

REAL-TIME SCHEDULING

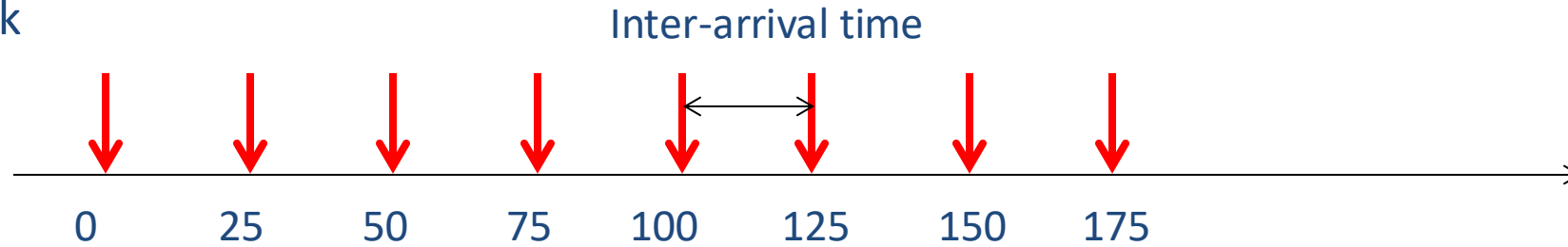
STEFANO DI CARLO

INTRODUCTION

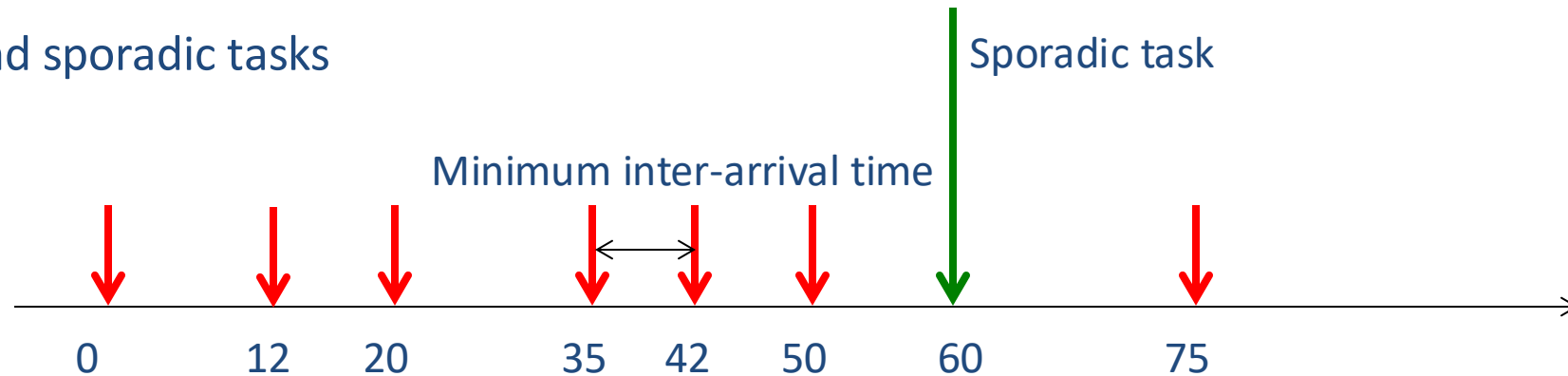
- ▶ Tasks can be
 - ▶ **Periodic**: it consists of an infinite sequence of identical activities called instances, or jobs, that are regularly activated, at a constant rate
 - ▶ **Aperiodic**: it consists of an infinite sequence of identical jobs, which are not activated at a regular rate
 - ▶ **Sporadic**: an aperiodic task for which it is not possible to determine a minimum inter-arrival time interval is called a sporadic task

EXAMPLES

► Periodic task

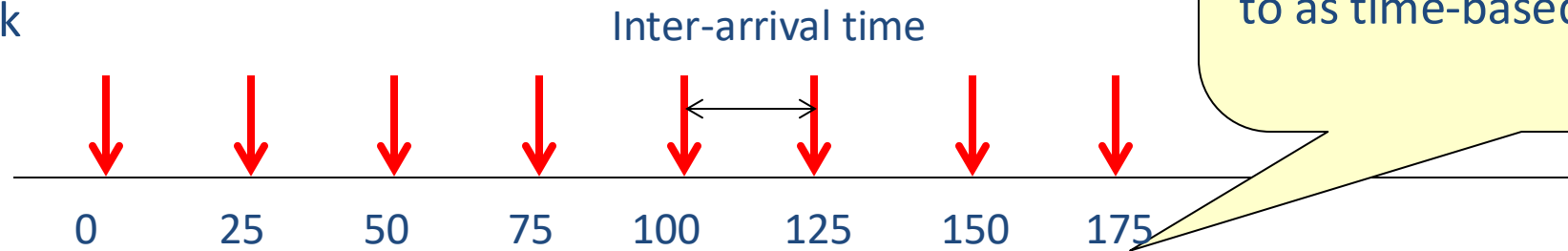


► Aperiodic and sporadic tasks



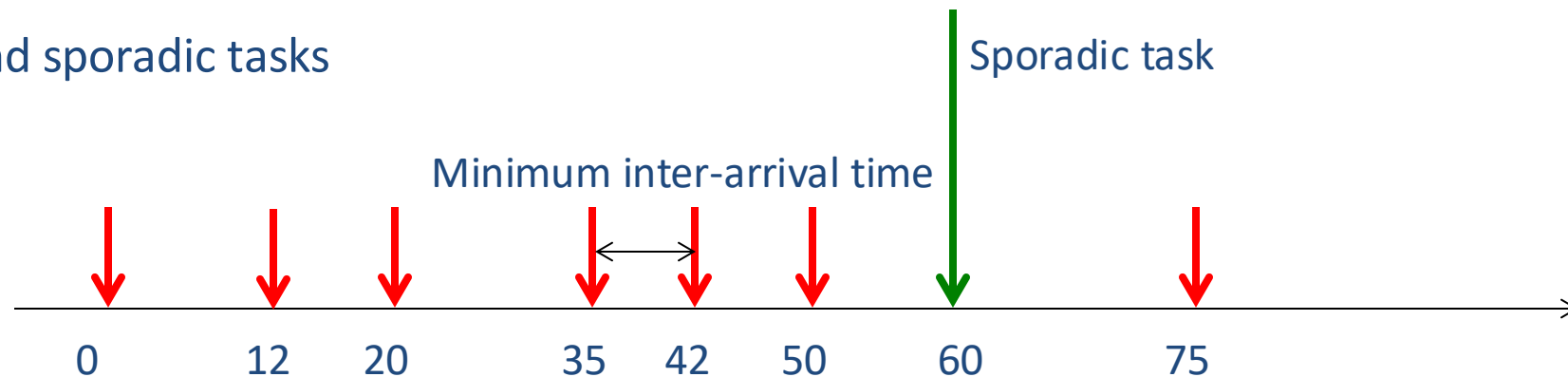
EXAMPLES

► Periodic task



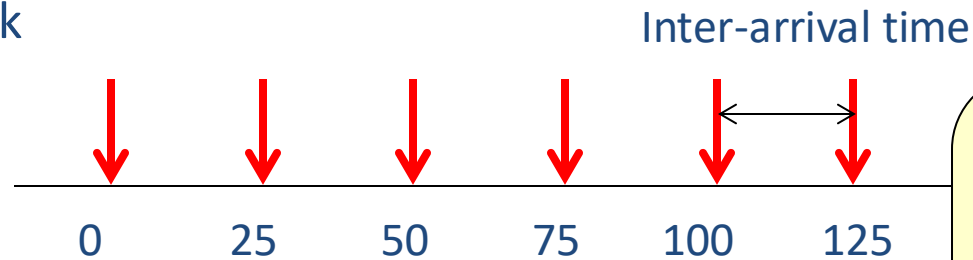
Periodic task: task to measure the in-take air temperature, which runs every 25 msec. Periodic tasks are often referred to as time-based tasks.

► Aperiodic and sporadic tasks



EXAMPLES

► Periodic task

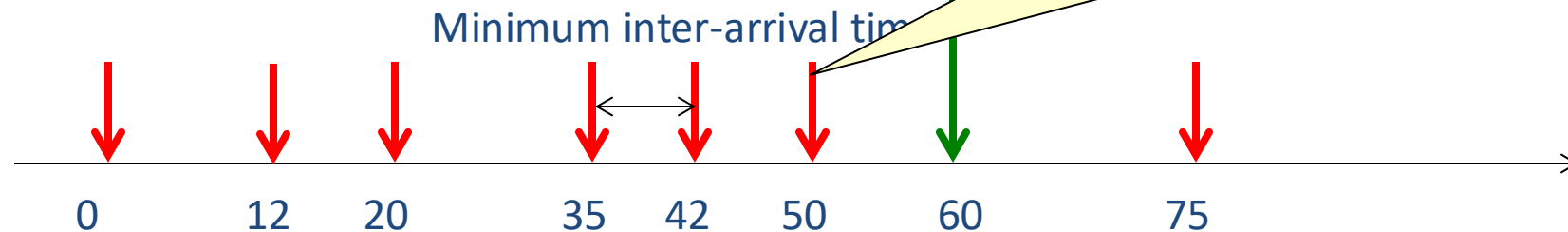


Aperiodic task: task to compute the center of combustion (MFB50), which runs every time the crank shaft turns by 1°.

1000 RPM → 25 msec

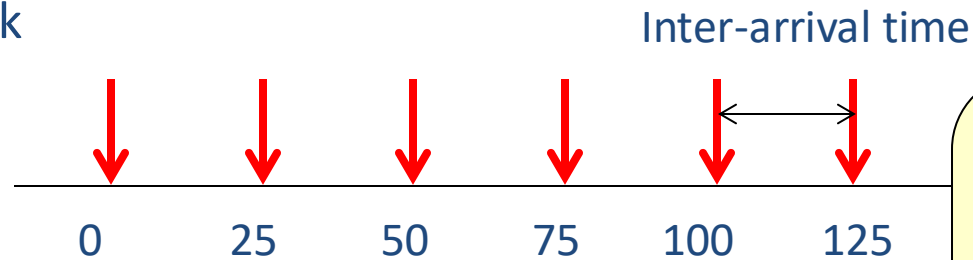
4000 RPM → 6,25 msec

► Aperiodic and sporadic tasks



EXAMPLES

► Periodic task

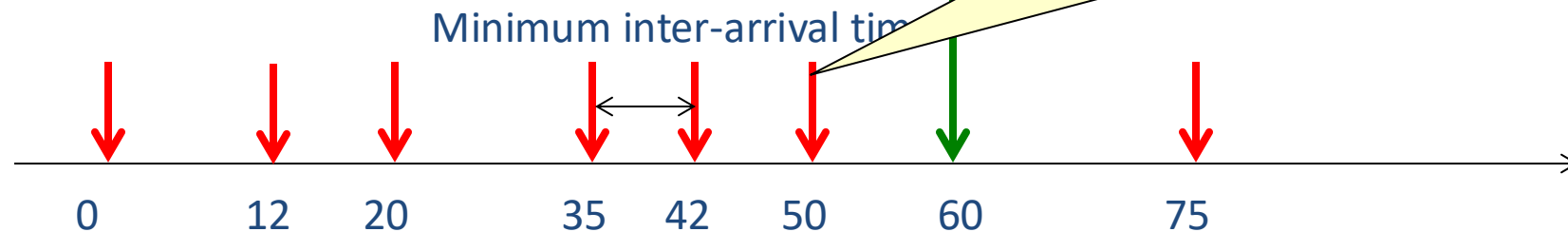


Aperiodic task: task to compute the center of combustion (MFB50), which runs every time the crank shaft turns by 1°.

1000 RPM → 25 msec

4000 RPM → 6,25 msec

► Aperiodic and sporadic tasks



ASSUMPTIONS

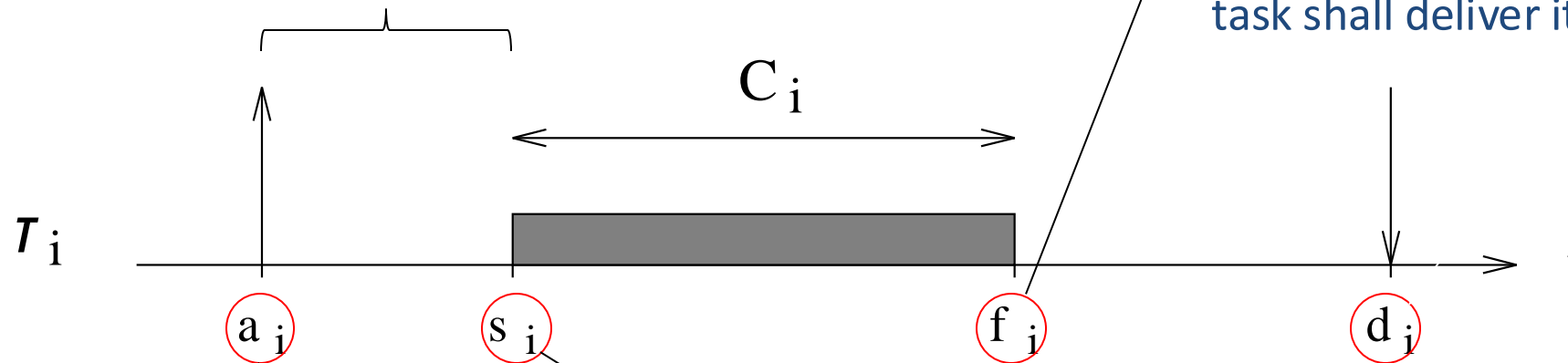
- ▶ Fixed number of tasks
- ▶ Periodic task with known period
- ▶ Independent task
- ▶ Hard deadline for each task equal to its period
- ▶ Known and fixed worst-case execution time
- ▶ All system's overheads, for example context switch times, are negligible

NOTATION

Dispatch latency = time the task spends in the ready queue before running

Finishing time = the time when the task actually delivers its outputs

Deadline time = the time when the task shall deliver its outputs



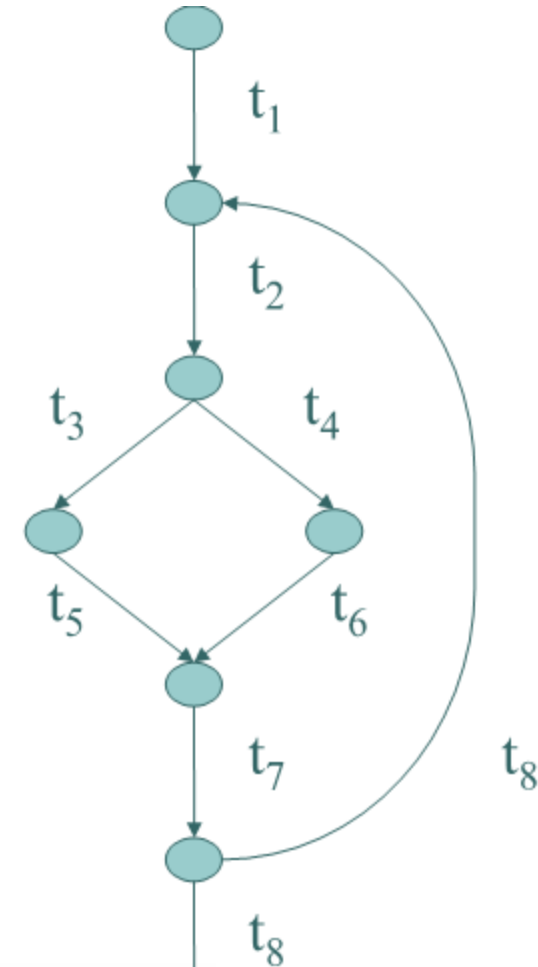
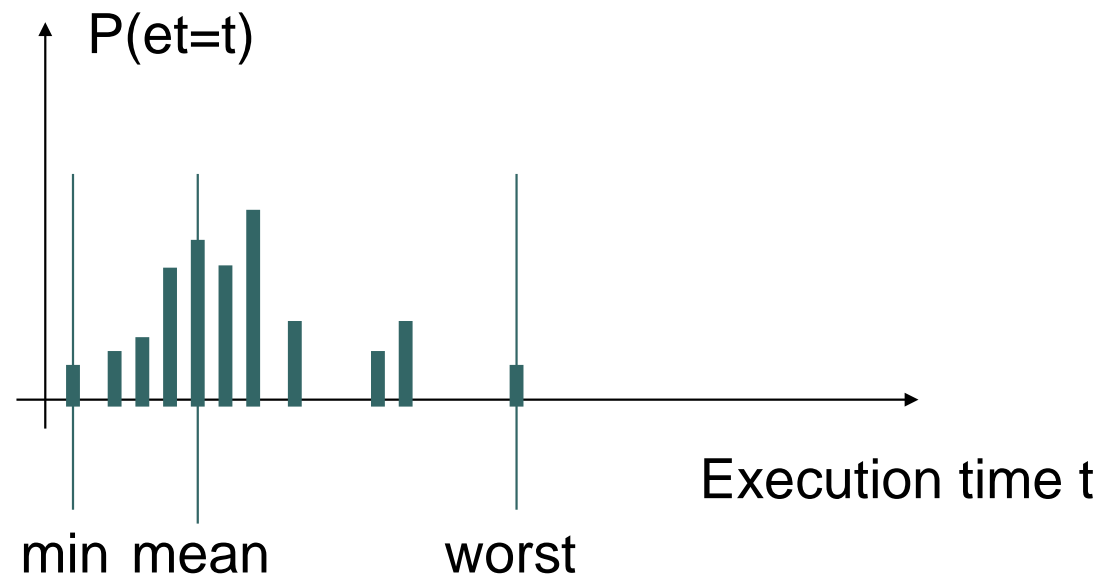
Activation time = the time when the task is needed

Start time = the time when the task starts running

Note: in general, $f_i \neq s_i + C_i$ due to preemption. Indeed, the task may be interrupted (due to I/Os or higher-priority tasks).

WHY WORST-CASE EXECUTION TIME?

- ▶ A task may be composed of different input-dependent execution paths
- ▶ Each execution path is composed of different actions, each with its own execution time
- ▶ Depending on the input, the task may exhibit different execution times, being



NOTATION

- ▶ The processor utilization factor U is the fraction of processor time spent in the execution of the n tasks
- ▶ $\frac{C_i}{T_i}$ is the fraction of processor time spent in executing task τ_i

$$U = \sum_{i=1}^n \frac{C_i}{T_i}$$

- ▶ A scheduling is said to be feasible if it satisfies a set of constraints
 - ▶ Timing constraints such as activation period, deadline, ...
 - ▶ Precedence in terms of order of execution of tasks
 - ▶ ...

EXERCISE

- ▶ Given the following task set compute the processor utilization factor U

Task	T [ms]	WCET [ms]
A	4	2
B	8	1
C	12	1

QUESTIONS?

THANK YOU!

