Data-Age Analysis and Optimisation for Cause-Effect Chains in Automotive Control Systems

Johannes Schlatow, Mischa Möstl, Sebastian Tobuschat, Tasuku Ishigooka*
and Rolf Ernst

*Hitachi Ltd.
Multi-rate control systems

Cause-effect chains reflect sensor-to-actuator processing:

Periodic task $\tau_1$

Periodic task $\tau_2$

Periodic task $\tau_3$

Sensor

Actuator
Multi-rate control systems

Cause-effect chains reflect sensor-to-actuator processing:

Data age reduces control performance

- mainly influenced by periods and synchronisation
- response times often only contribute a small fraction
Problem

How to assign priorities, allocate tasks to processors, synchronise tasks to reduce data age?

Automotive approach: rate-monotonic scheduling (RMS)
- only covers priority assignment
- optimises schedulability, not data age of an entire chain

Research approach (related work):
- priority assignment and processor allocation is an old issue
- existing work focuses on response times, not on cause-effect chains
System model

periodic tasks $\tau_x$ with
- harmonic periods $T_x$
- activation offset $\Phi_x$
- static priorities (preemptive)
- read on start, write on completion
- constrained deadlines $D_x$

\[
\begin{align*}
&\text{sync.} \\
&\tau_i \\
&\tau_j
\end{align*}
\]

$\tau_i$ and $\tau_j$ represent periodic tasks with harmonic periods $T_i$ and $T_j$, activation offsets $\Phi_i$ and $\Phi_j$, and deadlines $D_i(0)$ and $D_j(0)$. The diagram illustrates the timing relationships between these tasks, including their activation times and completion times.
Problem statement

Given

- task set including periods and worst-case/best-case execution times ($C_i^+ / C_i^-$)
- set of cause-effect chains (sequence of tasks)
- number of equal processors

Wanted

- priority assignment
- task-to-processor allocation
- activation offsets

Subject to (feasibility)

- satisfy response-time constraints (deadlines)

Subject to (optimality)

- minimise sum of data-age latencies for given chains
Outline

- System model and problem statement
- Response-time and data-age analysis
- Optimisation with mixed integer linear programming (MILP)
- Experiments
Apply RTA from [Tindell & Clark 1994] with modification to consider of offsets:

- worst-case response time
  \[ R_i^+ = WCET C_i^+ + \text{interference from higher-priority tasks} \]

Unless:
shorter period and activated after \( \tau_i \) latest completion

\( \tau_j \quad \Phi_j \quad T_j \quad \Phi_j \quad 2T_j \)

\( \tau_i \quad \Phi_i \quad T_i \)
Data-age analysis

Goal

- find the worst case w.r.t. data age among a sequence of reads and writes \((r_1, w_1, r_2, w_2, r_3, w_3)\)

Compositional approach [cf. Feiertag et al.] + offsets:

- sum of read-write delays \((R_x^+)\) + write-read delays \((d_{x,y}^+)\)
Write-read delay in case of **oversampling** $T_3 < T_2$

Worst-case write-read delay: 

$$d_{2,3}^+ \geq \max_{n \geq 0} (r_3^+(n) - w_2^-(0))$$

if $w_2^+(0) > r_3^-(n)$:

$$d_{2,3}^+ \geq \max_{n \geq 0} (r_3^+(n) - w_2^-(-1))$$

Remark: upper/lower bounds $r_x^\pm$ and $w_x^\pm$ to consider jitter
Write-read delay in case of undersampling $T_2 \geq T_1$

Worst-case write-read delay:

$$d_{1,2}^+ \geq \max_{n \geq 0} \left( r_2^+(0) - w_1^-(n) \right)$$
Write-read delay in case of **undersampling** $T_2 \geq T_1$

**Worst-case write-read delay:**

$$d_{1,2}^+ \geq \max_{n \geq 0} \left( r_{2}^+ (0) - w_1^- (n) \right)$$

*if* $w_1^+ (0) > r_2^- (0)$:

$$d_{1,2}^+ \geq r_{2}^+ (0) - w_1^- (-1)$$
MILP formulation

Goal: Transform formulas into linear constraints of the form:

\[ y \geq A \cdot x_1 + B \cdot x_2 + C \cdot x_3 + \ldots \]

where A, B, C constants.

RTA derived from [Wieder & Brandenburg 2013]

- with slight modifications to consider synchronisation/offsets
- details: see paper

Data-age analysis:

- additional variables: \( r_j^\pm, w_i^\pm, d_{i,j}^+ \)
- formulate constraints “\( d_{i,j}^+ \geq \ldots “ \) (for every pair of jobs within hyperperiod) using binary helper variables for conditionals
- details: see paper
Experiments

Case study 1:
- ADAS use case by Hitachi

Case study 2:
- Engine-control benchmark by Bosch (cf. WATERS challenge)

Implementation:
- reference implementation of RTA and data-age analysis in pyCPA
  - [https://bitbucket.org/pycpa](https://bitbucket.org/pycpa)
- MILP implementation with ZIMPL [Koch 2004], Solver: SCIP (primal-dual)
  - [https://www.ida.ing.tu-bs.de/pub2018/schlatow2018dataage.zip](https://www.ida.ing.tu-bs.de/pub2018/schlatow2018dataage.zip)

Remark:
- ISRs are modelled by non-harmonic periods
  (no synchronisation → conservative bound)
1a) Original mapping:
- two (overlapping) chains
- ISR: $T=550\mu s$
- Task E, G: $T=10ms$
- Task H, C: $T=50ms$
- sync. + priority assignment:
  $$\Rightarrow \text{data age} \approx 73ms$$

1b) Optimised mapping:
- sync., priority assignment: + allocation
  $$\Rightarrow \text{data age} \approx 12ms$$
2a) Original mapping:

- **two chains**
  - 1) 100ms, 10ms, 2ms
  - 2) ISR10, 2ms, 50ms
- **data age results**
  - 1) \(\sim 160\text{ms}\)
  - 2) \(\sim 5.2\text{ms}\)
Case study #2 – engine-control benchmark

2b) Optimised mapping:
- data age results
  - 1) ~134ms (before: 160ms)
  - 2) ~4.7ms (before: 5.2ms)
Solving time

Primal-dual algorithm calculates “gap” to optimum
→ solving can be aborted once a good enough solution was found

<table>
<thead>
<tr>
<th>case study</th>
<th>1a</th>
<th>1b</th>
<th>2a</th>
<th>2b</th>
</tr>
</thead>
<tbody>
<tr>
<td>time (feasibility)</td>
<td>14s</td>
<td>17s</td>
<td>0.8s</td>
<td>431s</td>
</tr>
<tr>
<td>time (10% gap)</td>
<td>29s</td>
<td>236s</td>
<td>126s</td>
<td>1528s</td>
</tr>
<tr>
<td>time (optimality)</td>
<td>29s</td>
<td>377s</td>
<td>&gt;127h</td>
<td>&gt;280h</td>
</tr>
<tr>
<td>#solutions</td>
<td>7</td>
<td>118</td>
<td>&gt; 189</td>
<td>&gt; 12788</td>
</tr>
</tbody>
</table>
Solving progress

10 randomised variations of 1b) with different system load:

- gap to optimum closes quite quickly ($\rightarrow$ practicality)
- tendency to be slower for higher load
Conclusion

Results

▪ use offset synchronization, priority assignment and processor allocation of harmonic tasks to reduce/optimise data age latency
▪ compositional data-age analysis suits MILP approach
▪ good solutions are found in short time (10% gap)
▪ optimal solutions not necessarily found in limited time
  ▪ problems: indeterminism, number of solutions
▪ improvement: MILP efficiency (e.g. elimination of integer variables)