technische universität
Dortmund

Department of
Computer Science

Computer
Science 12

Implementation and Evaluation of Multi-Mode Real-Time Tasks under Different Scheduling Algorithms

Anas Toma, Vincent Meyers and Jian-Jia Chen
TU Dortmund University, Germany

Supported by DFG as part of the Collaborative Research Center SFB876

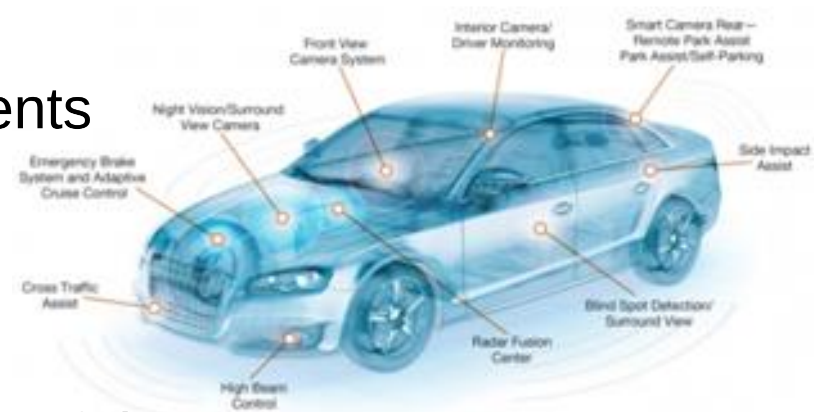
03.07.2018

Introduction - Automotive Systems

Electronic Control Units (ECUs)

- Control and improve functionalities, performance and safety

- Continuous interaction with components
  - Doors , lights, **engine**, etc.

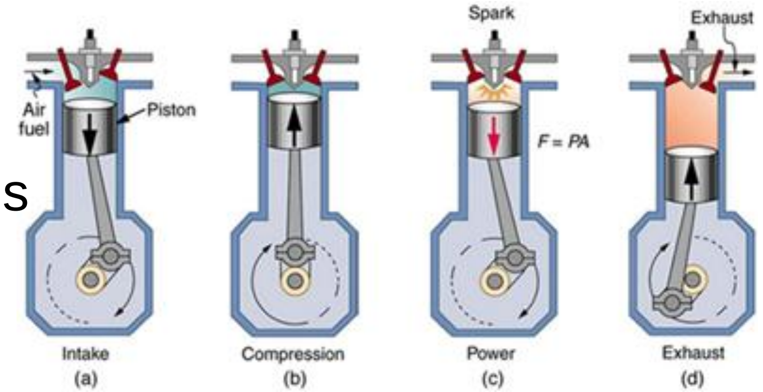


- Should react within a specific amount of time
  - A delayed reaction may affect the safety

[www.autotechreview.com]

Engine Control

- Tasks:
  - Adjusting the fuel flow
  - Calculating the time of the spark signal
  - Minimizing fuel consumption and emissions



- **Angular synchronous tasks**
  - Linked to the rotation of the crankshaft
  - Increasing rotation speed → Shorter period/deadline
    - Drop some non-critical functions to meet the deadline
  - Releases jobs depending on the engine's rotation speed
    - Different execution modes → *Multi-Mode Task Model*
      - » Digraph Real-Time model (DRT)
      - » Variable Rate-dependent Behavior (VRB) task model

[Engine Repair Indianapolis IN 317-876-9890]

Multi-Mode Tasks

An example of a multi-mode task with three different execution modes

Rotation Speed (rpm)	Mode Type	Executed Functions
$[0, 3000]$	A	$f1$, $f2$ and $f3$
$(3000, 6000]$	B	$f1$ and $f2$
$(6000, 9000]$	C	$f1$

- Different modes: (C^1, T^1, D^1) (C^2, T^2, D^2) (C^3, T^3, D^3)
  - C^j: worst-case execution time (WCET)
  - T^j: period
  - D^j: relative deadline
- Implicit deadline $T^j = D^j$
- The mode changes based on an external interrupt or any other event

The FreeRTOS Kernel

- A Real Time Operating System (RTOS) for microcontrollers and small microprocessors
- Supports many different architectures
- Open source RTOS
- Low ROM and RAM usage
- Simple and easy to use
- Can be also used for educational purposes

[https://www.freertos.org]

Contribution

- Modifying the FreeRTOS real-time operating system to consider the multi-mode real-time tasks
- Implementing the Rate-Monotonic (RM) and the Earliest Deadline First (EDF) scheduling algorithms
- Empirical evaluation of the multimode tasks under EDF and RM algorithms in a real environment
  - FreeRTOS running on Raspberry Pi B+ board

Multi-Mode Task Model Implementation

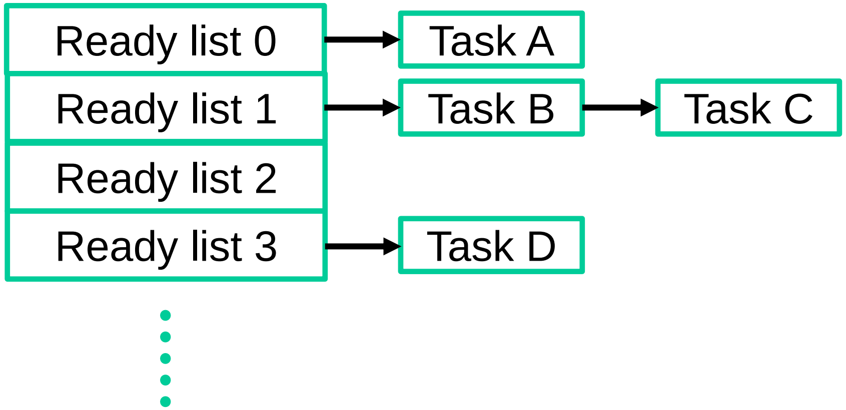
Periodic tasks

- Expanding the task control block (TCB)
  - Period, worst-case execution time, relative deadline and the previous wake time
- *vTaskDelayUntil()* function to delay the task for the specified period

Modes

- TCB fields with array data structure
- Additional attributes
  - number of the modes
  - threshold values for each mode level
- Global variable for the external input
  - Any changes will be applied starting from the next release

Rate-Monotonic (RM) Algorithm Implementation



- Tasks with a shorter period have a higher priority
- Assign priorities before starting the scheduler
- Doubly linked list to sort the tasks according to their periods
- The priorities are assigned for each task for all the modes
  - Array of priorities for each task
- The tasks are moved to their corresponding ready lists

Earliest Deadline First (EDF) Algorithm Implementation

- Assign the highest priority to the job with the earliest absolute deadline

- A doubly linked list for the ready jobs
  - Instead of the array of linked lists provided by FreeRTOS
  - Apply binary heap

- Once a job is added to the ready list
  - The absolute deadline is calculated
  - The job with the earliest absolute deadline is scheduled for execution

Scheduling in FreeRTOS

- Shared Processor Behavior (round-robin)
  - Context switching for every system tick ~4µs → additional overhead!
    - Two tasks with the same priority
    - one ready task

Additional Modifications

+ Tasks with the same priority are scheduled according to their insertion order in the ready list

+ Perform context switching only if
  - a new job with a higher priority arrives, or
  - the current job under execution is blocked

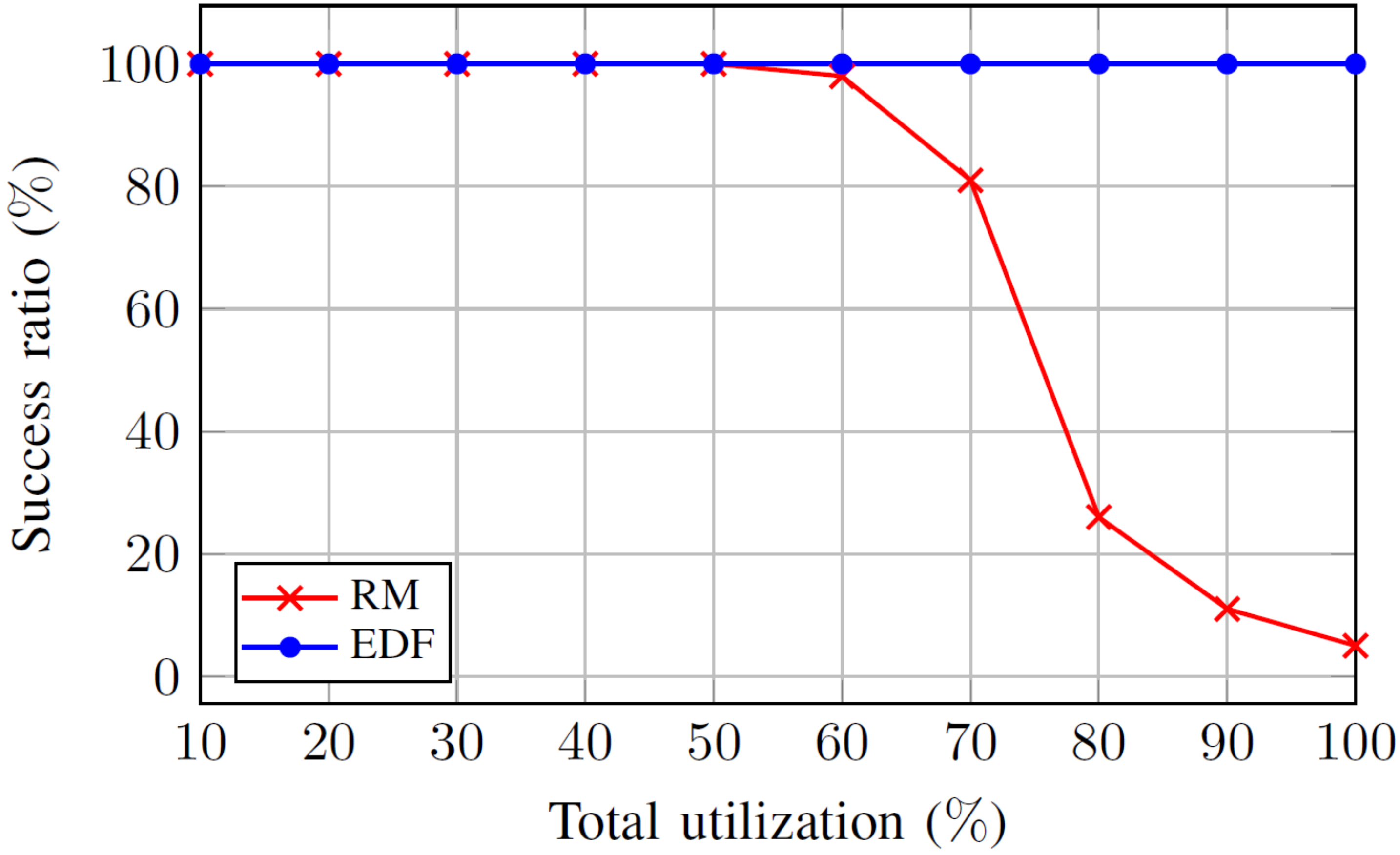
Experimental Evaluation - Synthetic Workload

- Utilizations and computation segments: [10%-100%]
  - Uniform distribution according to *UUniFast**
- Periods: [1-100ms]
  - Log uniform distribution
- For the multi-mode tasks, the WCET and the period values for the remaining modes were scaled by the factor of 1.5
  - $C_i^{m+1} = 1.5 * C_i^m$
  - $T_i^{m+1} = 1.5 * T_i^m$
- 100 task sets with 50% multi-mode tasks and cardinality of 10

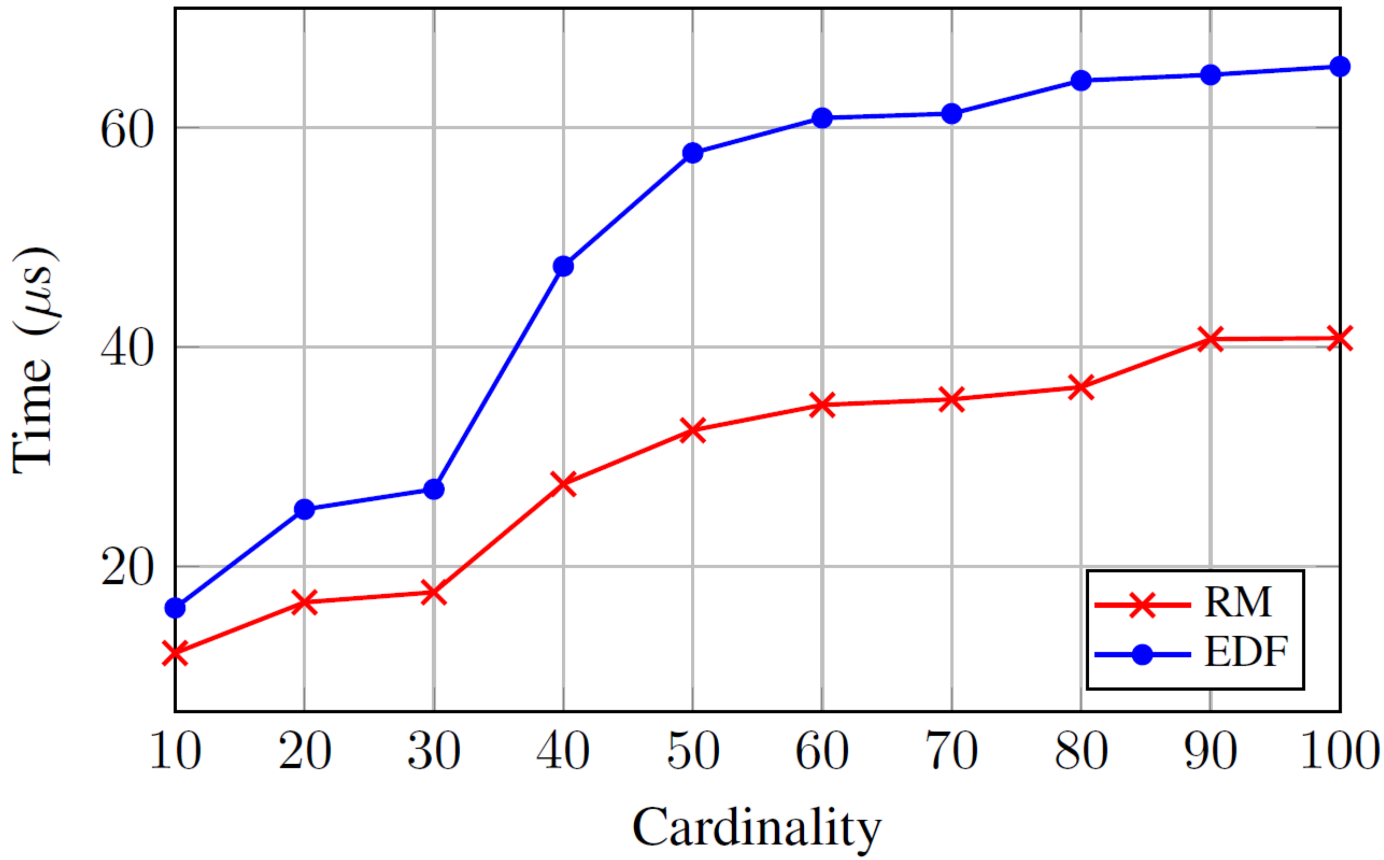
[*E. Bini and G. C. Buttazzo. Measuring the performance of schedulability tests. Real-Time Systems, 30(1):129–154, 2005]

Experimental Evaluation - Synthetic Workload

- 5 modes



Experimental Evaluation – Scheduling overhead



- Cardinality: the number of tasks per a set

Experimental Evaluation – Realistic Workload

- Shared the characteristics of an automotive software system*
  - The distribution of the tasks among the periods
  - The typical number of the tasks
  - The average execution time
  - Factors for determining the best- and worst-case execution times

Task distribution among periods

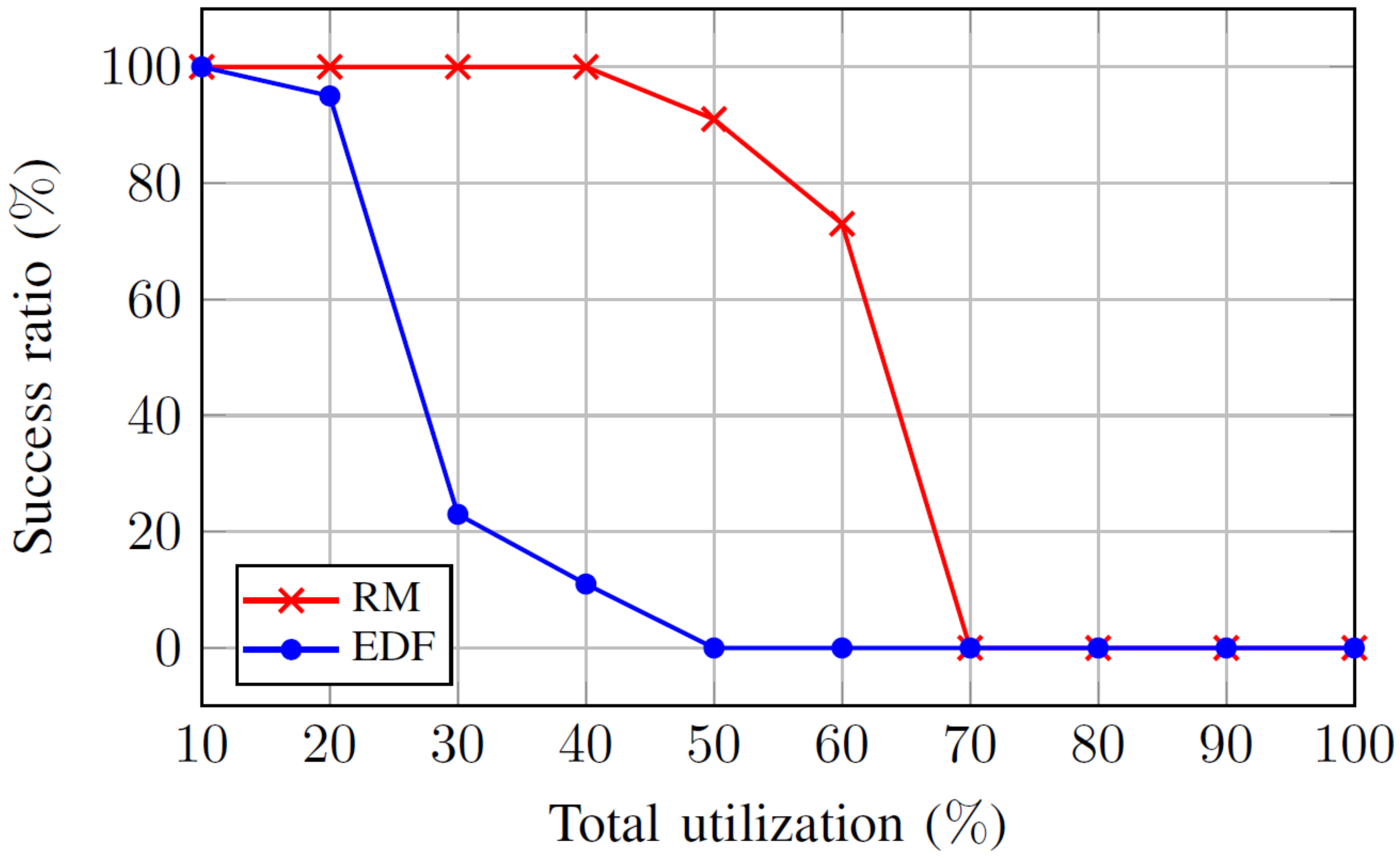
Period	Share
1 ms	3 %
2 ms	2 %
5 ms	2 %
10 ms	25 %
20 ms	25 %
50 ms	3 %
100 ms	20 %
200 ms	1 %
1000 ms	4 %
angle-synchronous ms	15 %

6 modes ranging from 0 to 6000 rpm with their periods in milliseconds

Mode	0	1	2	3	4	5
Min. Max	0 1000	1001 2000	2001 3000	3001 4000	4001 5000	5001 6000
Period	30	15	10	7.5	6	5

[S. Kramer, D. Ziegenbein, and A. Hamann. Real world automotive benchmarks for free]

Experimental Evaluation – Realistic Workload



Conclusion

- Multi-mode tasks were evaluated under the EDF and the RM scheduling algorithms in a real environment
  - FreeRTOS real-time operating system was modified
  - Raspberry Pi B+ board
  - Synthetic and realistic data sets

- Synthetic workload: The EDF algorithm was able to find more feasible schedules than the RM algorithm
  - for high utilization values

- Realistic workload: EDF performed poorly
  - Scheduling overhead of EDF
  - Tasks with shorter periods

Thank you