Period Optimization for Hard Real-time Distributed Automotive Systems

Presenters: Forrest Iandola and Ilge Akkaya

Abhijit Davare[1], Qi Zhu[1],
Marco Di Natale[2], Claudio Pinello[3],
Sri Kanajan[2], Alberto Sangiovanni-Vincentelli[1]

[1] EECS, UC Berkeley

[2] General Motors Research

[3] Cadence Research Labs

Big Picture

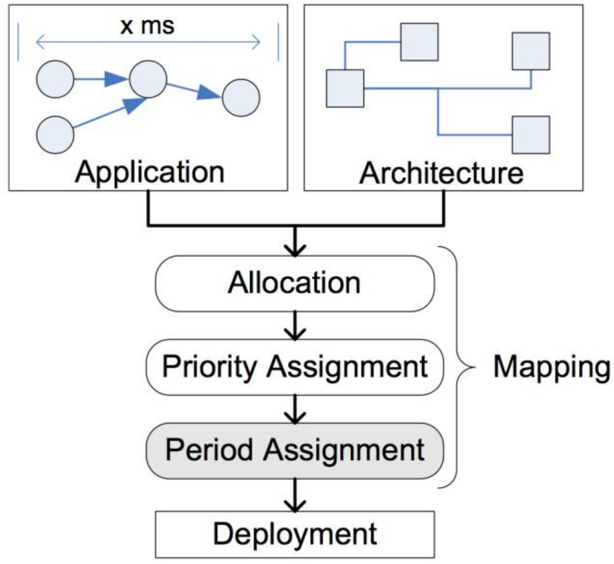


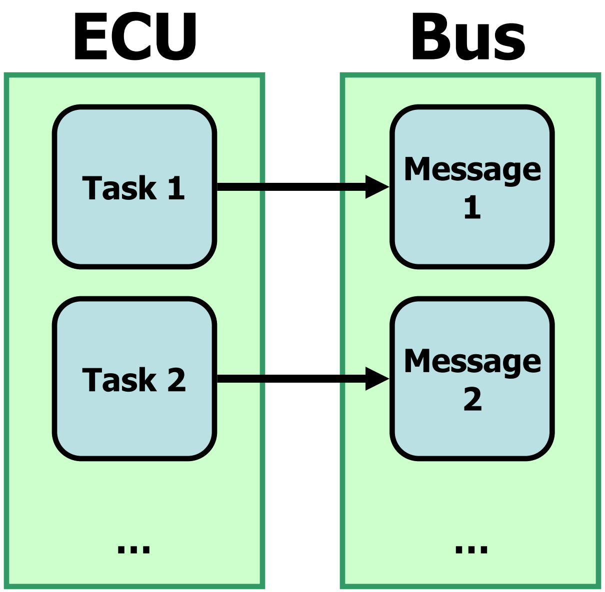
Figure 1: Period assignment within the overall design flow

- Goal: choose periods that minimize task response times

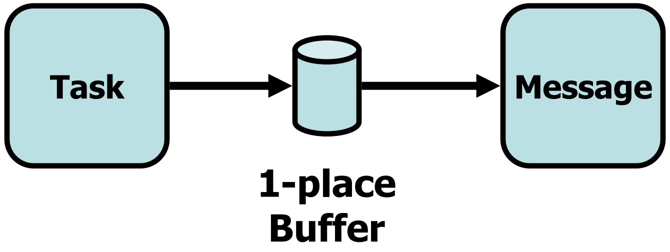
Outline

- System model
- Scheduling requirements
- Continuous approximation of response time analysis
- Formulate optimization problem to:
  - Choose periods
  - Minimize sum of task response times

System Model

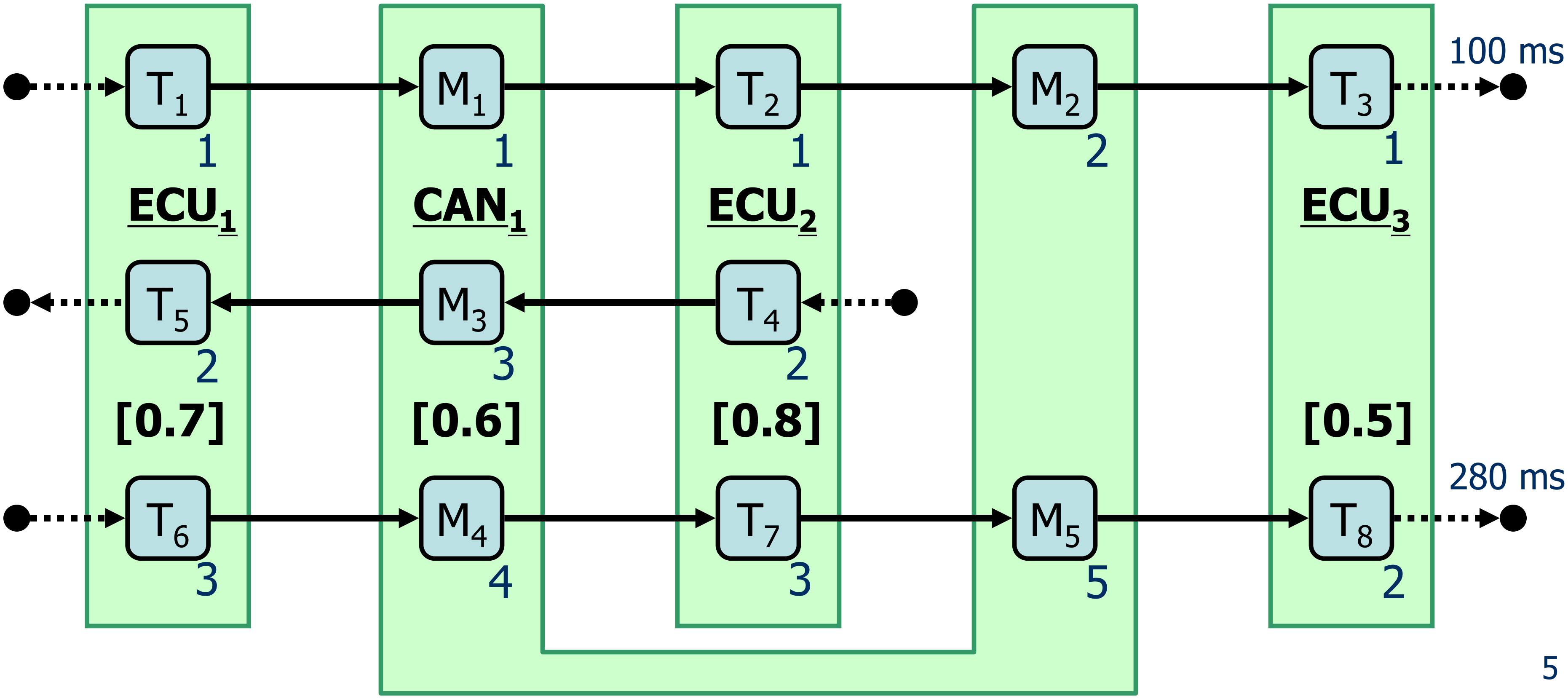


- Tasks allocated to ECUs
  - Preemptive execution
  - Scheduling: static priorities
- Messages allocated to buses
  - Non-preemptive transmission
  - Scheduling: static priorities



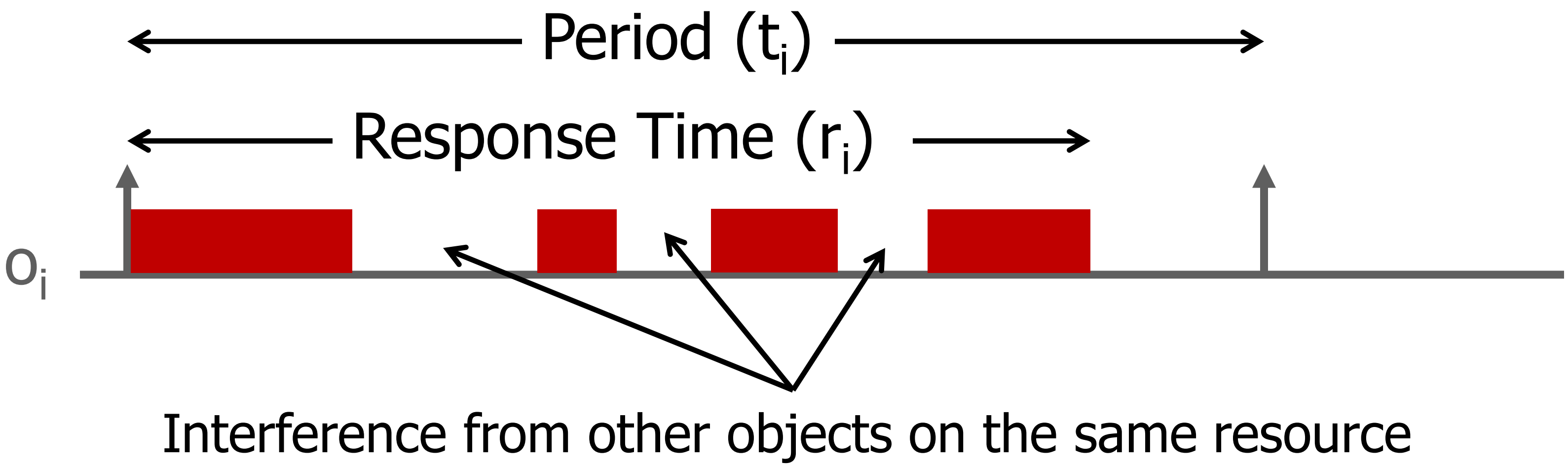
Problem Inputs

- Sets
  - Paths: $\mathcal{P}$
  - Objects: $\mathcal{O} = \mathcal{T} \cup \mathcal{M}$
  - Resources: $\mathcal{R}$



1. Object Schedulability

- Ensure that all objects are processed before their subsequent activations



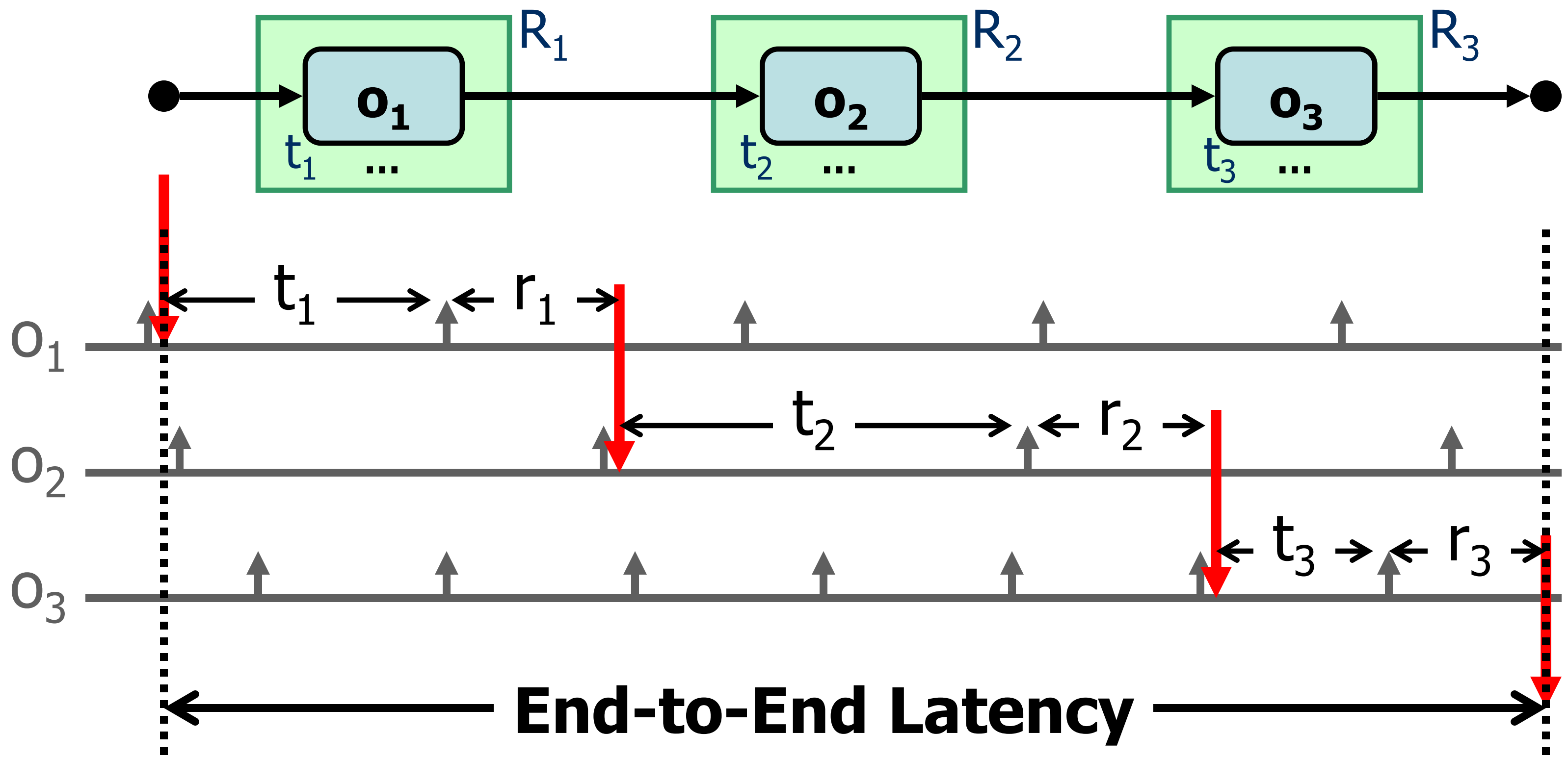
$$r_i \leq t_i \quad \forall o_i \in \mathcal{O}$$

2. Utilization Bounds

- Resource utilization
  - Fraction of time the resource (either ECU or bus) spends processing its objects (either tasks or messages)
- Utilization bounds less than 100%
  - To allow for future extensibility
- Intuition: Larger periods ➔ lower utilization

$$\left(\sum_{i:o_i \to R_j} \frac{c_i}{t_i}\right) \le u_j \quad \forall R_j \in \mathbf{R}$$

3. End-to-End Latency (minimize this)



- For each object in the path, add
  - Period (t_i)
  - Worst case response time (r_i)

$$\ell_p = \sum_{o_i \in p} t_i + r_i \qquad \forall p \in \mathcal{P}$$

Worst Case Response Times

Response time (r_i) = Processing time (c_i) + Interference time (w_i)

<u>Tasks</u>

Preemption (p_i)

$$\sum_{j \in hp(i)} z_{ij} c_j$$

$$\left\lceil \frac{r_i}{t_j} \right\rceil$$

<u>Messages</u>

Blocking (b_i) + Preemption (p_i)

$$\sum_{j \in hp(i)} z_{ij} c_j$$

$$\left\lceil \frac{r_i - c_i}{t_j} \right\rceil$$

Continuous RTA Approximation

- Getting rid of the ceiling – enables convex optimization
- Approximate the ceiling function
  - Constant parameter: $0 \leq \alpha_i \leq 1$
  - Approximated worst case response time: s_i

$$r_i = c_i + \sum_{j \in hp(i)} \left\lceil \frac{r_i}{t_j} \right\rceil c_j \qquad \forall o_i \in \mathcal{T}$$

$$\Downarrow$$

$$s_i = c_i + \sum_{j \in hp(i)} \left(\frac{s_i}{t_j} + \alpha_i \right) c_j \quad \forall o_i \in \mathcal{T}$$

Convex Optimization Formulation

- Sets
  - Paths: $\mathcal{P}$
  - Objects: $\mathcal{O} = \mathcal{T} \cup \mathcal{M}$
  - Resources: $\mathcal{R}$
- Parameters
  - Computation time: c
- Decision Variables
  - Periods: t
  - Approx. response times: s

Minimize the sum of approx. response times

$$min. \quad \sum_{o_i \in \mathcal{O}} s_i$$

Meet end-to-end latency deadlines

$$s.t. \quad \ell_p \leq d_p \quad \forall p \in \mathcal{P}$$

Approximate response times

$$\frac{\sum_{j \in hp(i)} c_j \alpha_i + c_i}{s_i} + \sum_{j \in hp(i)} \frac{c_j}{t_j} \leq 1 \quad \forall o_i \in \mathcal{T}$$

$$\frac{\sum_{j \in hp(i)} c_j \alpha_i + b_i}{s_i'} + \sum_{j \in hp(i)} \frac{c_j}{t_j} \leq 1 \quad \forall o_i \in \mathcal{M}$$

$$s_i = s_i' + c_i \quad \forall o_i \in \mathcal{M}$$

Ensure schedulability

$$\frac{s_i}{t_i} \leq 1 \quad \forall o_i \in \mathcal{O}$$

Meet utilization bounds

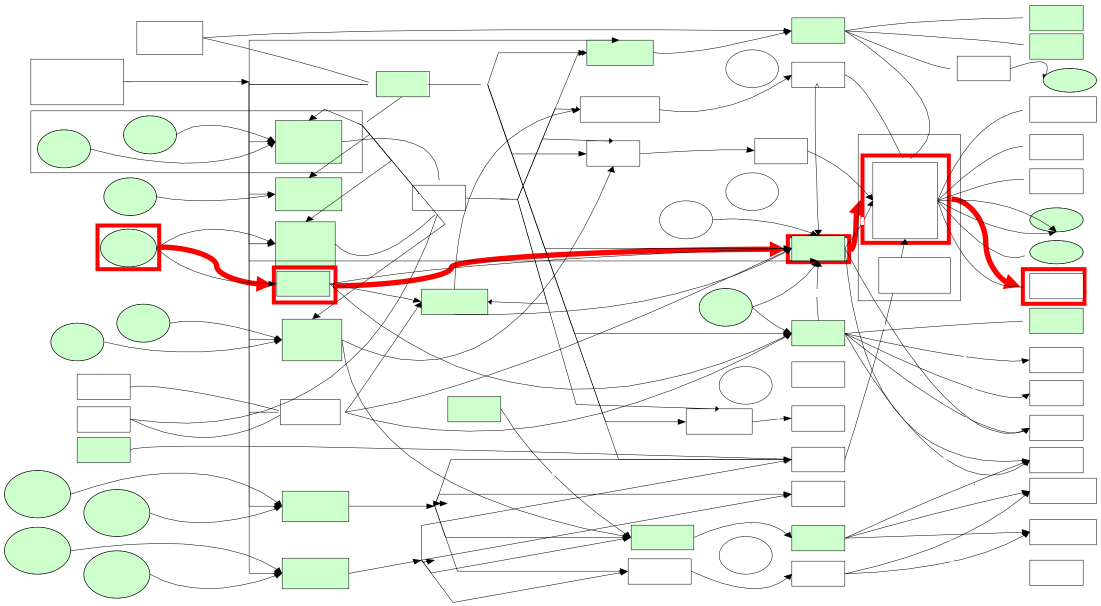
$$\sum_{o_i | R_{o_i} = j} \frac{c_i}{t_i} \leq u_j \quad \forall R_j \in \mathcal{R}$$

Lower and upper bounds for periods

$$n_i \leq t_i \leq x_i \quad \forall o_i \in \mathcal{O}$$

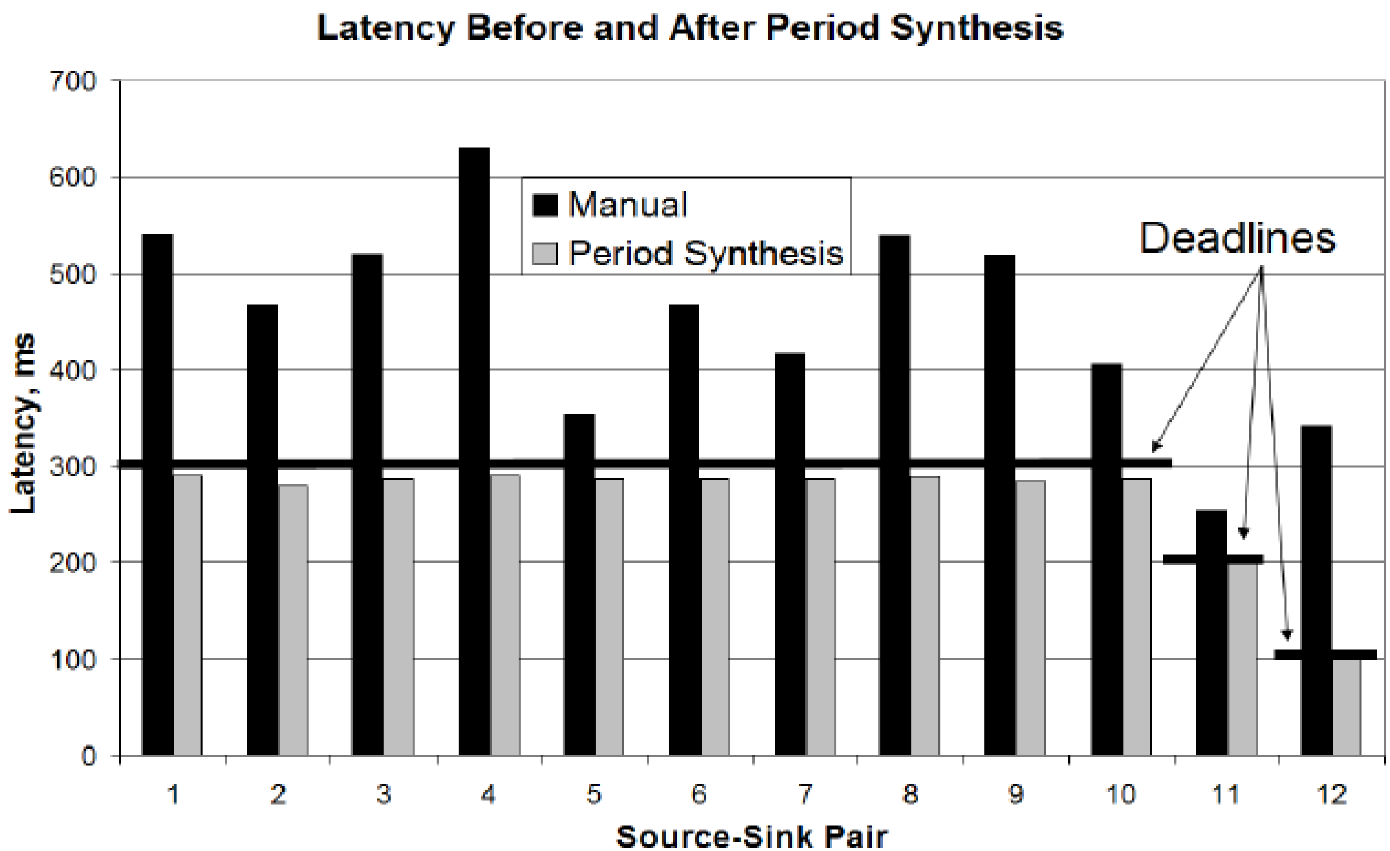
Case Study: GM Experimental Vehicle

- Functionality
  - 92 tasks
  - 196 messages
- Architecture
  - 38 ECUs
  - 4 buses



- End-to-end latency constraints
  - 12 source-sink task pairs
  - 222 total paths
  - Deadlines range from 100ms to 300ms

Experiments: Manual vs. Period Opt.



- Feasible schedule with α = 1 in 1st iteration
- Solution time: 24s on Pentium M with <1GB of RAM

Critique

- Good
  - Clear system model and scope
  - For a real system, the optimization takes only a few minutes on a cheap processor

- Future work
  - Paper only considers periodic tasks. Sporadic tasks (e.g. proportional to engine RPM) are common in automotive systems.
  - Cheap heuristics to approximate the optimization while the system is running?

Conclusions

- Described a continuous approximation of response time analysis
- Formulate optimization problem to:
  - Choose periods
  - Minimize sum of task response times
- Applied the analysis to a real system, outperformed hand-tuned schedule