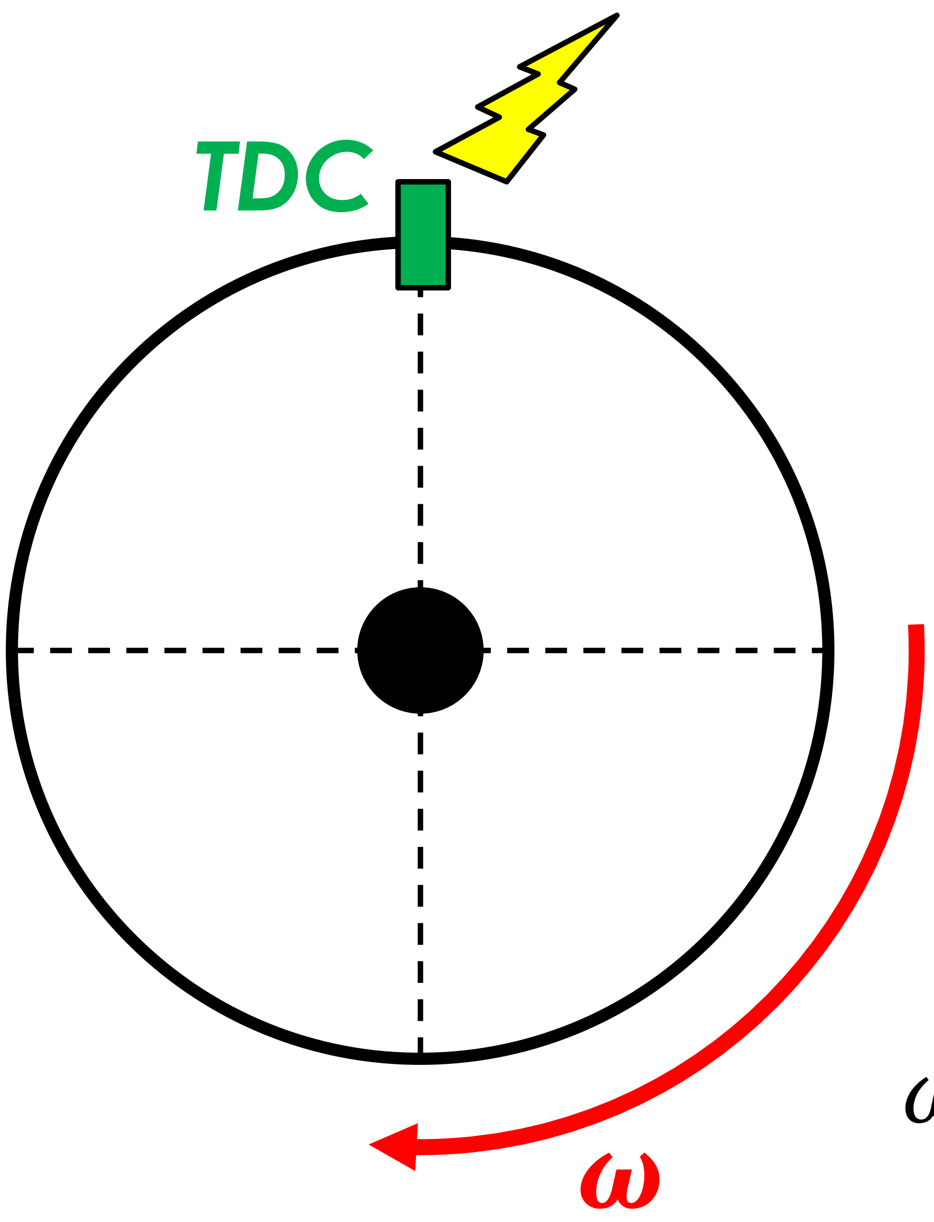


Introduction



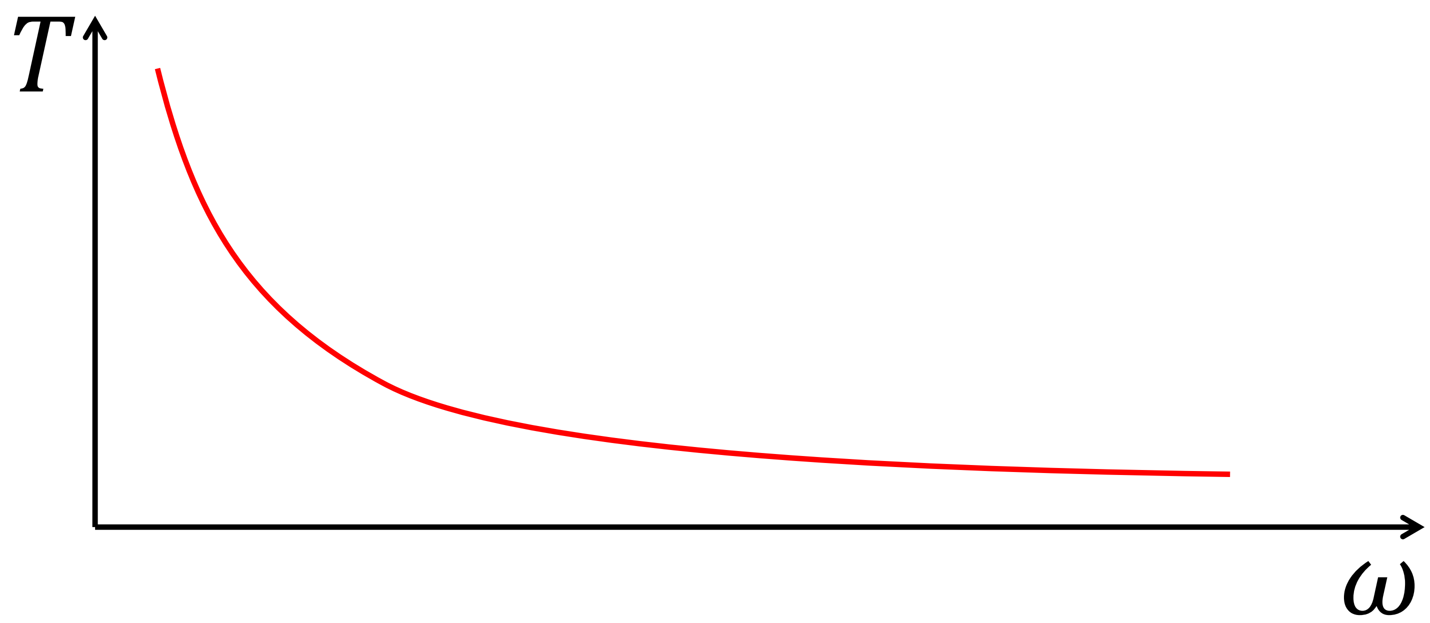
Engine-triggered Tasks

- **Engine-triggered** tasks – *single activation per revolution*



Inter-arrival time given a fixed speed ω

$$T = \frac{2\pi}{\omega}$$



$\omega^{min} = 500$ rpm ➡ $T^{max} =$ **120 ms**

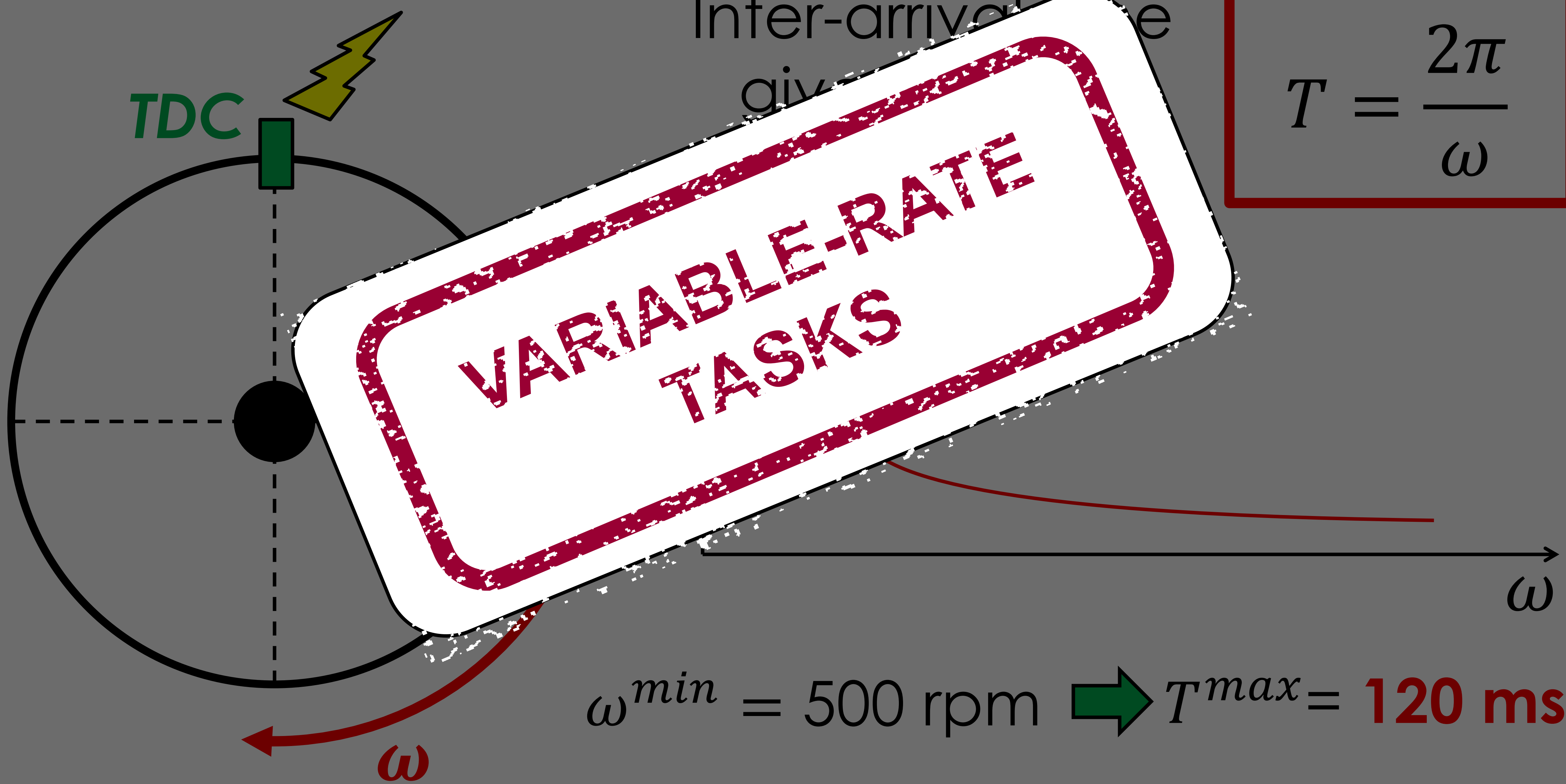
$\omega^{max} = 6500$ rpm ➡ $T^{min} \cong$ **10 ms**

Engine-triggered Tasks

- **Engine-triggered** tasks – *single activation per revolution*

Inter-arrival [illegible]e giv[illegible]

$$T = \frac{2\pi}{\omega}$$

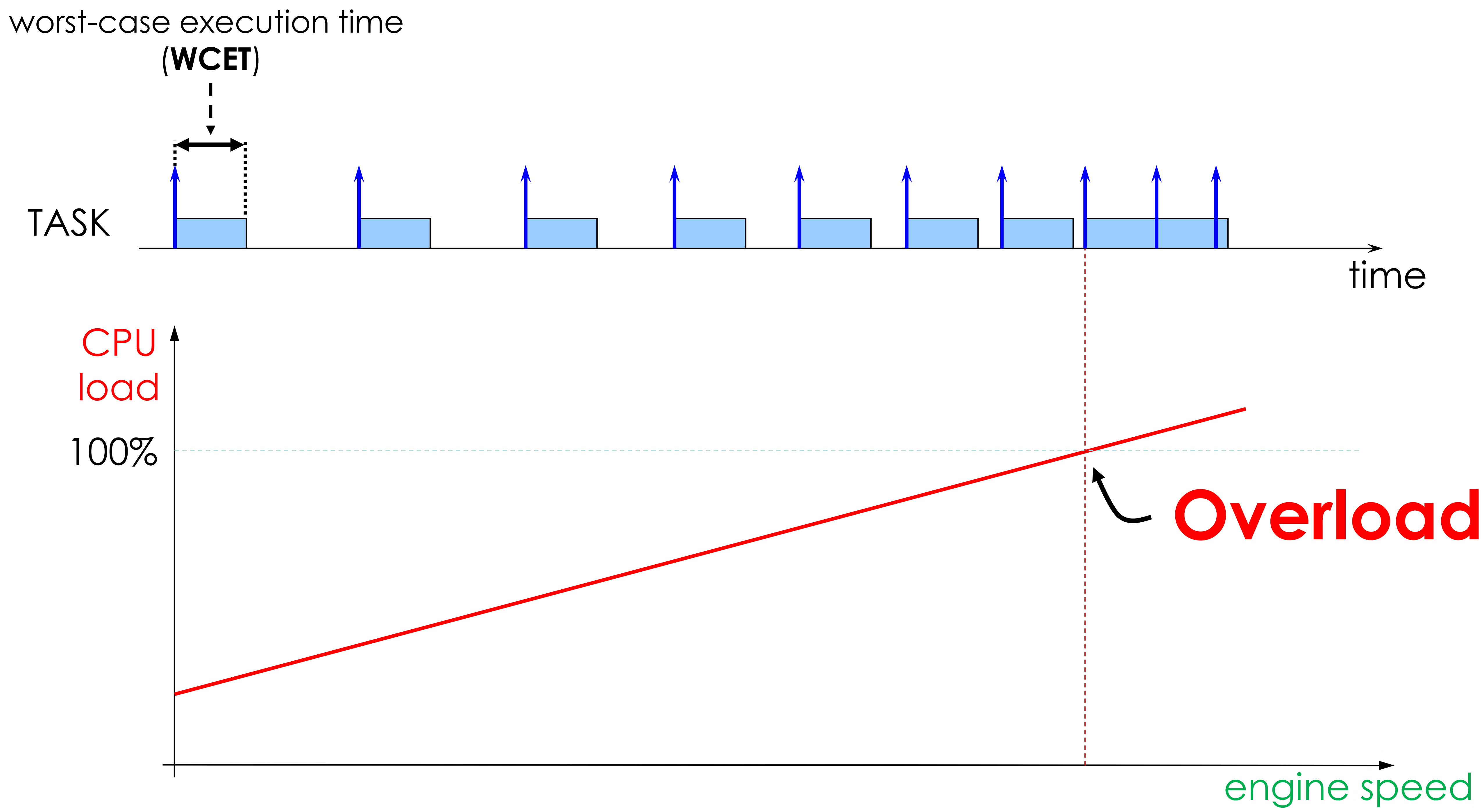


$\omega^{min} = 500$ rpm ➡ $T^{max} =$ **120 ms**

$\omega^{max} = 6500$ rpm ➡ $T^{min} \cong$ **10 ms**

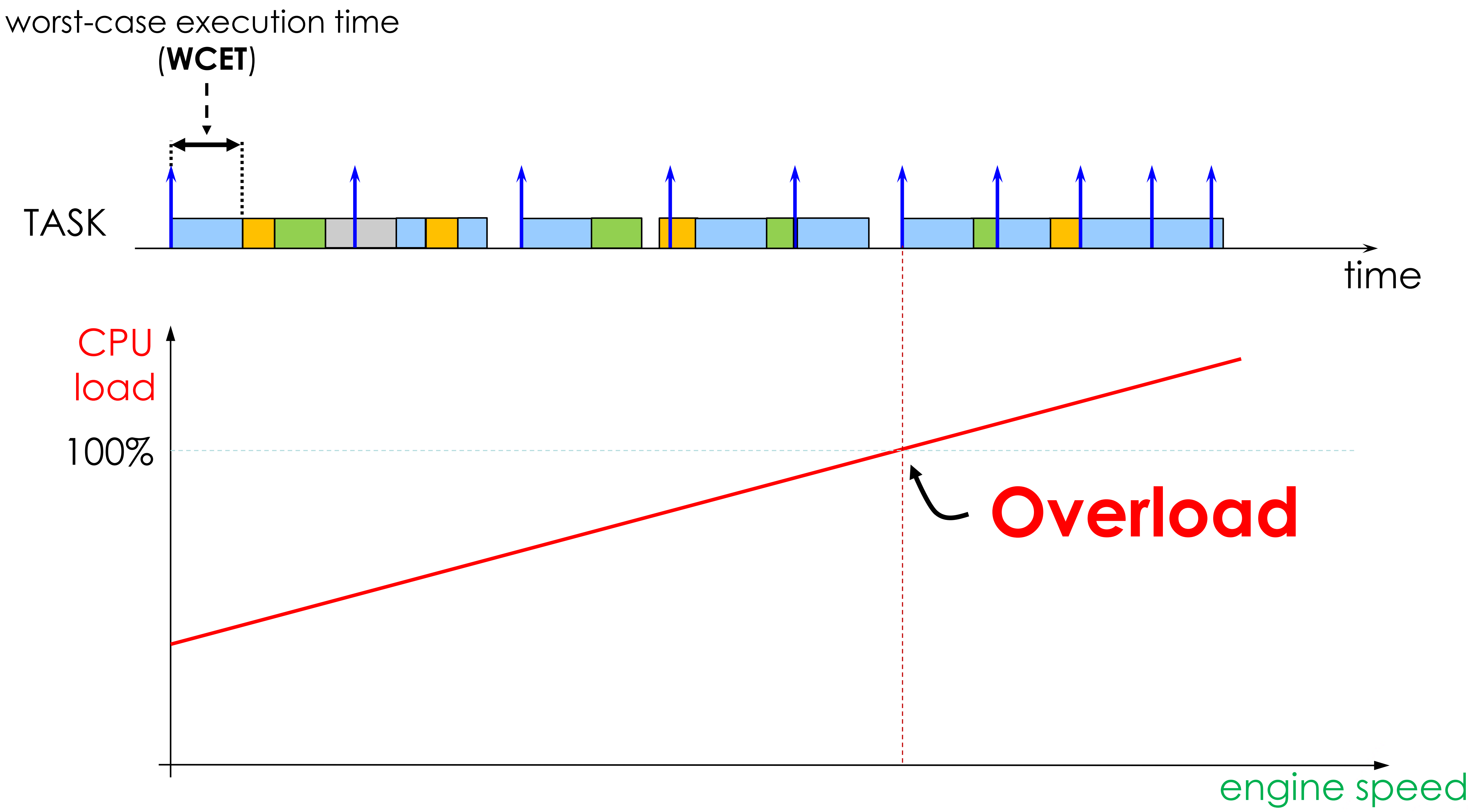
Engine-triggered Tasks

Suppose a fixed **WCET** for the task



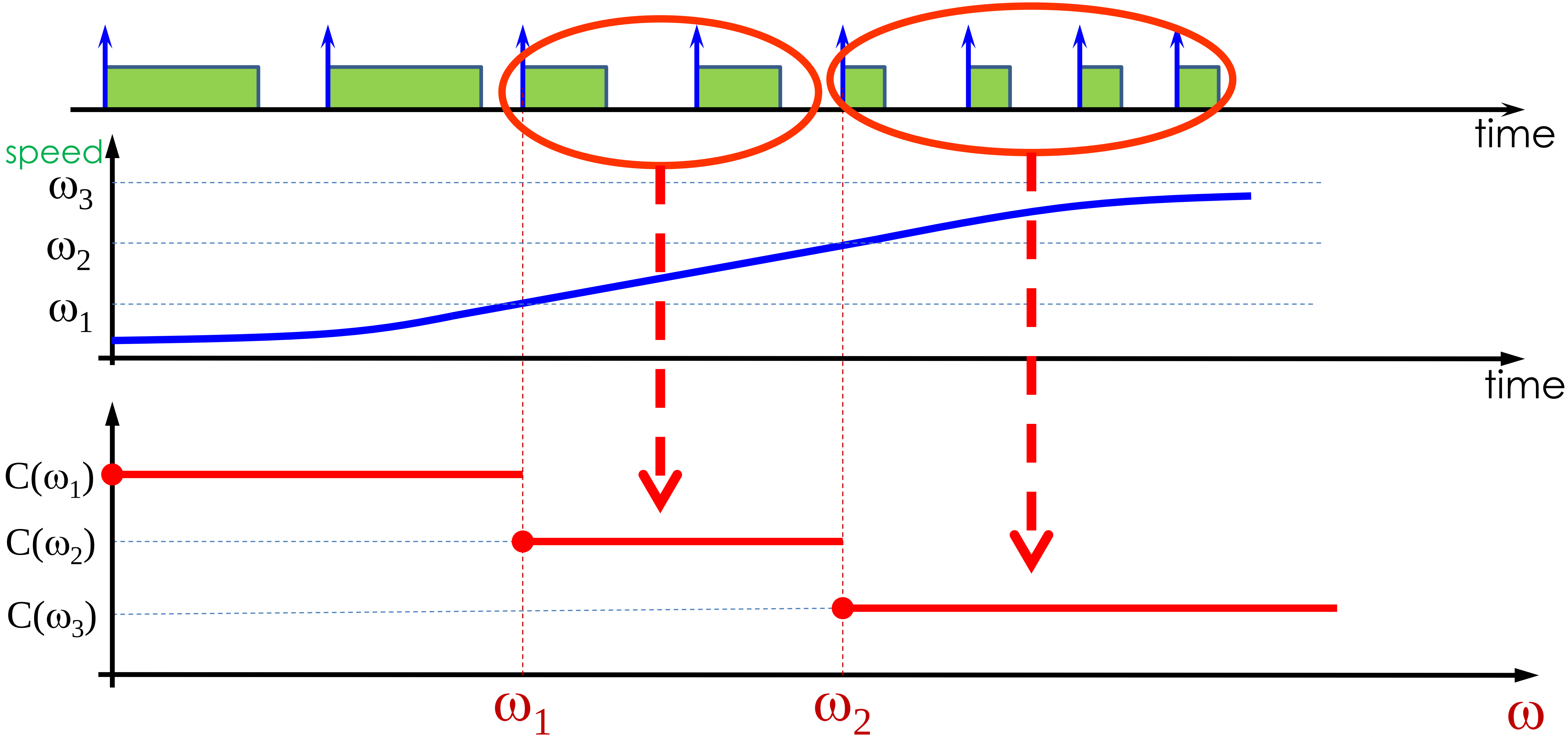
Engine-triggered Tasks

Suppose a fixed **WCET** for the task



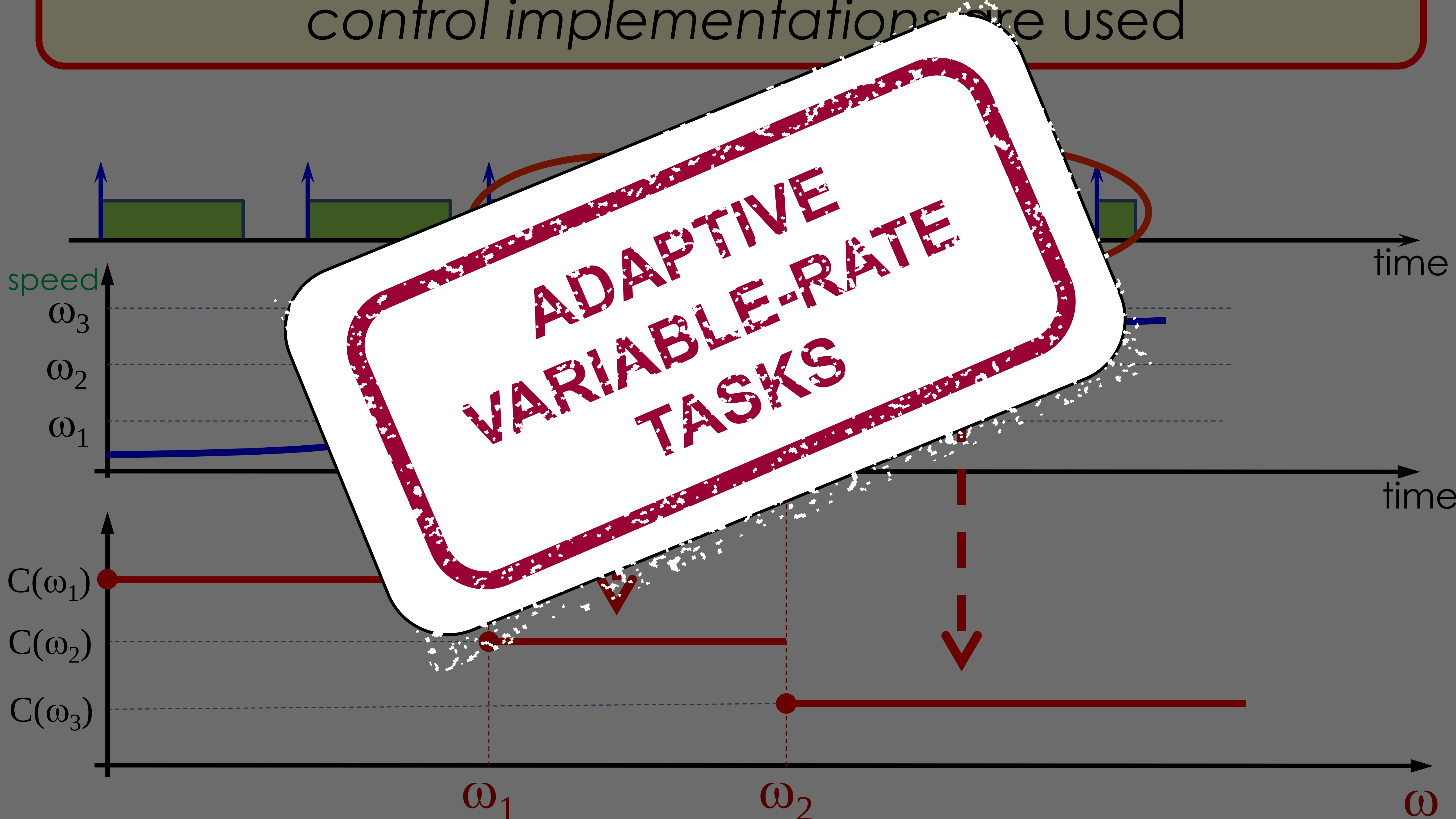
Engine-triggered Tasks

To prevent **overload** at high rates, *different control implementations* are used



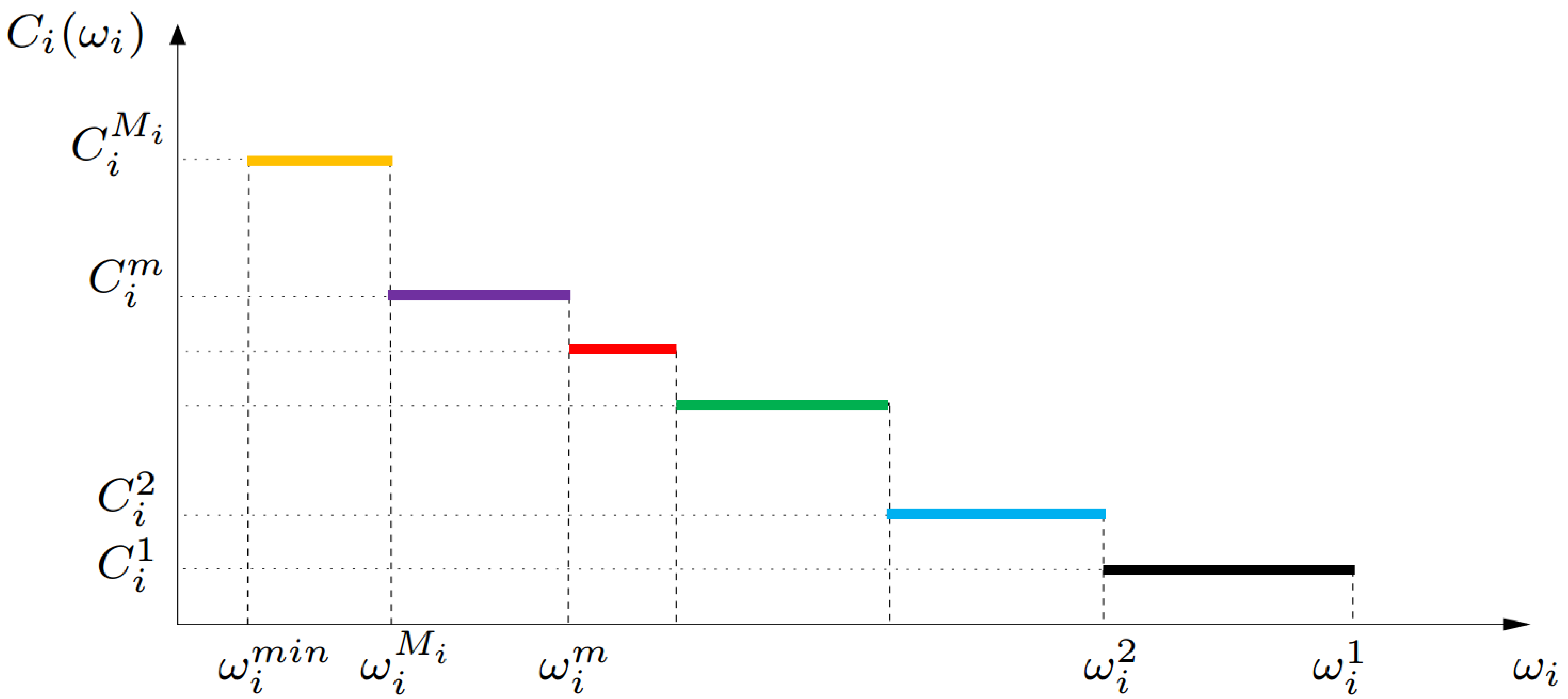
Engine-triggered Tasks

To prevent **overload** at high rates, *different control implementations* are used

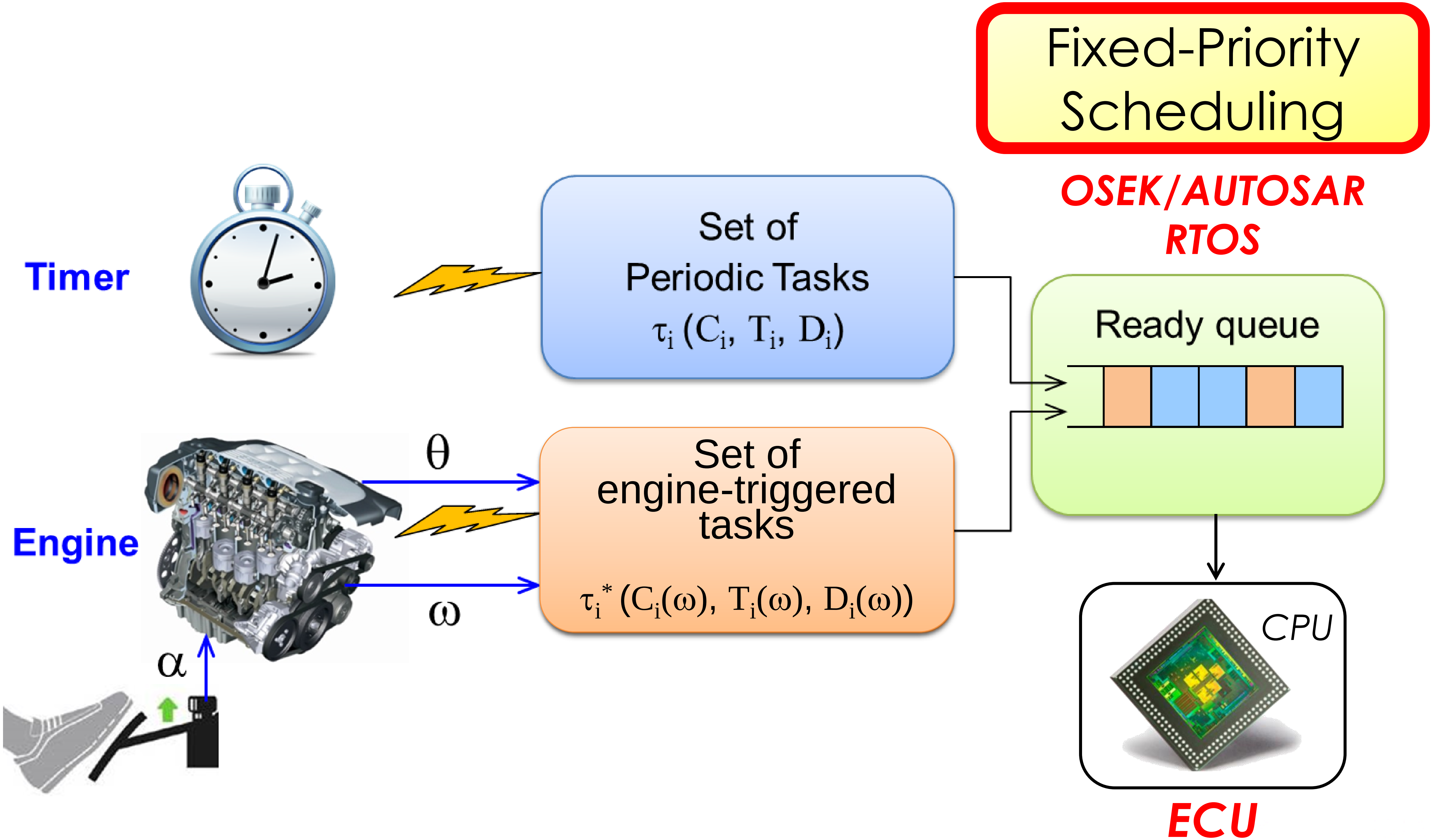


Adaptive Variable-Rate Tasks

- The AVR task implements a number of **execution modes**



Scheduling Infrastructure



INSTITUTE OF COMMUNICATION, INFORMATION AND PERCEPTION TECHNOLOGIES

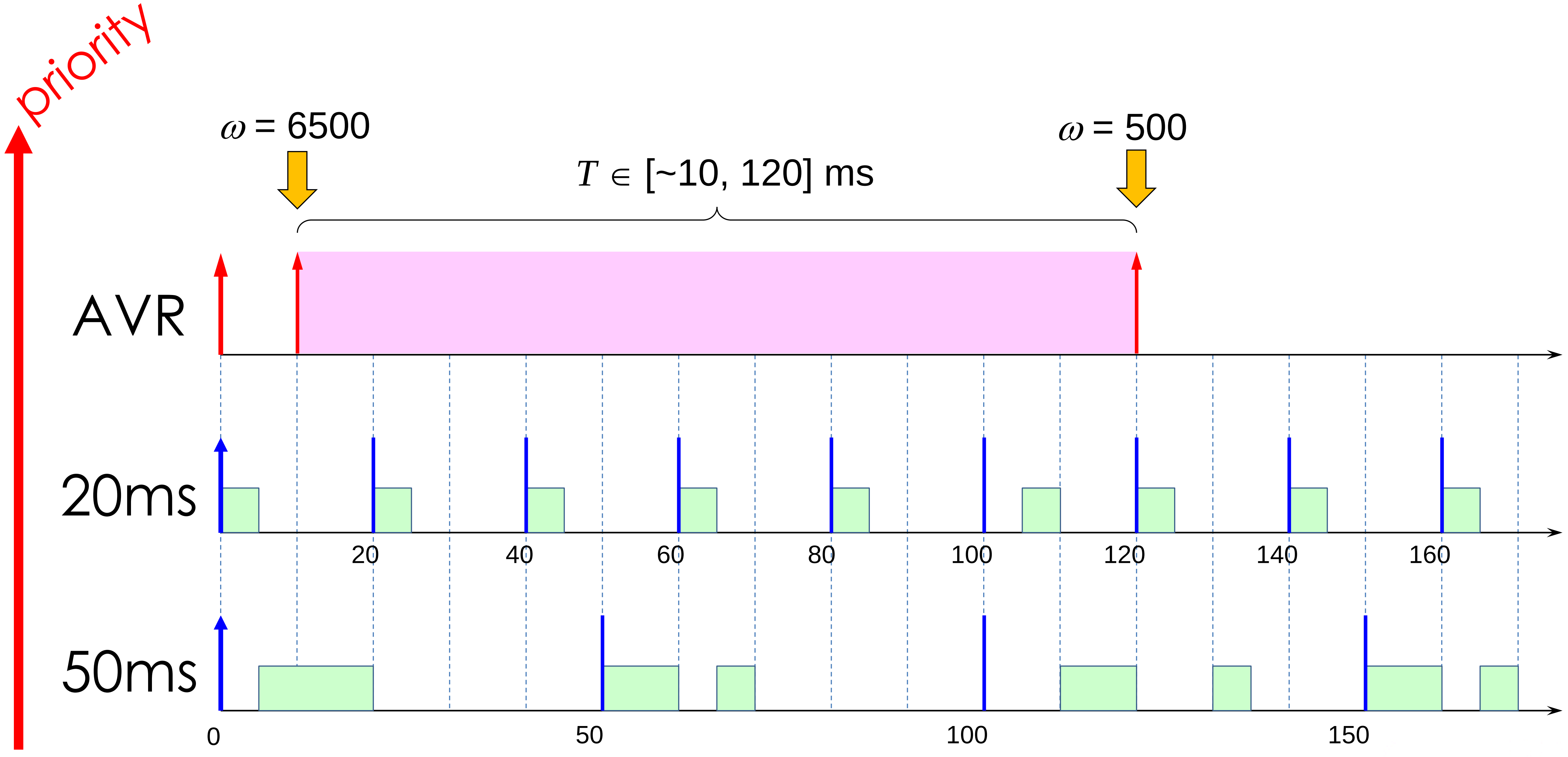
Scuola Superiore Sant'Anna

Retis Real-Time Systems Laboratory

IS FIXED-PRIORITY SCHEDULING THE BEST CHOICE FOR ENGINE CONTROL APPLICATIONS?

FP Scheduling of AVR Tasks

Since the inter-arrival time vary a lot with $\boldsymbol{\omega}$, any fixed priority assignment may **not be optimal** for some speed!

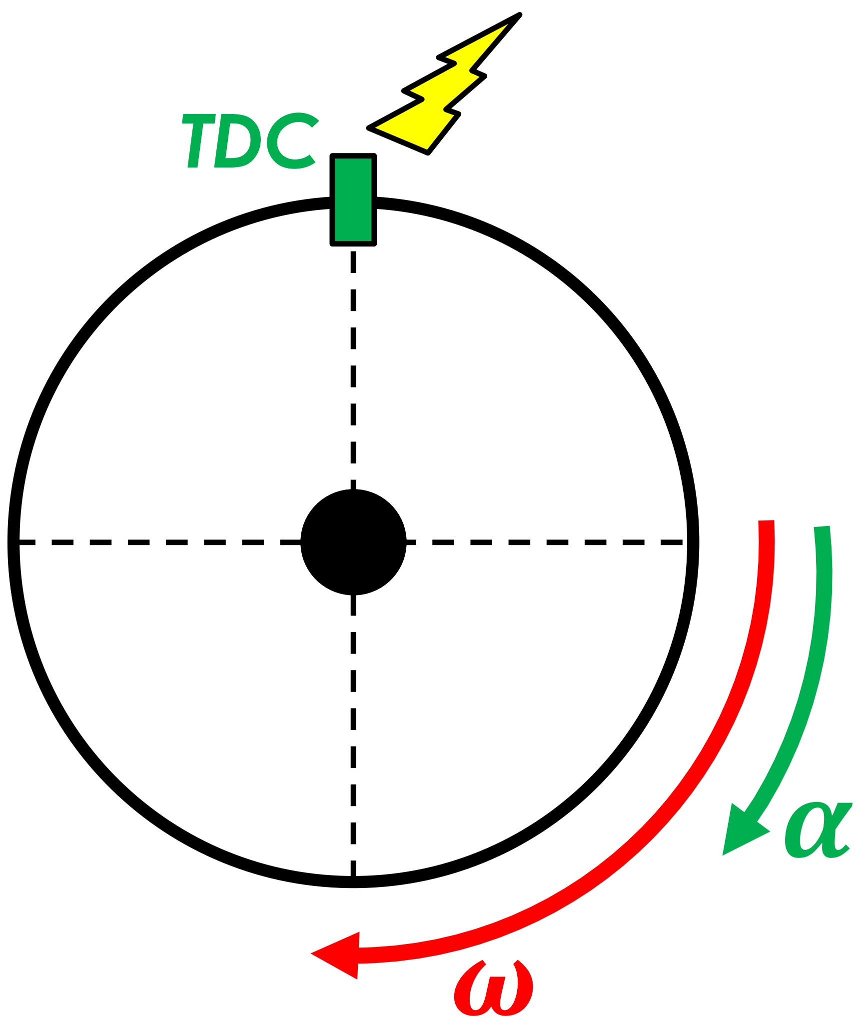


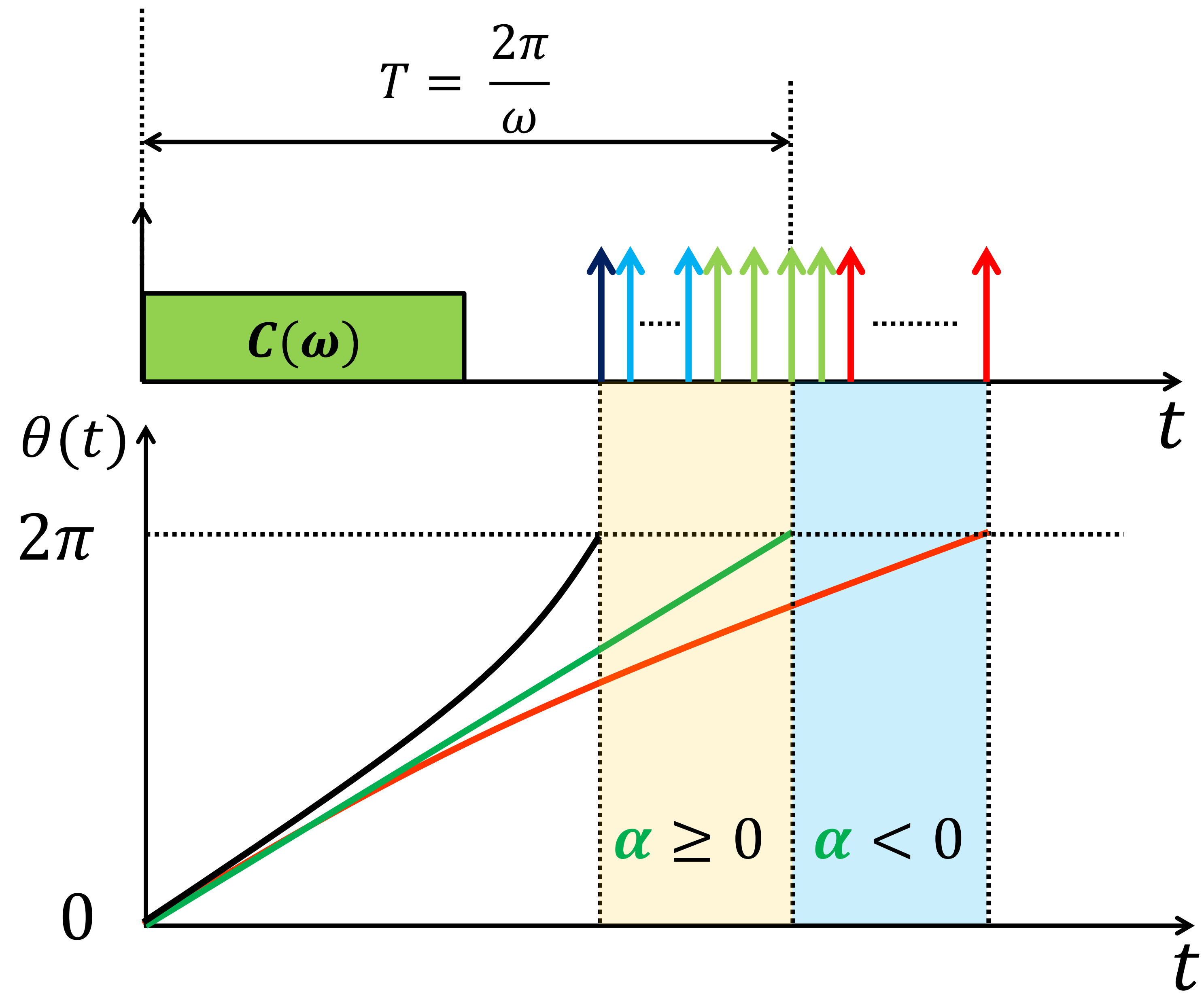
EDF Scheduling of AVR Tasks

- Job priorities are **adapted at run time** as a function of the engine speed at their release time
- Variable relative deadline for each job
- This is still EDF! (*job-level fixed-priority*)

Deadline Assignment

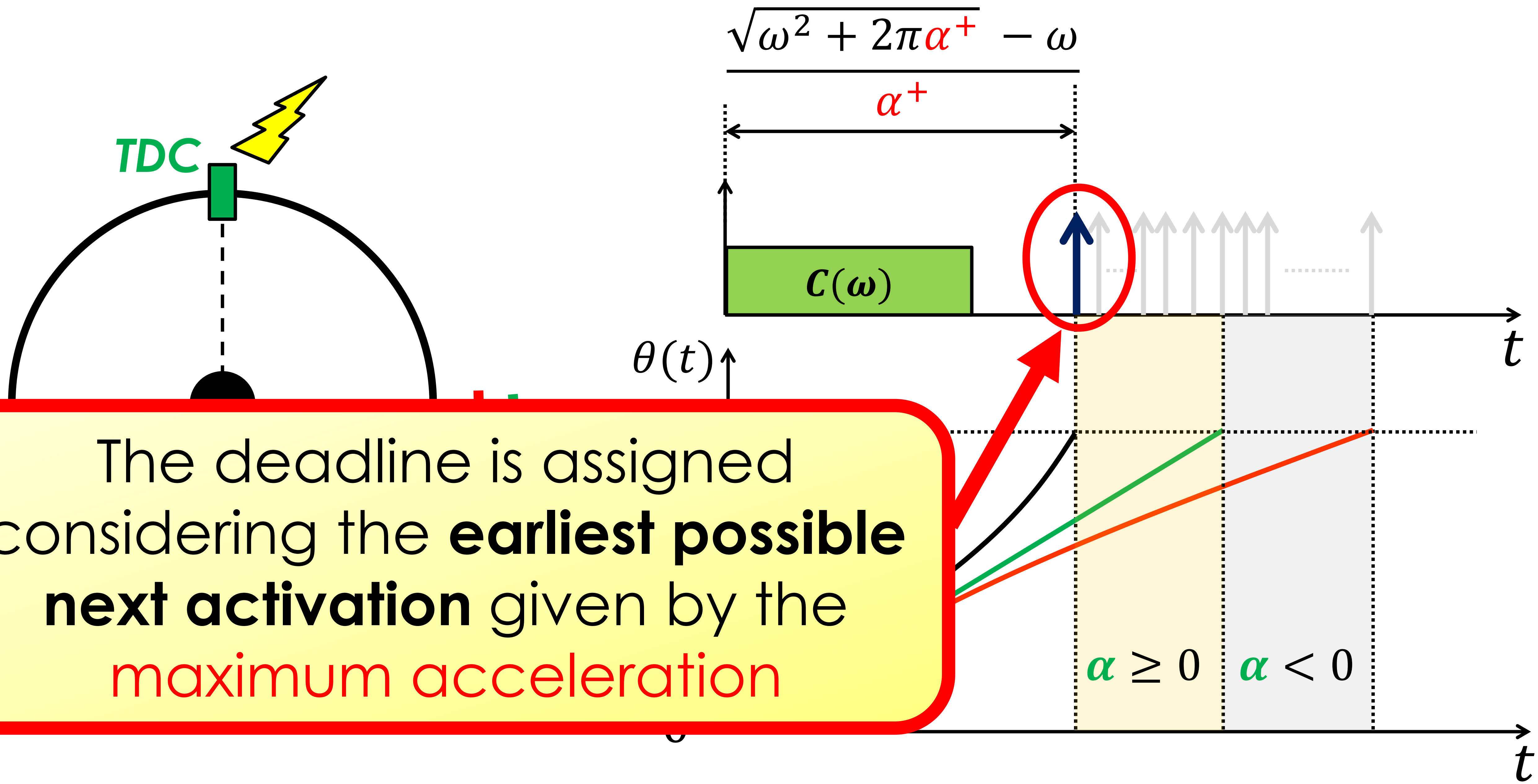
- **Engine-triggered** tasks – **Dynamic** condition





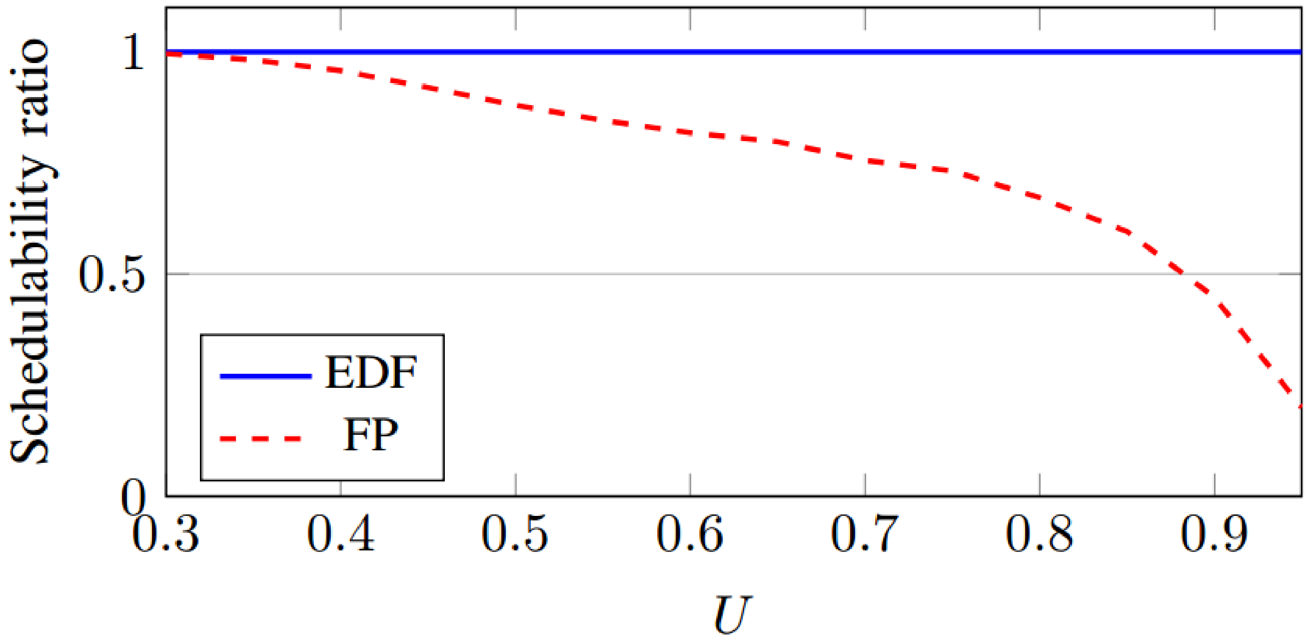
Deadline Assignment

- **Engine-triggered** tasks – **Dynamic** condition



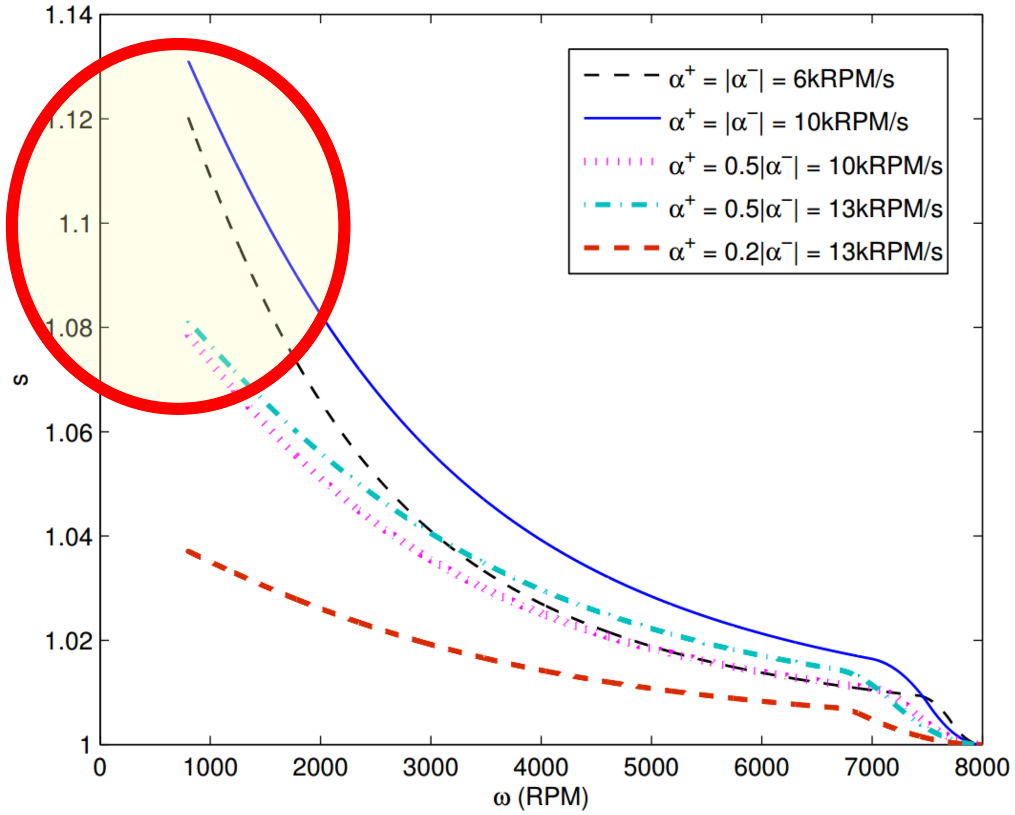
The deadline is assigned considering the **earliest possible next activation** given by the maximum acceleration

Benefits of EDF



Experimental results from [1]

EDF is "practically" optimal



Speed-up factor analysis Guo and Baruah [2]

Depends on engine speed and maximum acceleration

~1.1

[1] A. Biondi, G. Buttazzo, S. Simoncelli, "Feasbility Analysis of Engine Control Tasks under EDF Scheduling", ECRTS '15
[2] Z. Guo, S. Baruah, "Uniprocessor EDF Scheduling of AVR task systems", ICCPS '15

LET'S TRY TO USE EDF FOR REAL-WORLD ENGINE CONTROL APPLICATIONS…

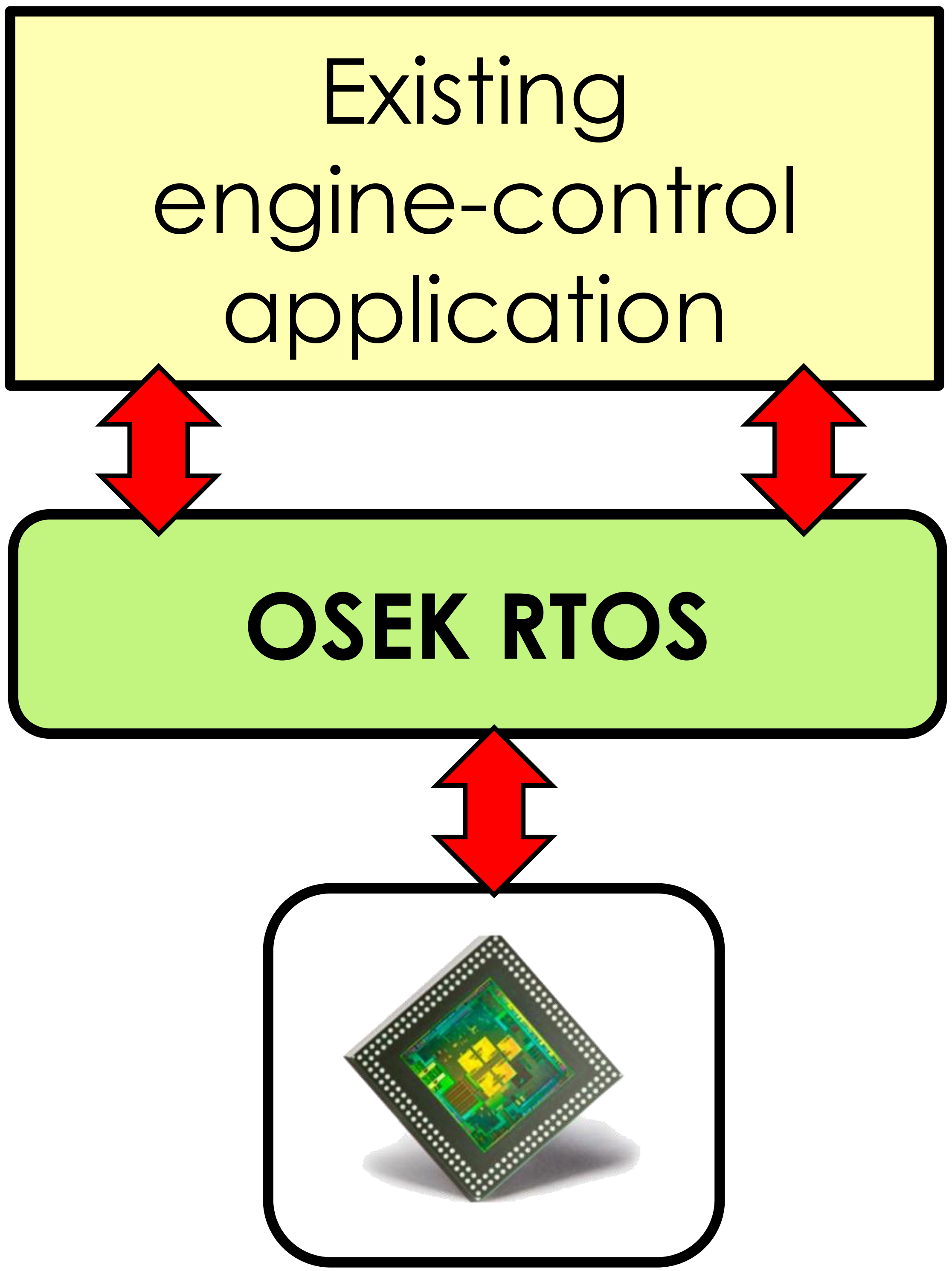
Our Goal

Motivated by the benefits of **EDF** observed in theory

- Design and implementation of a RTOS support for engine control applications under **EDF** scheduling
- Being **OSEK/AUTOSAR** the de-facto standard in the automotive industry

  - **Minimal changes** to the standard OSEK API
  - Integration with the OSEK standard configuration language (**OIL**)

Our Goal

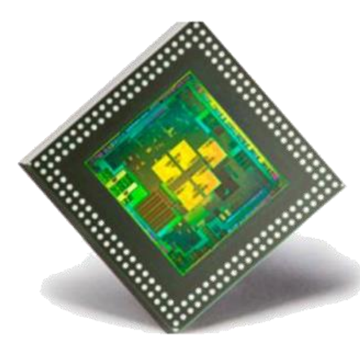


Our Goal

less changes as possible

Existing engine-control application

This Work



Erika Enterprise

ERIKA ENTERPRISE

- ERIKA Enterprise is an **OSEK/VDX** certified RTOS
- Offers a suitable open-source license allowing the static linking of closed source code
- Typical footprint around 2-4KB Flash
- Used by several automotive and white goods companies

Erika Enterprise

ERIKA ENTERPRISE

OSEK
certified
- BCC1
- BCC2
- ECC1
- ECC2

FP
minimal impl. of fixed-priority scheduling

EDF
EDF scheduling + (M)SRP

HR
two-level hierarchical scheduling (M)BROE

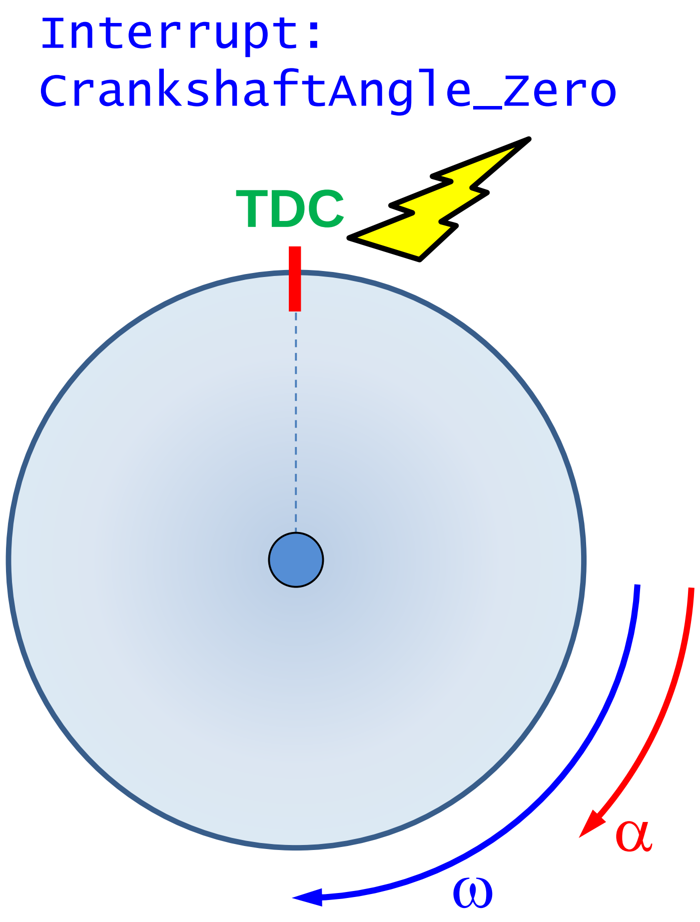
FRSH
EDF-based per-task resource reservation

OSEK-like API

Impact

- RTOS should be aware of the parameters of AVR tasks and the engine
- Needed support for variable relative-deadline as a function of the engine speed
- Needed extensions at the OSEK Configuration Language (OIL)
- Needed new support for deadline buffering to manage overloads
- Different requirements for stack sharing
- …

Activation of an AVR task




```
ISR(CrankshaftAngle_Zero) {

    ActivateTask(AVRtask);
}
```

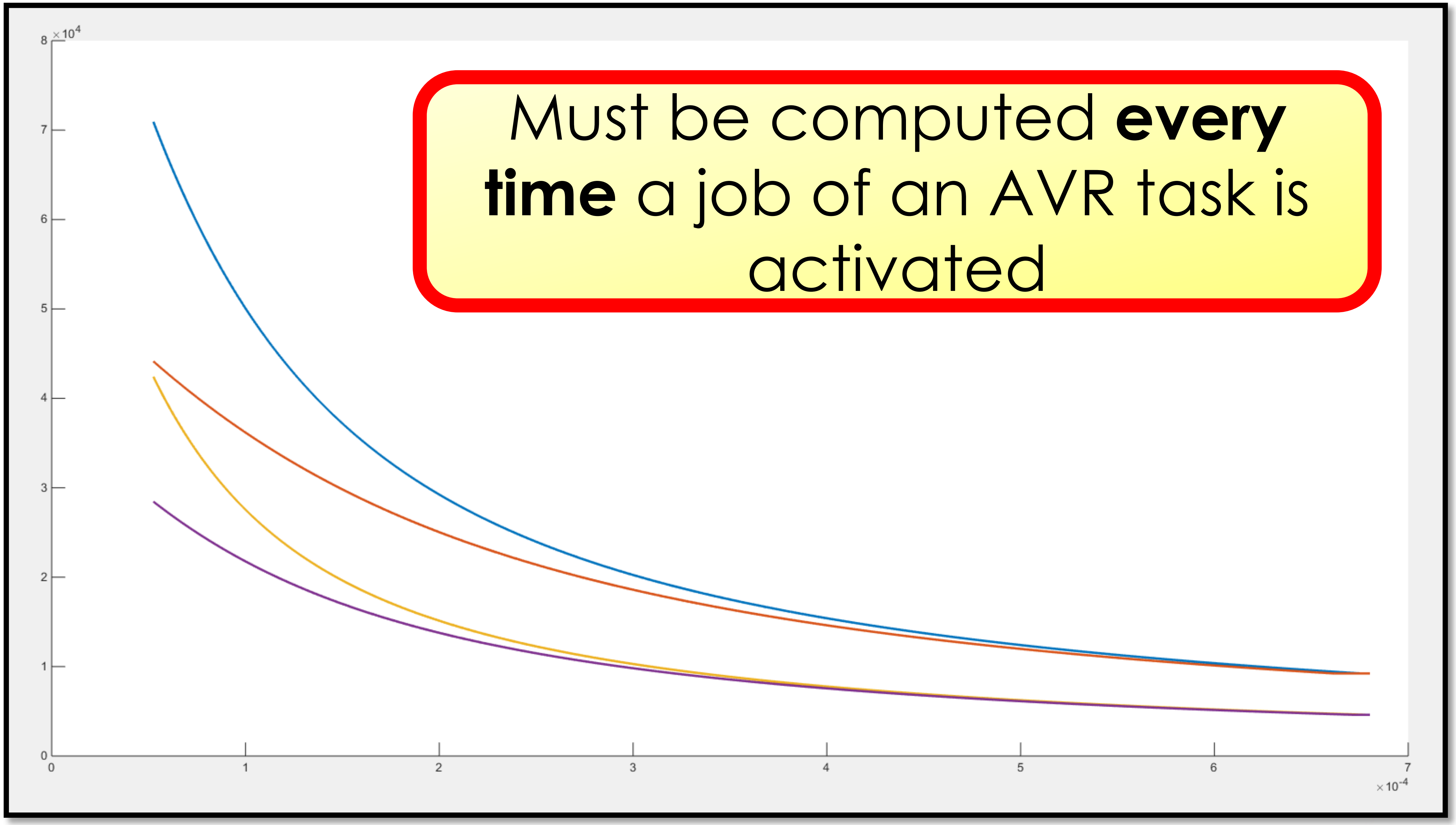
```
ISR(CrankshaftAngle_Zero) {

    ω = read_rotation_speed();

    ActivateTask(AVRtask, ω);
}
```

Not part of the
OSEK standard API

Deadline Computation

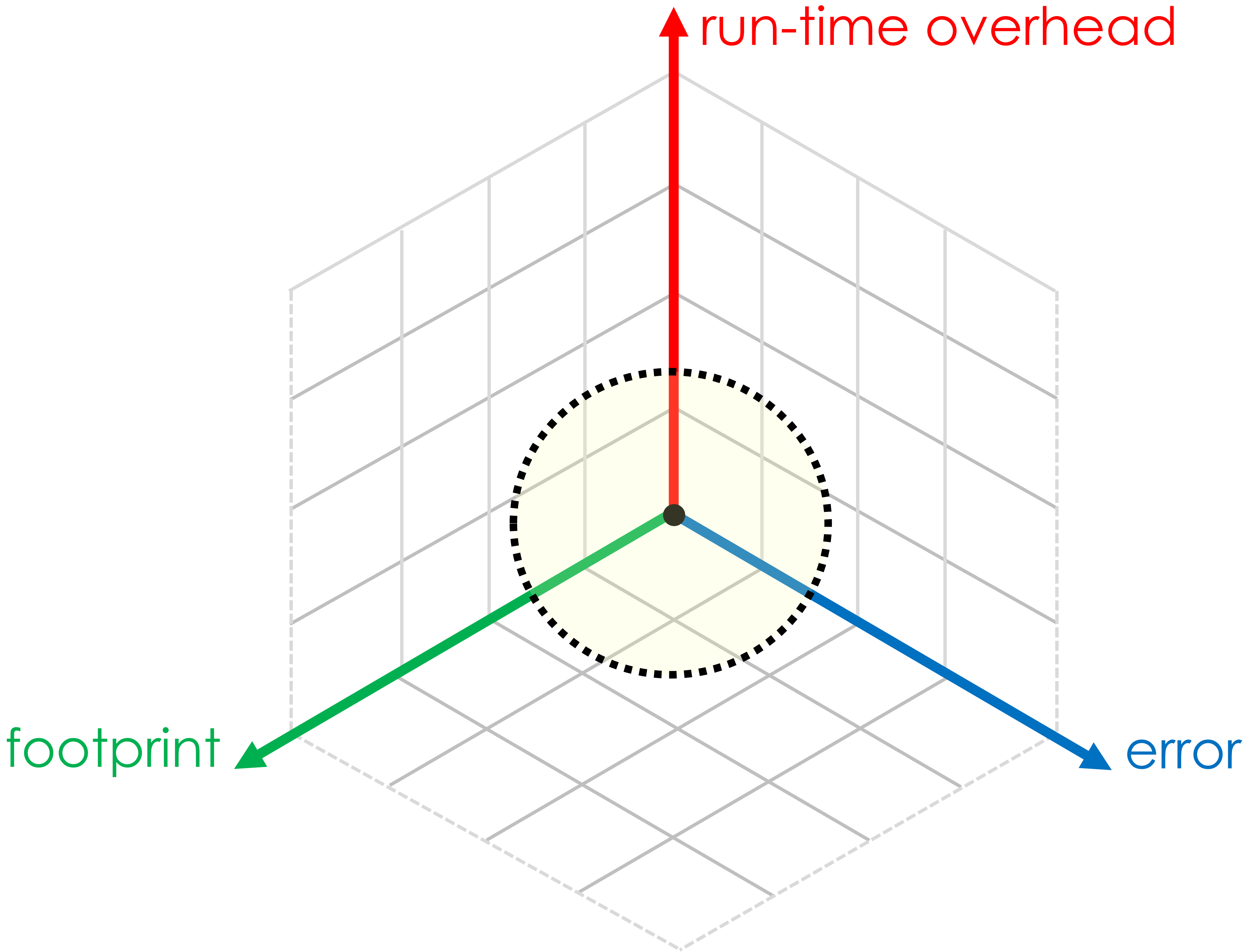
- The deadline of each job depends on the engine speed ω (at the *job release* time)

$D(\omega)$

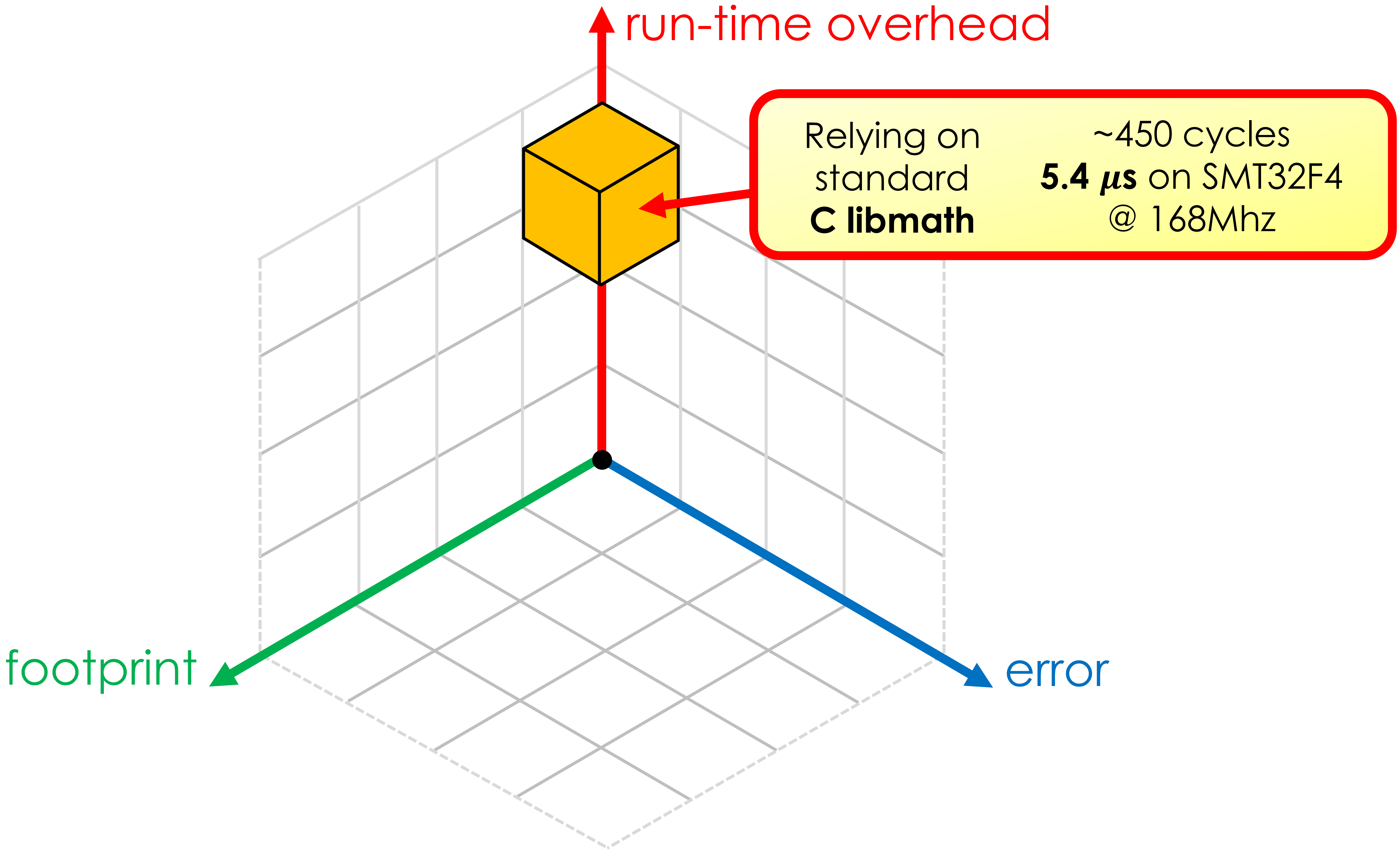


engine speed ω

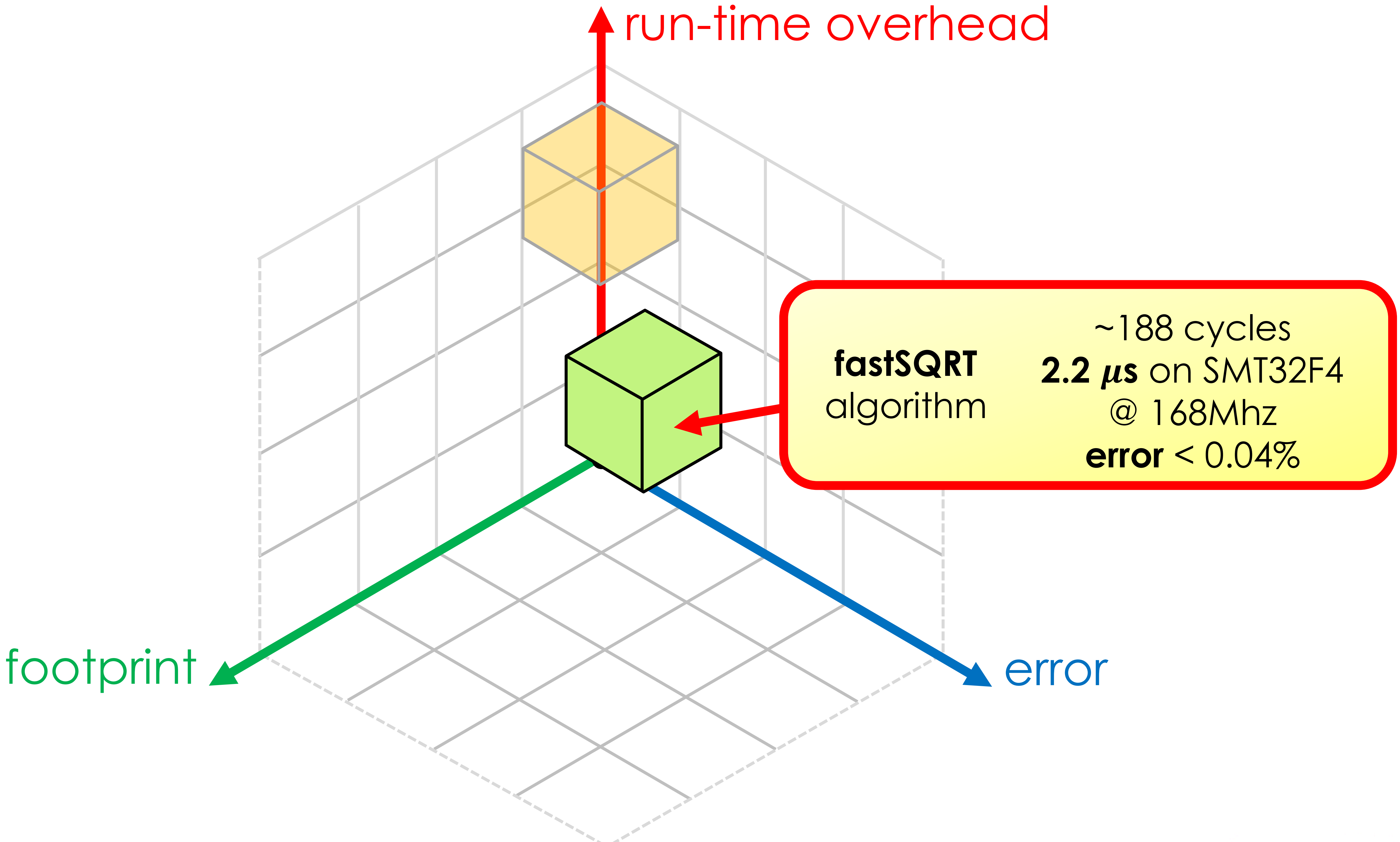
Deadline Computation



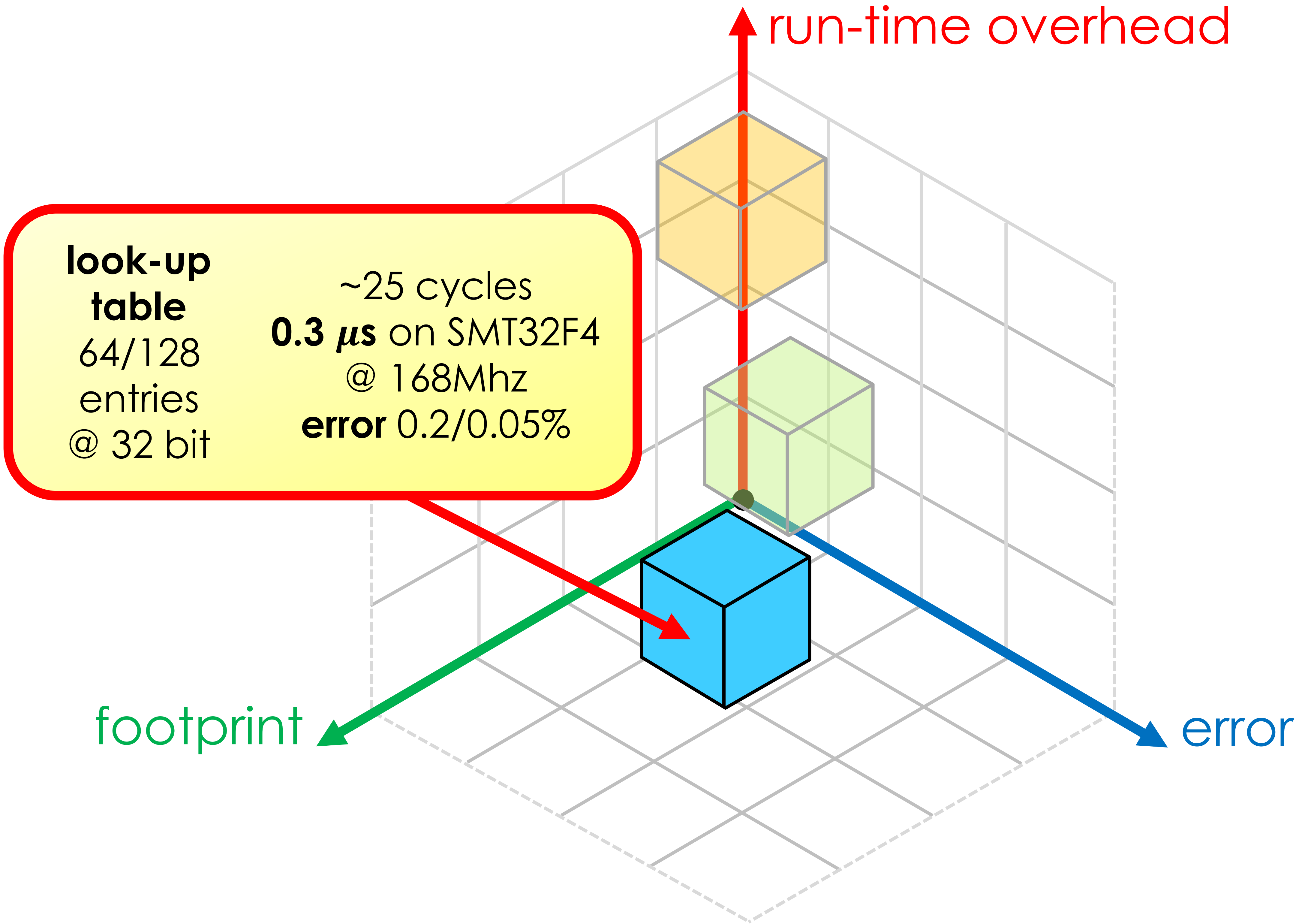
Deadline Computation



Deadline Computation



Deadline Computation



Experimental Results

Run-time overhead for the *ActivateTask*

context switch + deadline computation + ready queue management + …

Num. of Tasks		3	5	7	10
EDF-AVR (FastSQRT) MAX	μs	4.10	4.21	4.36	4.46
	cycles	689	707	732	749
EDF-AVR (FastSQRT) AVG	μs	4.0[illegible]	4.18	4.25	4.39
	cycles	762	[illegible]02	714	737
EDF-AVR (Lookup Table) MAX	μs	2.95	3.04	3.15	3.22
	cycles	495	510	529	541
EDF-AVR (Lookup Table) AVG	μs	2.91	3.01	3.14	3.20
	cycles	489	505	527	537

Fixed-priority

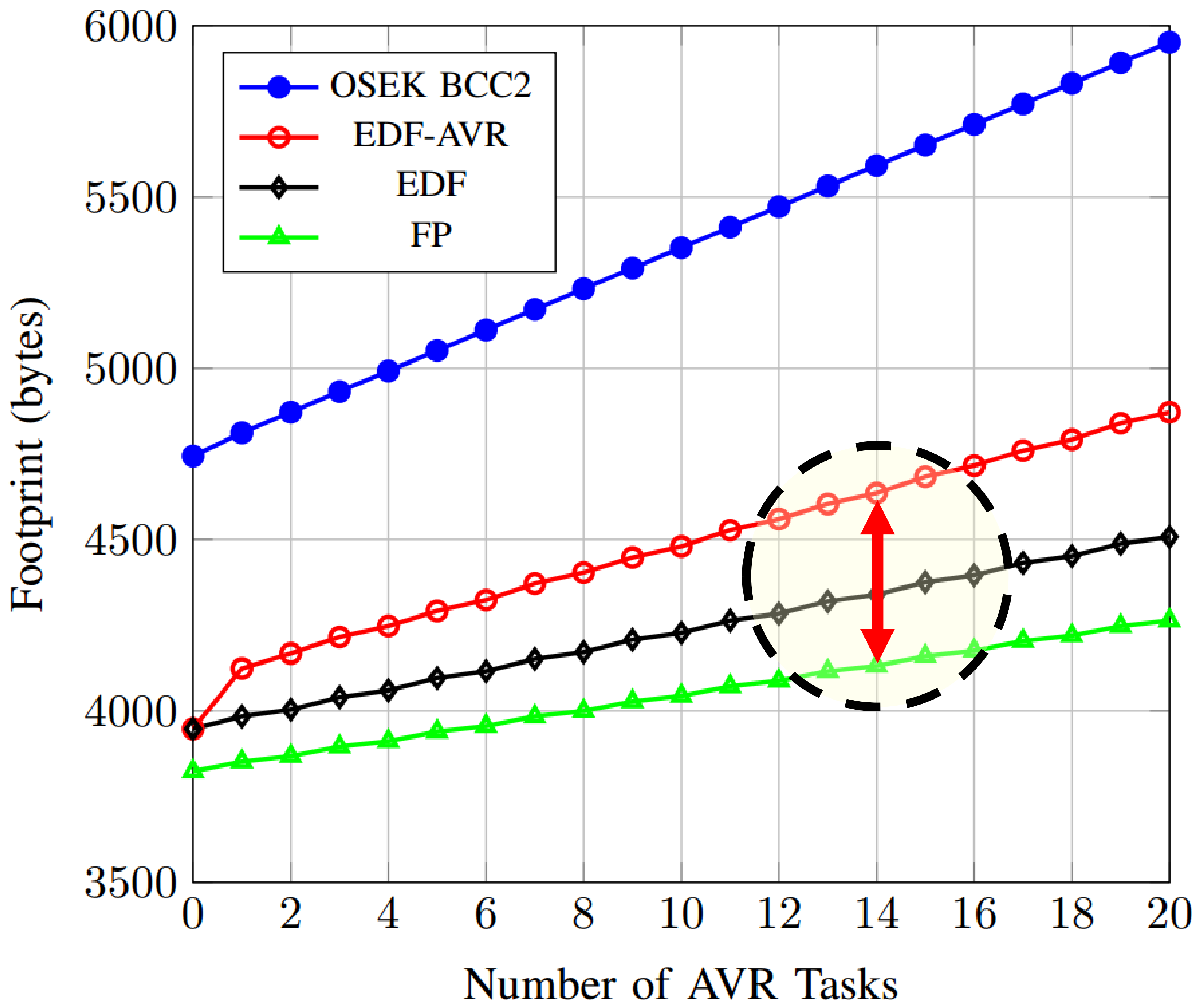
420 cycles

2.2 μs

STM32F4 @ 168Mhz – GNU ARM Compiler

Experimental Results

Footprint – 2 periodic tasks + #n AVR tasks



STM32F4 – GNU ARM Compiler with -Os

INSTITUTE OF COMMUNICATION, INFORMATION AND PERCEPTION TECHNOLOGIES

Scuola Superiore Sant'Anna

Real-Time Systems Laboratory

A SIMULATION FRAMEWORK

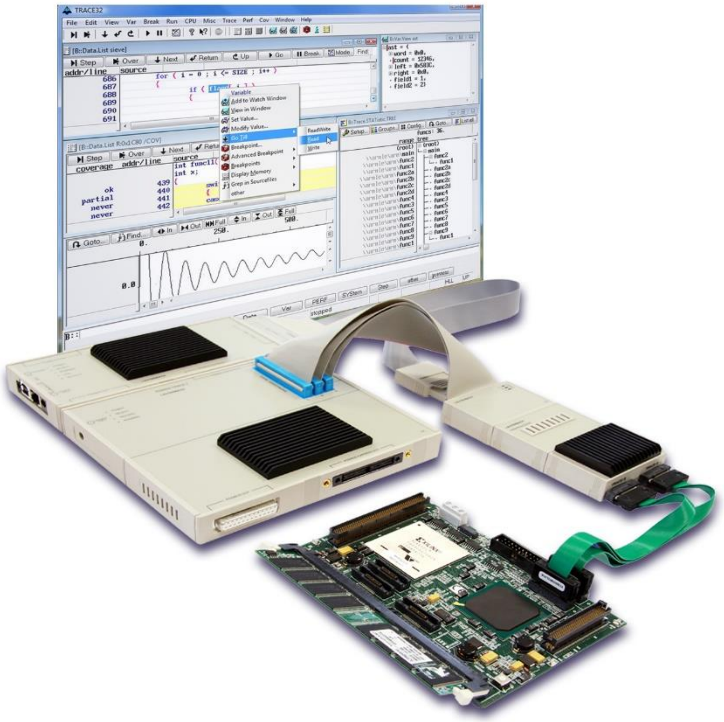
Simulation Framework

- **Lauterbach** is the world's larger producer of hardware assisted debug tools for microprocessors

- TRACE32® PowerView IDE

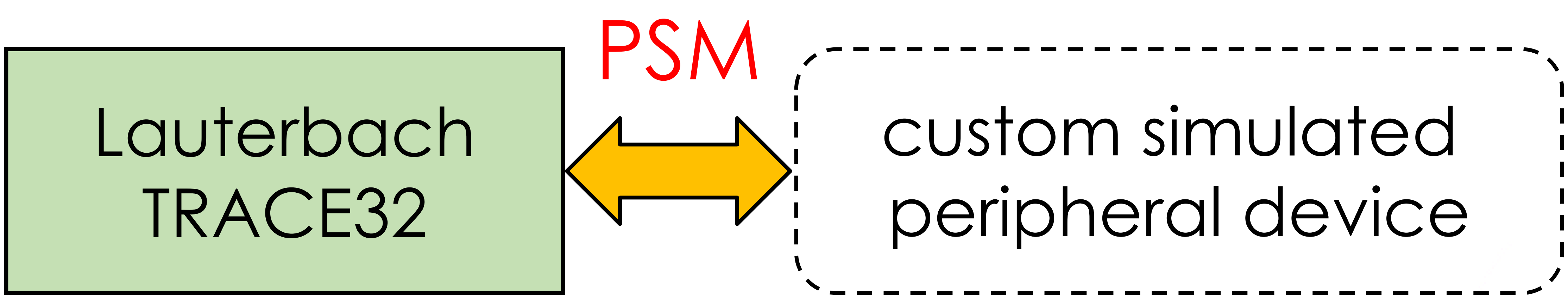
- Lauterbach makes available a version of their IDE based on an instruction-set simulator
  → trace & debug without any hardware!

LAUTERBACH
DEVELOPMENT TOOLS

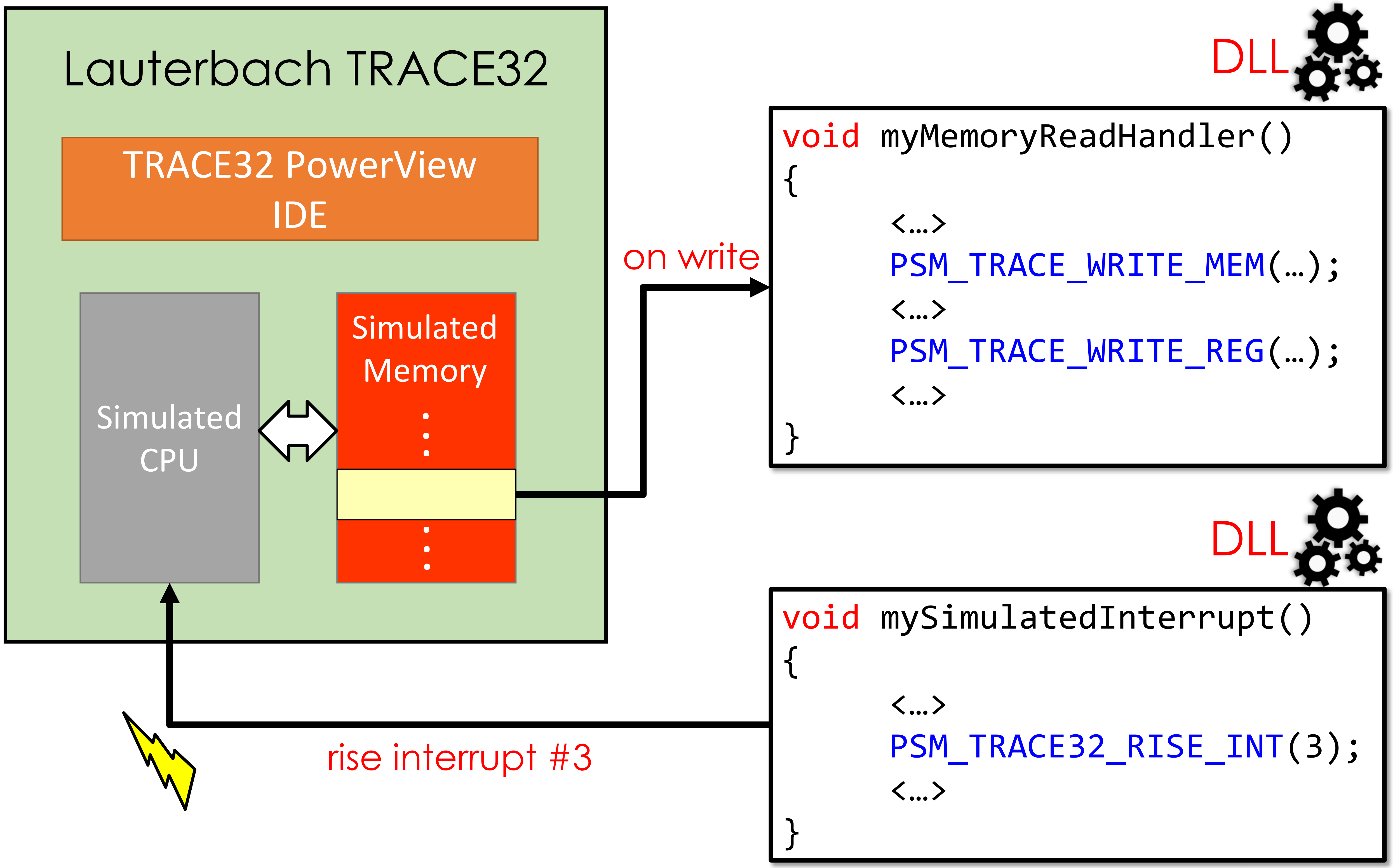


Simulation Framework

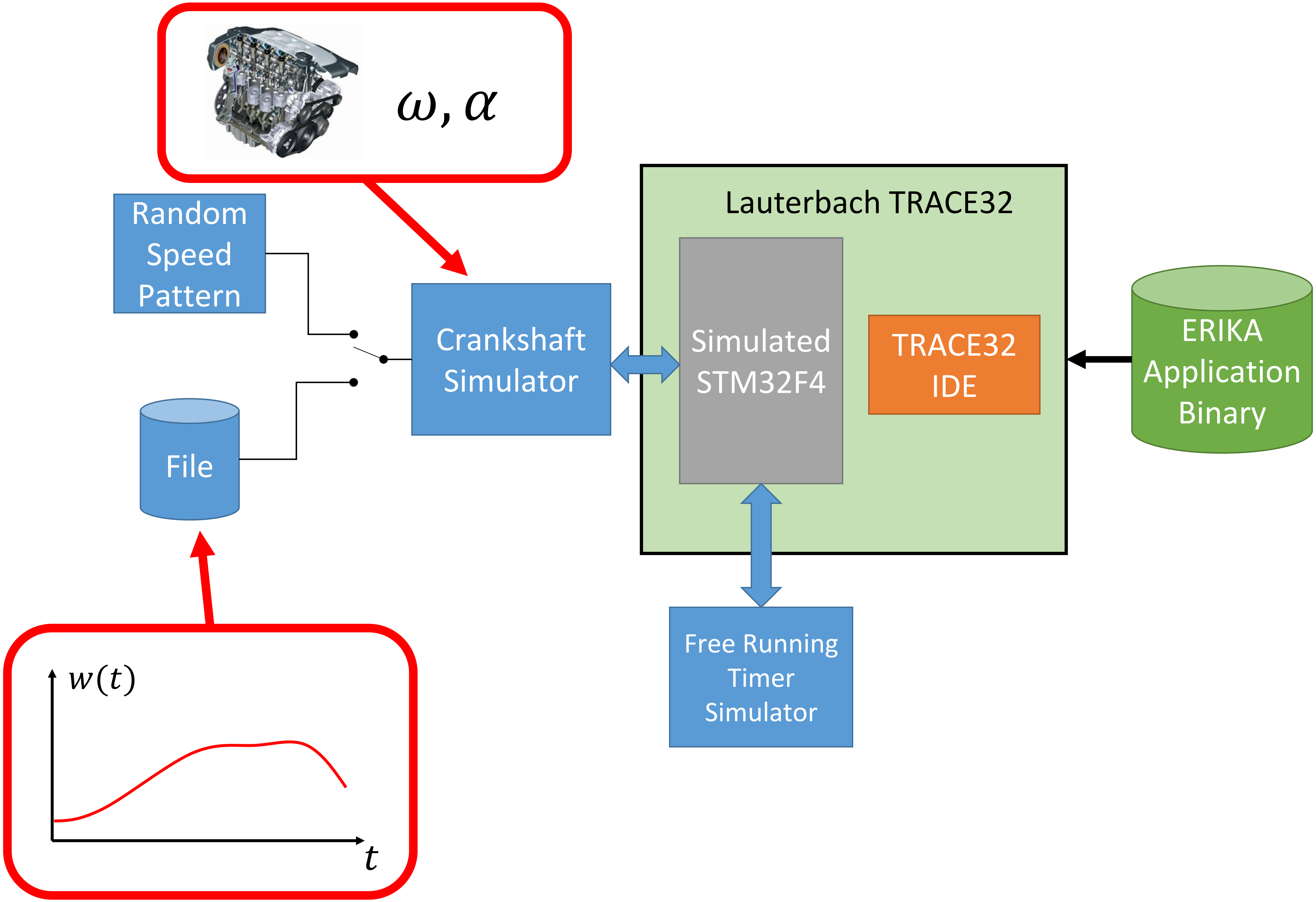
- The TRACE32 simulator offers a standard interface named *Peripheral Simulation Model* (PSM)
- The PSM allows developing custom simulated peripheral devices
  - react to events (e.g., memory read);
  - access to the simulated CPU registers;
  - …



Simulation Framework



Simulation Framework



INSTITUTE OF COMMUNICATION, INFORMATION AND PERCEPTION TECHNOLOGIES

Scuola Superiore Sant'Anna

Retis
Real-Time Systems Laboratory

DEMO

Conclusions

- We presented a new RTOS support for **EDF** scheduling of engine control applications
- The implementation has been conceived to require **minimal changes** to existing applications (**OSEK-like API**, integration with **OIL**)
- Run-time overhead and footprint are not problems (**+1.5 μs** and **+500** bytes over an implementation of fixed-priority scheduling)
- We also present a powerful simulation framework for studying the execution of real code under (but not only limited to) the proposed RTOS

Future Work

- We are going to test this implementation with a **real** engine control application controlling a **real** engine
- Integration of the TRACE32 simulator with MATLAB Simulink and/or other physical simulation tools

soon available as open-source

`http://erika.tuxfamily.org/`

Acknowledgements

- Thanks to **Paolo Gai** from Evidence S.R.L. and **Maurizio Menegotto** from Lauterbach Italia for their support which helped to improve this work.

Thank you!

Alessandro Biondi
alessandro.biondi@sssup.it