Engine Control Task Models

Mikail Yayla

LS 12, TU Dortmund

October 11, 2017

Outline

- Introduction and Motivation
  - Engine Basics
  - Challenges in Engine Control
- Engine Control Task Model
  - Task and Rotation Source Model
- Response-Time Analysis of Engine Control Systems
  - Exact test by Biondi et al.
  - ILP based sufficient test by Davis et al.
- Comparison and Experiment Results
- Related Work

Engine Control Task Models

- A. Biondi, M. Di Natale, and G. Buttazzo
  @ ICCPS 2015
  Response-Time Analysis for Real-Time Tasks in Engine Control Applications

- R. I. Davis, T. Feld, V. Pollex and F. Slomka
  @ RTAS 2014
  Schedulability Tests for Tasks with Variable Rate-Dependent Behaviour under Fixed Priority Scheduling

Introduction to Engine Control

- Typical example of Cyber-Physical Systems
- Control software adapts to physical characteristics of engine
- Engine Control Unit (ECU) processes data in real time
- The faster the motor speed, the faster reaction of system

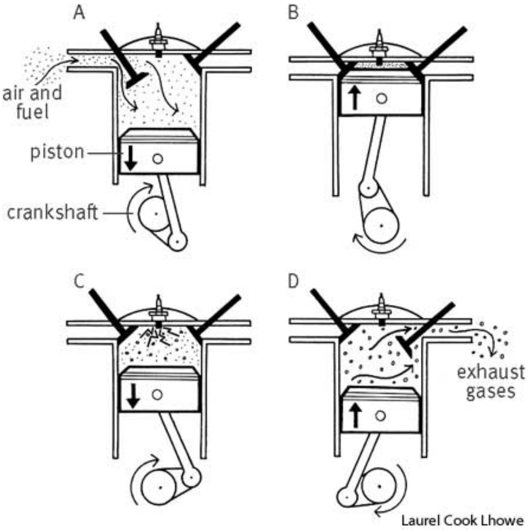
Control what ?

Source: motortrend.com/cars/bmw/6-series/2017/

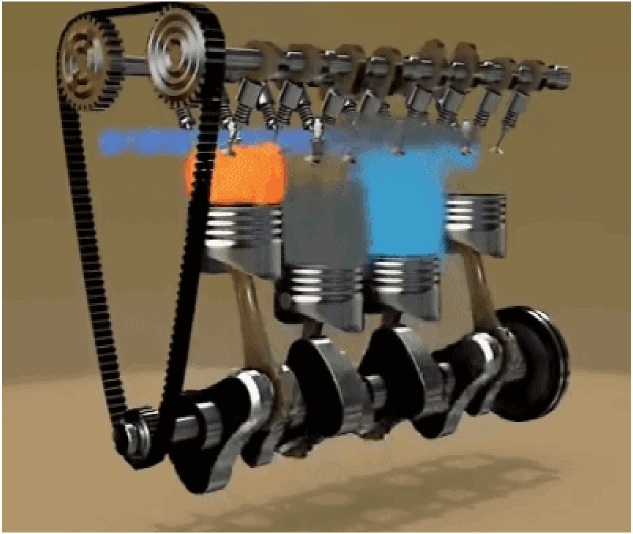
Engine Basics (1/3): Power Generation

Four Strokes

1. Intake
2. Compression
3. Combustion
4. Exhaust



Engine Basics (2/3): 3D View



Source: reddit.com/r/mechanical_gifs/comments/2j9t0x/four_stroke_engine/

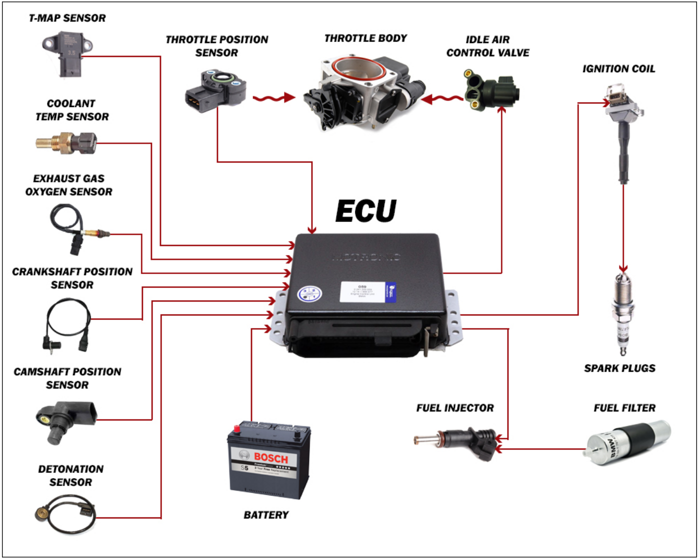
Engine Basics (3/3): A whole car engine

Engine Parts

- Engine block
- Cylinder
- Crankshaft
- Piston
- Piston rod
- Spark plug
- Intake and exhaust valves

Source: walkersautotech.co.uk/terraclean.htm

Engine Control Unit (ECU)



Source: bimmian.com/blog/2016/07/07/Software+and+Tuning

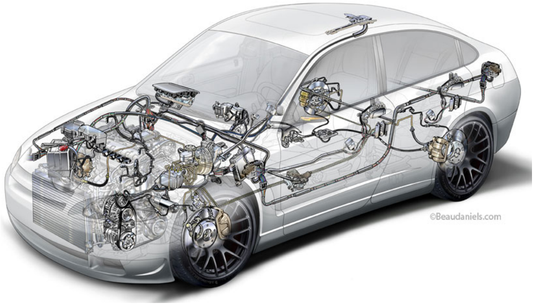
Issues in Engine Control

Engine Management

- Timing of injection and sparks
- Emission control

Real-time issues of Engines

- Deadlines, to guarantee timing of processes
- Control rotating or moving parts
- Data communication



Source: beaudaniels.com

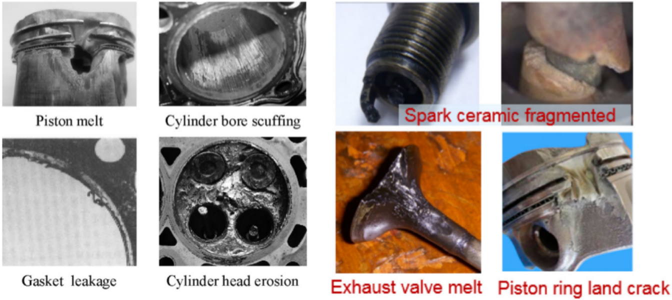
Challenges in Engine Control

Fuel Efficiency

- The better the air/fuel ratio, the better the energy efficiency

Knocking

- Ignition can occur on its own - causes separate explosion



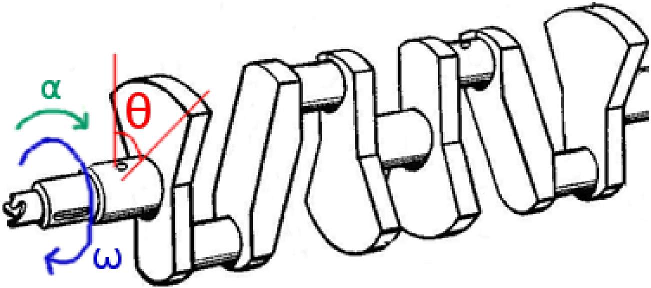
Engine damage due to knocking[1]

[1] Z. Wang, H. Liu, R. D. Reitz, Knocking combustion in spark-ignition engines, Progress in Energy and Combustion Science, Volume 61, 2017

Rotation Source Model

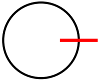
Engine State is described by:

- θ current rotation angle of the crankshaft
- ω current angular speed, $[\omega^{min}, \omega^{max}]$
- α current angular acceleration, $[\alpha^{-}, \alpha^{+}]$



Engine Control Task Model Requirements

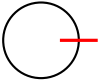
Description of our Tasks



- Triggered at predefined rotation angles
- Crankshaft: 600 to 6000 rpm
- Higher speed $\rightarrow$ higher task activation rate
- Execution times: a few ms up to 100 ms
- Speed changes due to acceleration

Engine Control Task Model Requirements

Description of our Tasks

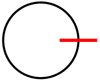


- Triggered at predefined rotation angles
- Crankshaft: 600 to 6000 rpm
- Higher speed → higher task activation rate
- Execution times: a few ms up to 100 ms
- Speed changes due to acceleration

Activations depend on Rotation Speed

Engine Control Task Model Requirements

Description of our Tasks



- Triggered at predefined rotation angles
- Crankshaft: 600 to 6000 rpm
- Higher speed $\rightarrow$ higher task activation rate
- Execution times: a few ms up to 100 ms
- Speed changes due to acceleration

Activations depend on Rotation Speed

Different execution versions for different speeds

- Certain task functions are disabled for high speeds

AVR Tasks

Activation of AVR Tasks

- Period depends on ω, i.e. $T_i(\omega) = \frac{\Theta_i}{\omega}$
- Can be activated at every full revolution: $2\pi, 4\pi, 6\pi, \ldots$

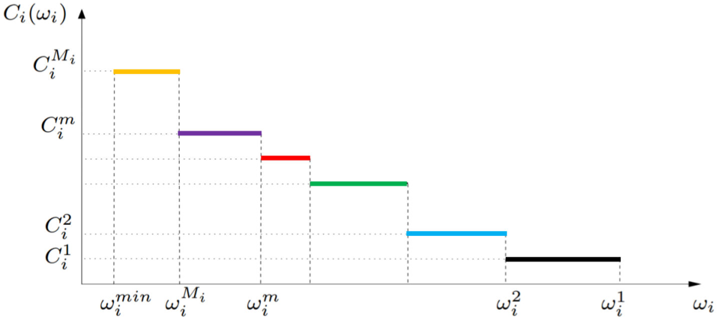
AVR Tasks

Activation of AVR Tasks

- Period depends on ω, i.e. $T_i(\omega) = \frac{\Theta_i}{\omega}$
- Can be activated at every full revolution: $2\pi, 4\pi, 6\pi, \ldots$

Implemention in Modes

- Set of modes $M_i = \{(C_i^m, \omega_i^m), m = 1, 2, \ldots, M_i\}$
- C_i^1 valid in speed range $(\omega_i^2, \omega_i^1]$, C_i^2 in $(\omega_i^3, \omega_i^2]$, etc.



Adapted from Biondi (2014)

Implementation of AVR Tasks

TASK *task*1 {

$\quad \omega = getRotationSpeed()$

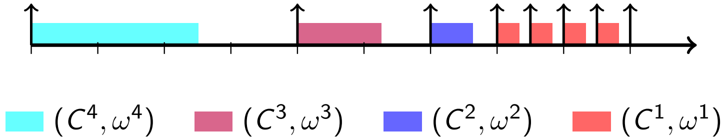
$\quad$ **if** $(\omega \leq 6000)$ $f1()$

$\quad$ **if** $(\omega \leq 4000)$ $f2()$

$\quad$ **if** $(\omega \leq 2000)$ $f3()$

$\quad$ **if** $(\omega \leq 1000)$ $f4()$

}



Engine Control Task Set

Task Properties

- Real-time preemptive tasks $\Gamma = \{\tau_1, \tau_2, \ldots, \tau_n\}$
- Either **periodic** or **AVR** tasks
- Set of periodic tasks and set of AVR tasks is disjunct
- WCET C_i, inter-arrival time T_i, and deadline D_i

Engine Control Task Set

Task Properties

- Real-time preemptive tasks $\Gamma = \{\tau_1, \tau_2, \ldots, \tau_n\}$
- Either **periodic** or **AVR** tasks
- Set of periodic tasks and set of AVR tasks is disjunct
- WCET C_i, inter-arrival time T_i, and deadline D_i

AVR vs. Periodic

Task type	AVR	Periodic
Activation	Angular position of crankshaft	T_i
Modes	m modes	One mode
Execution time	C_i^m	C_i
Release pattern	Not known a priori	Known a priori

Dynamics of the Engine

Relations between task parameters and dynamics of engine

- Given engine state (ω, α) at time t
- Next job release of an AVR task $T_i(\omega, \alpha)$
- Relative deadline of an AVR task $D_i(\omega)$
- Engine speed at next task release $\Omega_i(\omega_k, \alpha_k)$

[2] G. C. Buttazzo, E. Bini and D. Buttle, Rate-adaptive tasks: Model, analysis, and design issues, DATE, Dresden, 2014

Dynamics of the Engine

Relations between task parameters and dynamics of engine

- Given engine state (ω, α) at time t
- Next job release of an AVR task $T_i(\omega, \alpha)$
- Relative deadline of an AVR task $D_i(\omega)$
- Engine speed at next task release $\Omega_i(\omega_k, \alpha_k)$

Formulas can be derived[2]

- $\alpha(t) = \alpha_0$
- $\omega(t) = \alpha_0 t + \omega_0$
- $\theta(t) = \frac{1}{2}\alpha_0 t^2 + \omega_0 t$

- $T_i(\omega, \alpha) = \frac{\sqrt{\omega^2 + 2\Theta_i \alpha} - \omega}{\alpha}$
- $D_i(\omega) = \frac{\sqrt{\omega^2 + 2\Delta_i \alpha^+} - \omega}{\alpha^+}$
- $\Omega_i(\omega_k, \alpha_k) = \sqrt{\omega^2 + 2\Theta_i \alpha_k}$
- $\ldots$

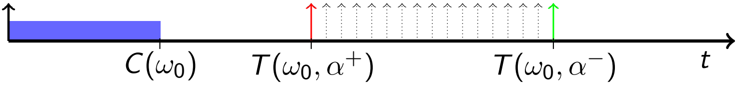
[2] G. C. Buttazzo, E. Bini and D. Buttle, Rate-adaptive tasks: Model, analysis, and design issues, DATE, Dresden, 2014

Job activations of AVR tasks

Consider a job executing in mode m

- Instantaneous speed is ω_0
- We assume $\alpha \in [\alpha^-, \alpha^+]$

When is the next task release?

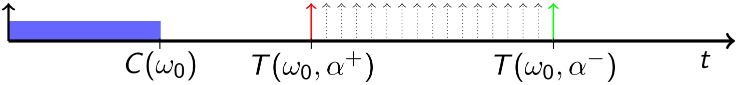


Job activations of AVR tasks

Consider a job executing in mode m

- Instantaneous speed is ω_0
- We assume $\alpha \in [\alpha^-, \alpha^+]$

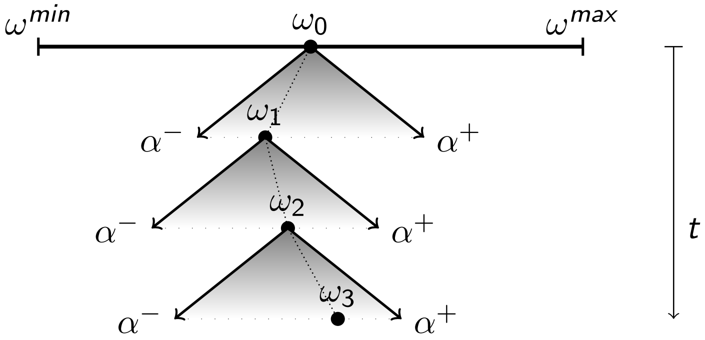
When is the next task release?



Infinitely many time instants for the next job release

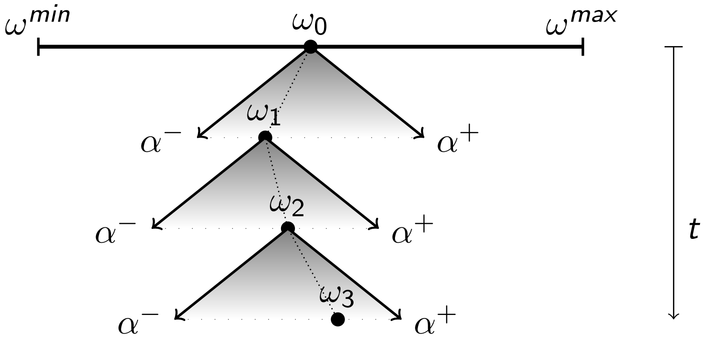
- Infinitely many speed evolution patterns $\{\omega_0, \omega_1, \omega_2, \ldots\}$
- A tree of possible sequences
- Infinitely many subtrees

Tree of possible Sequences for one AVR task

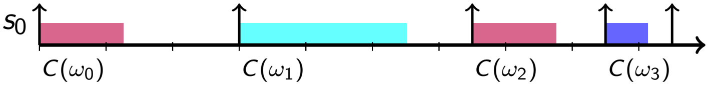


In $[0, t]$**,** $\{\omega_0, \omega_1, \omega_2, \omega_3\}$ **is one sequence** $s_0 \in S(t)$

Tree of possible Sequences for one AVR task



In $[0, t]$**,** $\{\omega_0, \omega_1, \omega_2, \omega_3\}$ **is one sequence** $s_0 \in S(t)$



Recap: Task and Rotation Source Model

Rotation Source

- Engine speed $\omega \in [\omega^{min}, \omega^{max}]$, acceleration $\alpha \in [\alpha^{-}, \alpha^{+}]$
- Angular period $\Theta = 2\pi$

AVR Tasks

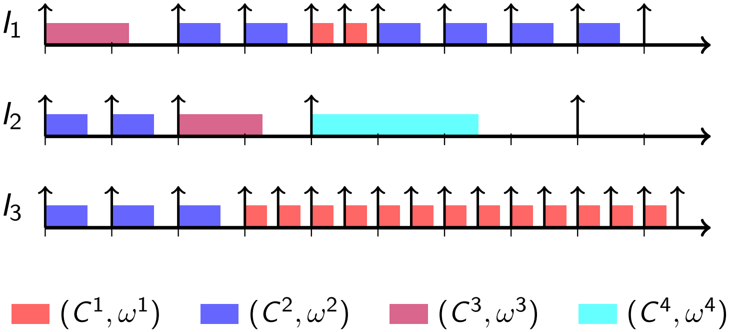
- Period $T_i(\omega, \alpha)$
- Deadline $D_i(\omega)$
- Certain number of task execution modes m
- WCET C_i^m for speed ω_i^m

Speed Evolution pattern

- Tree of possible sequences of ω

AVR Task Interference

Possible activation patterns

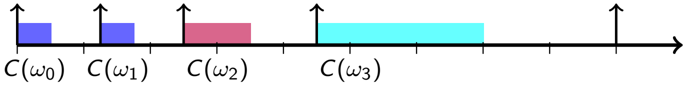


Which one is worst-case?

Interference Computation (Biondi et al.)

Construct Interference of AVR task for one specific speed evolution pattern

$$\boxed{I^{(s)}(t) = C(\omega_0) + \sum_{k=1}^{L} C(\omega_k)\ step(z)}$$

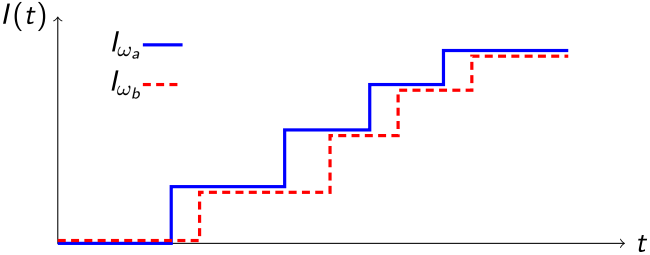


- Workload of one AVR task, for one single path in tree
- Iterate through all possible ω_0 ⚠
- Build up infinite tree at every new ω ⚠

Is there a Way to eliminate Sequences?

If $\omega_a \geq \omega_b$ **and** $C(\ \Omega\ (\omega_a, \alpha^-)\ ) = C(\ \Omega\ (\omega_b, \alpha^-)\ )$

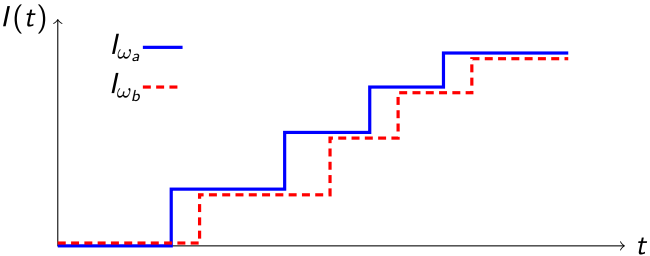
Then $I_{\omega_a}(t) \geq I_{\omega_b}(t)$



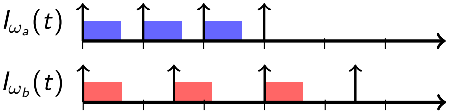
Is there a Way to eliminate Sequences?

If $\omega_a \geq \omega_b$ **and** $C(\ \Omega\ (\omega_a, \alpha^-)\ ) = C(\ \Omega\ (\omega_b, \alpha^-)\ )$

Then $I_{\omega_a}(t) \geq I_{\omega_b}(t)$



$T(\omega_a, \alpha^+) \leq T(\omega_b, \alpha^+)$, $\quad T(\omega_a, \alpha^-) \leq T(\omega_b, \alpha^-)$



Dominant Speeds and Critical Sequence

Dominant Speeds

- $W = \{\omega_1^d, \omega_2^d, \ldots\}$ is the set of dominant engine speeds at release of AVR task instance
- For any $\omega' \notin W$ there exists at least one $\omega_i^d \in W$ with an interference higher or equal to the interference of ω'
- $I^{(s)}(t, i, \omega_i^d) \geqslant I^{(s)}(t, i, \omega')$

Dominant Speeds and Critical Sequence

Dominant Speeds

- $W = \{\omega_1^d, \omega_2^d, \dots\}$ is the set of dominant engine speeds at release of AVR task instance
- For any $\omega' \notin W$ there exists at least one $\omega_i^d \in W$ with an interference higher or equal to the interference of ω'
- $I^{(s)}(t, i, \omega_i^d) \geqslant I^{(s)}(t, i, \omega')$

Critical Sequence

- Sequence $s \in S(t)$ in $[0, t]$ is a critical job sequence of τ_i if all jobs in the sequence are released at a dominant speed
- $CS(t)$ contains all possible critical job sequences

Dominant Speeds and Critical Sequence

Dominant Speeds

- $W = \{\omega_1^d, \omega_2^d, \ldots\}$ is the set of dominant engine speeds at release of AVR task instance
- For any $\omega' \notin W$ there exists at least one $\omega_i^d \in W$ with an interference higher or equal to the interference of ω'
- $I^{(s)}(t, i, \omega_i^d) \geqslant I^{(s)}(t, i, \omega')$

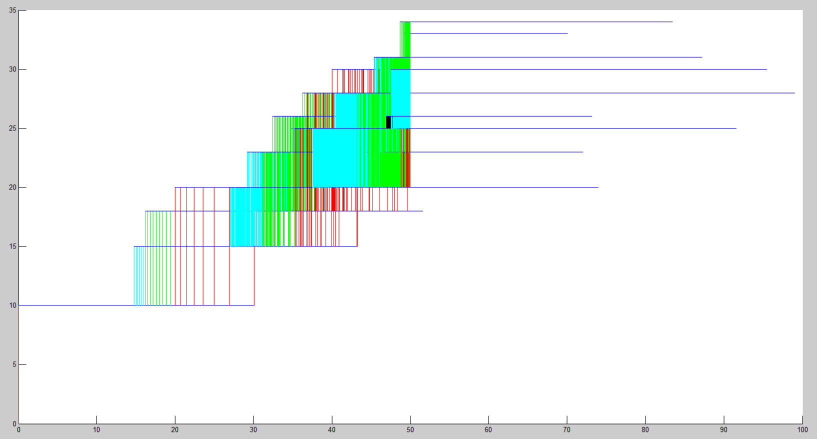
Critical Sequence

- Sequence $s \in S(t)$ in $[0, t]$ is a critical job sequence of τ_i if all jobs in the sequence are released at a dominant speed
- $CS(t)$ contains all possible critical job sequences

Prune search tree and avoid quantization!

Pruning Tree for Interference Calculation (1/2)

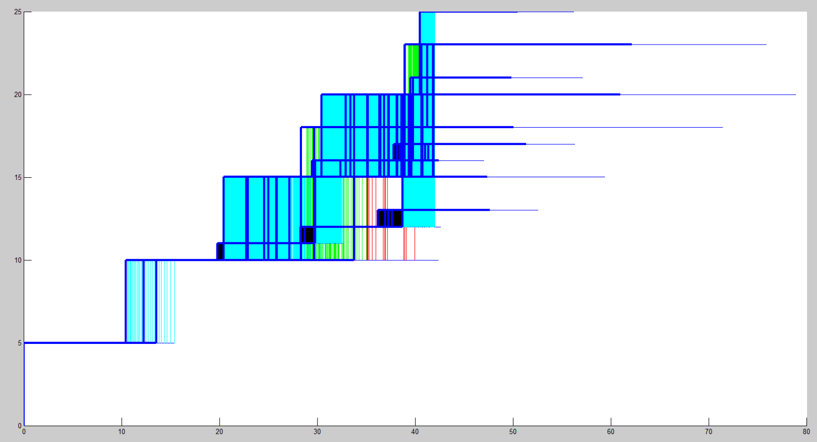
Quantization: 1 hour



Adapted from Biondi (2014)

Pruning Tree for Interference Calculation (2/2)

Dominant Speeds: A few seconds



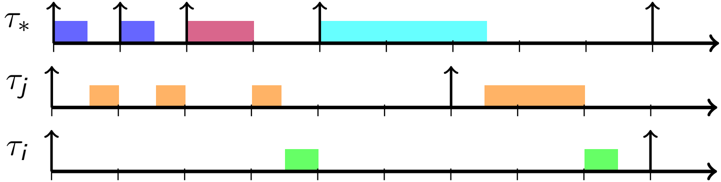
Adapted from Biondi (2014)

Find Worst-Case Response-Time

- Critical instant theorem
- Worst-case interference with dominant speeds

Priorities

- One AVR task τ_* with highest priority
- τ_j with higher priority than τ_i
- We want worst-case response time of τ_i



Exact Response-Time (Biondi et al.)

Response Time $R_i^{(s)}$ of Periodic Task τ_i:

$$\boxed{R_i^{(s)} = min\{t \mid C_i + I^{(s)}(t) + \sum_{j \in hp(\tau_i)} \lceil \frac{t}{T_j} \rceil C_j = t\}}$$

$$\boxed{I^{(s)}(t) = C(\omega_0) + \sum_{k=1}^{L} C(\omega_k)\ step(z)}$$

- $R_i^{(s)}$ requires to calculate $I^{(s)}(t)$ for every critical sequence
- Maximum response time is $R_i = \max_{s \in CS} R_i^{(s)}$

Response Time Calculation Example (Biondi et al.)

$$R_i^{(s)} = min\{t \mid C_i + I^{(s)}(t) + \sum_{j \in hp(\tau_i)} \lceil \frac{t}{T_j} \rceil C_j = t\}$$

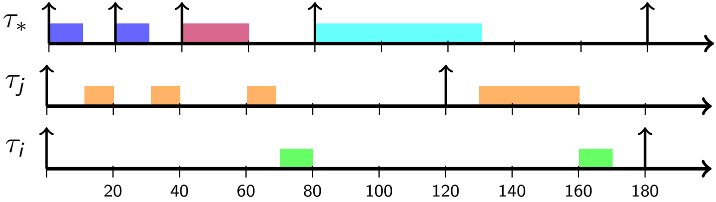
Task	Mode	$C_{i,m}$	$T_{i,m}$
τ_*	1	10	20
	2	20	40
	3	50	100
τ_j	-	30	120
τ_i	-	20	180

$I^{(s)}(t) = 10 + 10 + 20 + 50 = 90$

$C_i = 20$

Response Time

1. $20 + 90 + \lceil \frac{20}{120} \rceil 30 = 140$
2. $20 + 90 + \lceil \frac{140}{120} \rceil 30 = 170$
3. $20 + 90 + \lceil \frac{170}{120} \rceil 30 = 170$



Biondi et al.

Recap of main ideas

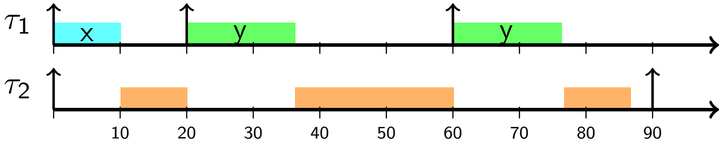
- Tree of possible sequences
- Calculate AVR interference
- Dominant speeds to prune tree
- WCRT and schedulability

Other approaches ?

Task Release Patterns

Task	Mode	$C_{i,m}$	$T_{i,m}$
τ_1	x	10	20
	y	16	40
τ_2		45	89

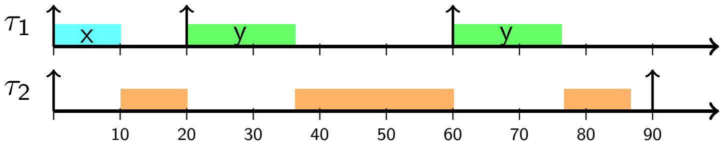
Patterns of τ_1, RT of τ_2			
~~x,x,y,y~~	~~97~~	x,y,y	87
~~y,y,y~~	~~92~~	x,x,x,x	85
~~x,x,x,x,x~~	~~90~~	x,x,y	81



Task Release Patterns

Task	Mode	$C_{i,m}$	$T_{i,m}$
τ_1	x	10	20
	y	16	40
τ_2		45	89

Patterns of τ_1, RT of τ_2			
~~x,x,y,y~~	~~97~~	x,y,y	87
~~y,y,y~~	~~92~~	x,x,x,x	85
~~x,x,x,x,x~~	~~90~~	x,x,y	81



Worst-case response time:

$$1C_{1,x} + 2C_{1,y} + C_2 = 10 + 32 + 45 = 87$$

Interference of AVR Tasks (Davis et al.)

Interference $I_j(S,t)$ of task τ_j in $[0,t]$, starting with speed S

$$\boxed{I_j(S,t) = \sum_{\forall x} k_{j,x} C_{j,x}}$$

Constraints limit the number of:

1. job releases of mode m that can be released in time interval
2. jobs of any mode that can be released in the interval
3. jobs released in adjacent modes; if a job was in mode $m-1$ and $m+1$, it passed through mode m

We can now determine the response time of τ_i

Response Time of task τ_i (Davis et al.)

$$\boxed{w_{i,m}^{q+1}(S) = B_{i,m} + C_{i,m} + \sum_{j \in hp(\tau_i)} I_j(S, w_{i,m}^q)}$$

$$\boxed{I_j(S,t) = \sum_{\forall x} k_{j,x} C_{j,x}}$$

- Begin at a certain speed
- Iteration starts with $w_{i,m}^0 = C_{i,m}$

Quantization of S...

Response Time Calculation (Davis et al.)

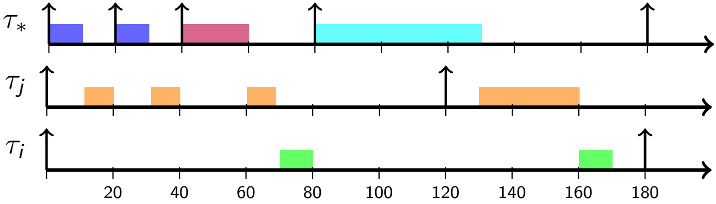
$$w_{i,m}^{q+1}(S) = B_{i,m} + C_{i,m} + \sum_{j \in hp(\tau_i)} I_j(S, w_{i,m}^q)$$

Task	Mode	$C_{i,m}$	$T_{i,m}$
τ_*	1	10	20
	2	20	40
	3	50	100
τ_j	-	30	120
τ_i	-	20	180

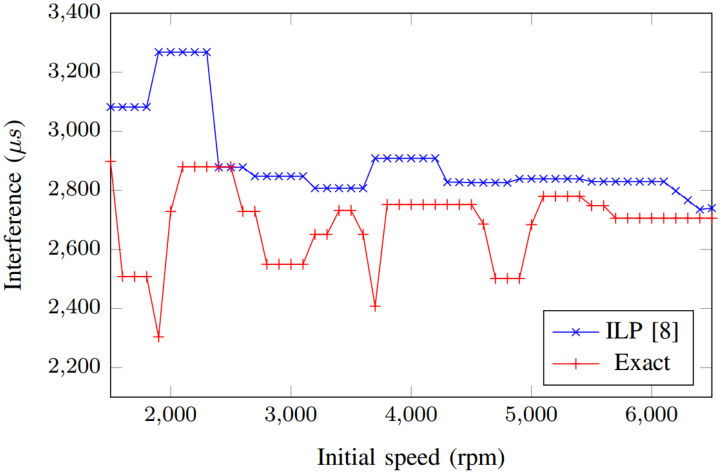
$I_j(S,t) = 10 + 10 + 20 + 50 = 90$
$C_i = 20$

Response Time

1. $20 + 90 + \lceil \frac{20}{120} \rceil 30 = 140$
2. $20 + 90 + \lceil \frac{140}{120} \rceil 30 = 170$
3. $20 + 90 + \lceil \frac{170}{120} \rceil 30 = 170$



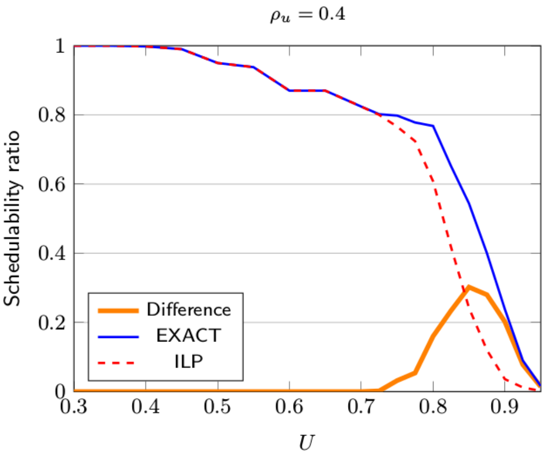
Biondi vs. Davis: Interference



Interference comparison between the two approaches[3]

[3] A. Biondi, A. Melani, M. Marinoni, M. D. Natale and G. Buttazzo, "Exact Interference of Adaptive Variable-Rate Tasks under Fixed-Priority Scheduling,"(ECRTS), Madrid, 2014

Biondi vs. Davis: Schedulability



Schedulability comparison between the two approaches from Biondi (2015)

Comparison of Experiment Results

Workload Generation

- Synthetic workloads, 500 randomly generated task sets
- $4 - 8$ AVR modes, one AVR task, $n = 5$ periodic tasks

Response Time

- Biondi's approach computes the exact response time
- Davis' approach is more pessimistic, because of quantization

Schedulability

- Biondi's test can admit ten times more task sets with processor utilization of over 80%

Conclusion

- A. Biondi, M. Di Natale, and G. Buttazzo
  @ ICCPS 2015
  Response-time analysis for real-time tasks in engine control applications
  - Exact worst-case interference of AVR tasks
  - Pruning search domain with dominant speeds

- R. I. Davis, T. Feld, V. Pollex and F. Slomka
  @ RTAS 2014
  Schedulability tests for tasks with Variable Rate-dependent Behaviour under fixed priority scheduling
  - ILP-based worst-case interference of AVR tasks
  - Quantization of speed

Further Reading

- G. Buttazzo, E. Bini, and D. Buttle @ DATE 2014
  Rate-adaptive tasks: Model, analysis, and design issues
- A. Biondi et al, @ ECRTS 2014
  Exact Interference of Adaptive Variable-Rate Tasks under Fixed-Priority Scheduling
- A. Biondi and G. Buttazzo @ DATE 2016
  Real-time analysis of engine control applications with speed estimation
- A. Biondi, M. Di Natale, and G. Buttazzo @ ICCPS 2016
  Performance-Driven Design of Engine Control Tasks
- M. Mohaqeqi et al. @ ECRTS 2017
  Refinement of Workload Models for Engine Controllers by State Space Partitioning

Response Time Algorithm Biondi et al.

Periodic Task interfered by AVR task

```
procedure ResponseTime((i, ω, Π, t))
    jobRT ← SingleJobRT;
    RTCandidates.add(jobRT);
    if jobRT > MaxTime then return ;
    end if

    dominants ← GetDominants
    for ω^next in dominants do
        ResponseTime(i, ω^next, Π^next, t^next)
    end for
end procedure
```

Schedulabilty Test: $ResponseTime(i, \omega_0, 0, 0)$, **list candidates, max of RT candidates, check** $D_i \geq R$

Schedulability Algorithm Davis et al.

procedure $\textsc{SchedulabilityTest}(i)$
    for all modes of τ_i *initialize* $R_{i,m} = 0$
        for each instantaneous engine speed S
            for each mode m *corresponding to* S
                Compute WCRT $R_{i,m}^{(S)}$ *with the current mode* m
                $R_{i,m} = max(R_{i,m}(S), R_{i,m})$
                if $R_{i,m} > D^*_{i,m}$
                    return *unschedulable*
    return *schedulable*;
end procedure

Quantization of S**...**

- Quantization is pessimistic
- May not be safe

Pessimistic Interference of AVR Tasks (Davis et al.)

Interference $I_j(w)$ of task τ_j in $[0, w]$

$$\boxed{I_j(w) = \sum_{\forall x} k_{j,x} C_{j,x}}$$

- $k_{j,x}$ is the integer number of jobs of execution mode x
- $C_{j,x}$ is the execution time of τ_j in mode x

Constraints

1. Last job release strictly before end of last window
2. Last job is released with the execution time C_j^{max}
3. $k_{j,x} \geq 0$

Maximum interference

- There is a sequence jobs of τ_j, that release the maximum interference
- Maximize $I_j(w)$