Budgeting Under-specified Tasks for Weakly-Hard Real-Time Systems

Zain A. H. Hammadeh, Sophie Quinton, Marco Panunzio, Rafik Henia, Laurent Rioux, and Rolf Ernst
Motivational example: satellite on board software (TAS) – 1

- satellite **On-Board SoftWare** (OBSW)
  - payload: main satellite mission
  - **platform**: governs the satellite

- The platform is treated as **hard real-time system**

- Some tasks *may* occasionally *miss deadlines*
  - *without* dreadful consequences on the mission

It might be worthy to relax the constraints to weakly-hard constraints (accept limited number of deadline misses)
Motivational example: satellite on board software (TAS) – 2

- Two different kinds of tasks:
  - **nominal** tasks: active in the represented operational scenario
  - **recovery** tasks: triggered only on given fault/error occurrences
    - **Under-specified**

The specification of recovery tasks typically occurs in the **latest** development phases

- specified: **priorities**
- unspecified: load budget
  - **execution times** and **activation patterns**
Overview

**Problem formulization:** provide a set of *constraints* on the *execution times* and the *activation patterns* of the under-specified tasks to guarantee *(m,k)-schedulability* of all nominal tasks

- (m, k)-schedulable: no more than *m* deadline misses *out-of* a sequence of *k* consecutive executions

**Contribution:**

- *budgeting* the under-specified tasks based on
  - *hard* real-time constraints (SotA)
  - *weakly-hard* real-time constraints
  - *multiframe* task model [Mok97]

- a *case study* dealing with satellite on-board software and *synthetic* test cases
Outline

• System model
• Budgeting under-specified tasks
• Experimental results
• Conclusion
Outline

• System model
  • Budgeting under-specified tasks
• Experimental results
• Conclusion
System model

System
- *single-core* processor
- Fixed Priority Preemptive scheduling policy *FPP*
- *independent* tasks
- nominal tasks: fully specified
- under-specified tasks

Nominal task $\tau_i$
- minimum distance $\delta_i$
- worst case execution time $C_i$
- priority $\pi_i$
- constrained deadline $D_i$ ($D_i \leq \delta_i$)
- (m,k) constraint: *m* deadline misses *out-of k* consecutive executions

Under-specified task $\tau_u$
- priority $\pi_u$
Outline

- System model
- **Budgeting under-specified tasks**
  - with hard real-time constraints
  - with weakly-hard real-time constraints
  - with multiframe execution time model
- Experimental results
- Conclusion
Budgeting with hard real-time constraints

Available slack $S_4^0 = 4 \Rightarrow C_u = 4$
Budgeting with hard real-time constraints

Available slack $S_4^0 = 4 \Rightarrow C_u = 4$
The **slack** $S_i^0$ is the maximum amount of processing time which may be stolen from any job of $\tau_i$ *without* causing its deadline to be *missed*.

The execution time of an under-specified task is *bounded* by $S_i^0$.

Available slack $S_4^0 = 4 \Rightarrow C_u = 4$
The **slack** $S_i^0$ is the maximum amount of processing time which may be stolen from any job of $\tau_i$ *without* causing its deadline to be *missed*.

The execution time of an under-specified task is *bounded* by $S_i^0$.

Available slack $S_4^0 = 4 \Rightarrow C_u = 4$
The **slack** $S_i^0$ is the maximum amount of processing time which may be stolen from any job of $\tau_i$ *without* causing its deadline to be *missed*

The execution time of an under-specified task is **bounded** by $S_i^0$

*If the execution time of an under-specified task is bounded by $S_i^0$ and $\delta_u > D_i$ then $\tau_i$ is schedulable*
Weakly-hard slack

Available slack = 5 \Rightarrow C_u = 11
Weakly-hard slack

Available slack = 5 \Rightarrow C_u = 11
The **weakly-hard slack** $S_i^{\mu}$ is the maximum amount of processing time which may be stolen from $\tau_i$ within a window of $(\mu - 1)\delta_i + D_i$ **without** causing more than $\mu$ deadlines of $\tau_i$ to be missed in a row.
Weakly-hard slack

The **weakly-hard slack** $S_i^\mu$ is the maximum amount of processing time which may be stolen from $\tau_i$ within a window of $(\mu - 1)\delta_i + D_i$ **without** causing more than $\mu$ deadlines of $\tau_i$ to be missed in a row.

Could we use $S_i^\mu$ to bound the execution time of under-specified tasks as we did for hard real-time constraints?
Budgeting with weakly-hard constraints

\[ C_{u_1} + C_{u_2} = S_4^1 = 11 \]
\[ C_u = 6 > S_4^0, C_{u_2} = 5 > S_4^0 \]

\[ \tau_4: (m, k) = (1, 5) \]
Budgeting with weakly-hard constraints

\[ C_{u_1} + C_{u_2} = S_{4}^1 = 11 \]
\[ C_u = 6 > S_{4}^0, C_{u_2} = 5 > S_{4}^0 \]

\[ \tau_4: (m, k) = (1, 5) \]
Budgeting with weakly-hard constraints

\[ C_{u_1} + C_{u_2} = S_4^1 = 11 \]
\[ C_u = 6 > S_4^0, C_{u_2} = 5 > S_4^0 \]

\[ \tau_4: (m, k) = (1, 5) \]

\[ C_{u_1} + C_u = (m + 1)S_4^0 = 8 \]
\[ C_{u_1} = 5 > S_4^0, C_{u_2} = 3 < S_4^0 \]

\[ \delta_u \]
Budgeting with weakly-hard constraints

\[ C_{u_1} + C_{u_2} = S_4^1 = 11 \]
\[ C_u = 6 > S_4^0, C_{u_2} = 5 > S_4^0 \]

\[ \tau_4: (m, k) = (1, 5) \]

\[ C_{u_1} + C_u = (m + 1)S_4^0 = 8 \]
\[ C_{u_1} = 5 > S_4^0, C_{u_2} = 3 < S_4^0 \]
Budgeting with weakly-hard constraints

\[ C_{u_1} + C_{u_2} = S_4^1 = 11 \]
\[ C_u = 6 > S_4^0, C_{u_2} = 5 > S_4^0 \]

\[ \tau_4: (m, k) = (1, 5) \]

\[ C_{u_1} + C_u = (m + 1)S_4^0 = 8 \]
\[ C_{u_1} = 5 > S_4^0, C_{u_2} = 3 < S_4^0 \]

Proof is in the paper
If the execution time of an under-specified task is bounded by 

$$(m + 1)S_i^0$$

and $\delta_u > \Delta_i^k$ then $\tau_i$ is $(m,k)$-schedulable
**Budgeting with weakly-hard constraints**

If the execution time of an under-specified task is bounded by \((m + 1)S^0_i\) and \(\delta_u > \Delta^k_i\) then \(\tau_i\) is \((m,k)\)-schedulable.

**Budgeting under-specified tasks**

Hard real-time constraints:
\[
\sum_{u \in U} C_u \leq S^0_i \quad \text{and} \quad \delta_u > D_i
\]

Weakly-hard real-time constraints:
\[
\sum_{u \in U} C_u \leq (m + 1)S^0_i \quad \text{and} \quad \delta_u > \Delta^k_i
\]
Multiframe task model

- Specific application scenario
  - A frequent *monitoring activity* with a relatively short execution time
  - A less frequent *failure recovery* activity which requires a longer execution time

- Multiframe task model: it assigns to each periodic task \( N \) execution times as a sequence \((C^0, C^1, \ldots, C^N)\)

- The execution time model of any under-specified task is \((C^s_u, C^l_u)\):
  - \(C^s_u\) is the short execution time
  - \(C^l_u\) is the long execution time
  - \((C^s_u, C^l_u)\) is not a sequence
Multiframe task model

Assumptions on multiframe model

- A frequent monitoring activity with a relatively short execution time
- A less frequent failure recovery activity which requires a longer execution time

- **Assumption 1.** Only one execution of $\tau_u$ within the window of $\Delta_i^k$ is long $C_u^l$

- An execution of $\tau_i$ may get an interference from a combination $\bar{c}$ of executions of under-specified tasks
  
  $\bar{c} = (c_1, c_2, \ldots, c_{|U|})$ where $c_u = C_u^l$, $c_u = C_u^s$

- **Assumption 2.** Any combination of $C_u^s$ causes no deadline misses

- The execution time model of any under-specified task is $(C_u^s, C_u^l)$:
  
  - $C_u^s$ is the short execution time
  - $C_u^l$ is the long execution time
  - $(C_u^s, C_u^l)$ is not a sequence
Budgeting with multiframe model – 1

\( \tau_4: (m, k) = (2, k) \)

- **Assumption 1.** only one instance has one \( C_u^l \)
- **Assumption 2.** Any combination of \( C_u^s \) causes no deadline misses

\[ \downarrow C_u^s \downarrow C_u^l \]

\[ \tau_u1 \]

\[ \tau_u2 \]

\[ \tau_4 \]

\[ \Delta^k_4 \]
Budgeting with multiframe model – 1

\( \tau_4: (m, k) = (2, k) \)

- **Assumption 1.** only one instance has one \( C_u^l \)
- **Assumption 2.** Any combination of \( C_u^s \) causes no deadline misses

\[ \begin{align*}
\tau_u^1 \downarrow C_u^s \downarrow C_u^l \\
\tau_u^2 \downarrow \downarrow \\
\tau_4 \downarrow \downarrow \downarrow \downarrow \downarrow \ \\
\Delta^k_4 \\
\end{align*} \]

Unschedulable combination

Schedulable combinations
Budgeting with multiframe model – 1

\( \tau_4: (m, k) = (2, k) \)

- **Assumption 1.** only one instance has one \( C_u^l \)
- **Assumption 2.** Any combination of \( C_u^s \) causes no deadline misses

\[ \downarrow C_u^s \downarrow C_u^l \]

Unschedulable combination

Gang

Schedulable combinations

\( \tau_u^1 \)

\( \tau_u^2 \)

\( \tau_4 \)

\( \Delta_{4}^{k} \)

\textit{Gang} \( \mathcal{G} \) : is a set of combinations which contain at least one long execution time

\( \mathcal{G}_{\tau_i} \) is the set of all gangs related to \( \tau_i \)
Let $\mu_c$ denote the **maximum** number of deadline misses which may be caused by a combination $\bar{c}$

- If $\forall G \in \mathcal{G}_i: \sum_{c \in G} \mu_c \leq m$ then $\tau_i$ is $(m,k)$-schedulable
- $\sum_{u \in \bar{c}} c_u \leq S_i^{\mu_c}$ where $c_u = C^l_u, c_u = C^S_u$

**Gang $G$:** is a **set** of combinations which contain at least one long execution time

$\mathcal{G}_i$ is the set of all gangs related to $\tau_i$
Budgeting with multiframe model – 2

\( \tau_4: (m, k) = (2, k) \)

\[ \mu_c \text{ Constraints} \]

\begin{align*}
1. \mu_{c_1} + \mu_{c_2} & \leq 2 \\
2. \mu_{c_1} + \mu_{c_4} & \leq 2 \\
3. \mu_{c_2} + \mu_{c_3} & \leq 2 \\
4. \mu_{c_3} + \mu_{c_4} & \leq 2 \\
5. \mu_{c_5} & \leq 2 \\
6. \mu_{c_1} & \leq \mu_{c_3} \\
7. \mu_{c_3} & \leq \mu_{c_5} \\
8. \mu_{c_2} & \leq \mu_{c_4} \\
9. \mu_{c_4} & \leq \mu_{c_5}
\end{align*}

\( \forall G \in \mathcal{G}_i: \sum_{c \in G} \mu_c \leq m \)
Budgeting with multiframe model – 2

\textbf{\(\mu_c\) Constraints}

1. \(\mu_{c_1} + \mu_{c_2} \leq 2\)
2. \(\mu_{c_1} + \mu_{c_4} \leq 2\)
3. \(\mu_{c_2} + \mu_{c_3} \leq 2\)
4. \(\mu_{c_3} + \mu_{c_4} \leq 2\)
5. \(\mu_{c_5} \leq 2\)

6. \(\mu_{c_1} \leq \mu_{c_3}\)
7. \(\mu_{c_3} \leq \mu_{c_5}\)
8. \(\mu_{c_2} \leq \mu_{c_4}\)
9. \(\mu_{c_4} \leq \mu_{c_5}\)

\(\forall G \in G_t: \sum_{c \in G} \mu_c \leq m\)

\(\tau_4: (m, k) = (2, k)\)

#Gangs
Budgeting with multiframe model – 2

**μ_c Constraints**

1. $\mu_{c_1} + \mu_{c_2} \leq 2$
2. $\mu_{c_1} + \mu_{c_4} \leq 2$
3. $\mu_{c_2} + \mu_{c_3} \leq 2$
4. $\mu_{c_3} + \mu_{c_4} \leq 2$
5. $\mu_{c_5} \leq 2$

6. $\mu_{c_1} \leq \mu_{c_3}$
7. $\mu_{c_3} \leq \mu_{c_5}$
8. $\mu_{c_2} \leq \mu_{c_4}$
9. $\mu_{c_4} \leq \mu_{c_5}$

∀ $G \in \mathcal{G}_i$: $\sum_{c \in G} \mu_c \leq m$

**C_r^l, C_r^s Constraints**

1. $C_1^l \leq S_1^1$
2. $C_2^l \leq S_1^1$
3. $C_1^l + C_2^s \leq S_1^1$

4. $C_1^s + C_2^l \leq S_1^1$
5. $C_1^l + C_2^l \leq S_2^2$
6. $C_1^s + C_2^s \leq S_1^0$

$\sum_{u \in c} c_u \leq S_i^{\mu_{c}}$
Budgeting with multiframe model – 2

μ\_c Constraints

1. \( \mu_{c_1} + \mu_{c_2} \leq 2 \)
2. \( \mu_{c_1} + \mu_{c_4} \leq 2 \)
3. \( \mu_{c_2} + \mu_{c_3} \leq 2 \)
4. \( \mu_{c_3} + \mu_{c_4} \leq 2 \)
5. \( \mu_{c_5} \leq 2 \)

6. \( \mu_{c_1} \leq \mu_{c_3} \)
7. \( \mu_{c_3} \leq \mu_{c_5} \)
8. \( \mu_{c_2} \leq \mu_{c_4} \)
9. \( \mu_{c_4} \leq \mu_{c_5} \)

\( \forall G \in G_t: \sum_{c \in G} \mu_{\bar{c}} \leq m \)

τ\_4: (m, k) = (2, k)

#Gangs

C\_r^l, C\_r^s Constraints

1. \( C_1^l \leq S_i^1 \)
2. \( C_2^l \leq S_i^1 \)
3. \( C_1^l + C_2^s \leq S_i^1 \)
4. \( C_1^s + C_2^l \leq S_i^1 \)
5. \( C_1^l + C_2^l \leq S_i^2 \)
6. \( C_1^s + C_2^s \leq S_i^0 \)

\( \sum_{u \in \bar{c}} c_u \leq S_i^{\mu_{\bar{c}}} \)

#Combinations
Budgeting under-specified tasks

Hard real-time constraints: \( \sum_{u \in \mathcal{U}} C_u \leq S_i^0 \)

\( \tau_4: (m, k) = (2, 10) \)

Weakly-hard real-time constraints: \( \sum_{u \in \mathcal{U}} C_u \leq (m + 1)S_i^0 \)

Multiframe task model: \( C_u^s, C_u^l \)
Outline

• System model
• Budgeting under specified tasks
• Experimental results
  ➢ the OBSW case study
  ➢ synthetic examples
• Conclusion
The OBSW satellite case study (TAS)

- **Single-core** processor
- **FPP** scheduling
- 30 tasks
  - 27 nominal
  - 3 **recovery** (under-specified)
- **Nominal tasks** are currently **analyzed** with **hard** real-time techniques
- By **experience**, the system is **robust** to occasional deadline misses
  - no **formal** bounds on the allowed deadline misses for lower priority tasks
  - **synthetic** (m,k) constraints of tasks (in the paper)
The OBSW case study (TAS) – results

- The **worst-case** response time analysis of the nominal mode shows that the system is **schedulable**.
- Our goal is to synthesize a **load budget** for the under-specified tasks $\tau_{10}, \tau_{11}, \tau_{21}$. 
The OBSW case study (TAS) – results

- The **worst-case** response time analysis of the nominal mode shows that the system is **schedulable**
- Our goal is to synthesize a **load budget** for the under-specified tasks $\tau_{10}, \tau_{11}, \tau_{21}$

Budgeting with **hard** real-time constraints

$$C_{10} + C_{11} + C_{21} = 48.01 \text{ ms}$$
The OBSW case study (TAS) – results

- The **worst-case** response time analysis of the nominal mode shows that the system is **schedulable**
- Our goal is to synthesize a **load budget** for the under-specified tasks \( \tau_{10}, \tau_{11}, \tau_{21} \)

Budgeting with **hard** real-time constraints
\[
C_{10} + C_{11} + C_{21} = 48.01 \text{ ms}
\]

Budgeting with **weakly-hard** real-time constraints
\[
C_{10} + C_{11} + C_{21} = 96.02 \text{ ms}
\]
The OBSW case study (TAS) – results

- The **worst-case** response time analysis of the nominal mode shows that the system is **schedulable**
- Our goal is to synthesize a **load budget** for the under-specified tasks $\tau_{10}, \tau_{11}, \tau_{21}$

Budgeting with **hard** real-time constraints

$$C_{10} + C_{11} + C_{21} = 48.01 \, ms$$

Budgeting with **weakly-hard** real-time constraints

$$C_{10} + C_{11} + C_{21} = 96.02 \, ms$$

Budgeting for **multiframe** tasks

$$C_{10}^l = C_{11}^l = C_{21}^l = 24.005$$

$$C_{10}^s = C_{11}^s = C_{21}^s = 12.0025$$
Synthetic examples

- **1000** task sets randomly generated depending on *UUniFast*
- Utilization $\in \{0.4, 0.5, 0.6, 0.7, 0.8\}$
- A set of *nominal* tasks $\mathcal{T}$
  - number of nominal tasks $\in [1, 20]$
  - $k \in [2, 100]$, $m \in [1, k - 1]$
- Under-specified tasks $|\mathcal{U}| \in \{1, 2, 3\}$
- We focus on the load budget
  - **metric**: $\text{load}_{MF}/\text{load}_H$
Synthetic examples: results – 1

- A histogram shows how much we gain in terms of load budget

\[
load_{MF} \geq load_{H}
\]

![Histogram showing load budget gain](image)
The larger the number of under-specified tasks the less load we gain
• sharing the available slack among more under-specified tasks
Outline

• System model
• Budgeting under specified tasks
• Experimental results
• Conclusion
Conclusion

• We have shown how to budget under-specified tasks in the early design of weakly-hard real-time systems by providing sufficient conditions which guarantee (m, k)-schedulability.

• We budget them based on:
  ➢ **hard** real-time constraints
  ➢ **weakly-hard** real-time constraints
  ➢ **multiframe** task model

• We show a case study dealing with satellite on-board software.
  • **recovery** tasks are under-specified.

• Our analysis is validated on synthetic test cases.

• We have not at all addressed the issue of the running time of the analysis.
Conclusion

• We have shown how to budget under-specified tasks in the early design of weakly-hard real-time systems by providing sufficient conditions which guarantee \((m, k)\)-schedulability.

• We budget them based on:
  - hard real-time constraints
  - weakly-hard real-time constraints
  - multiframe task model

• We show a case study dealing with satellite on-board software recovery tasks are under-specified.

• Our analysis is validated on synthetic test cases.

• We have not at all addressed the issue of the running time of the analysis.

Thank you!

Questions?
Methodology

How to dimension the tasks that are still under-specified in the system:

1. Execution time budgeting with **hard real-time constraints**
   - acceptable for the architect?

2. Execution time budgeting with **weakly-hard real-time constraints**
   - larger execution time budget, *longer* minimum distance

3. Are **activation patterns** of the under-specified tasks known? is **Multiframe** execution time model meaningful?
   - more relaxed bounds on execution time budgets.