Rhythmic Tasks: A New Task Model with Continually Varying Periods for Cyber-Physical Systems

Junsung Kim, Karthik Lakshmanan and Raj Rajkumar

Electrical and Computer Engineering
Carnegie Mellon University
Outline

- Motivation
- Rhythmic Tasks
- Case Studies
- Related Work
- Conclusion
Motivation → Rhythmic Tasks → Case Studies → Related Work → Conclusion

Engine Control Module Operations

- The computation repeats ‘periodically’
- The task is triggered by the physical piston location, which is governed by engine speed
- Engine speed is not constant

How to model and analyze such tasks with cyber-physical constraints?
Self-driving Cars

- GM Chevy Tahoe named “Boss”
- Won 2007 DARPA urban challenge
Self-driving Cars

**Boss**
- Senses environment
- Fuses sensor data to form a model of the real world
- Plans navigation paths
- Actuates steering wheel, brake, and accelerator

**Boss requires**
- Safety-critical operations
- Timing guarantees
- Robustness to harsh environments
- Dealing with dynamic circumstances
An Example Scenario: Sensor Recovery

Motivation → Rhythmic Tasks → Case Studies → Related Work → Conclusion

Lidar fails...
Video Image Used

- Radar data ranging from 0m to 40m should be analyzed.
- Radar data handler will consume more execution time.
- Video data can be used:
  - Point Grey Firefly Camera
    - 320x240 at 112FPS
    - 752x480 at 60FPS
    - 1328x1048 at 23FPS
Outline

- Motivation
- Rhythmic Tasks
- Case Study
- Related Work
- Conclusion
Observations

- The period of the engine control task can vary very dramatically
  - from a few hundred RPM to several thousand RPM
- The corresponding worst-case execution time (WCET) can also vary based on
  - RPM
  - Number of active cylinders
  - Gear ratio
  - The amount of fuel injected into active cylinders
Rhythmic Tasks
Tasks with Parameters dependent on CPS State

CPS State
- Represented by $v_s$, an $m$-dimensional vector that denotes the state of a CPS
  - e.g. $v_s$: [RPM, Number of Active Cylinders, The amount of fuel injected to cylinders, Gear ratio]
  - e.g. $v_s$: [fps, resolution]

Rhythmic Task
- $T(v_s)$: Period for a given CPS state $v_s$
- $C(v_s)$: CPU demand for a given $v_s$
- $U(v_s) = C(v_s)/T(v_s)$: Utilization of a rhythmic task
Rhythmic Tasks: Assumptions

- Given \( n \) tasks, there are \( m \) rhythmic tasks and \((n - m)\) periodic tasks.
- In this paper, we assume \( m = 1 \) (i.e. one rhythmic task, \( \tau^*_1 \)).
  - \( T^*_1 \): the period of the rhythmic task.
  - \( C^*_1 \): the worst-case execution time of the rhythmic task.
- We also assume that
  - \( T_{1,\text{min}} \leq C^*_1 \leq T_{1,\text{max}} \leq T_2 \), i.e. \( \tau^*_1 \) is the highest-priority task.
- Rhythmic task classifications
  - **Constant Computation Rhythmic Task**
    - \( C^*_1 \) is constant.
  - **Constant Utilization Rhythmic Task**
    - \( U_1 = C^*_1 / T^*_1 \) is constant.
  - **General Rhythmic Task**
    - \( C^*_1 = f(v_s) \).
Rhythmic Tasks: Questions

- Steady-state analysis
  - Given a set of periodic tasks, what is the feasible $C_1^*$?
  - How does schedulable utilization change with $C_1^*$?

- “Acceleration” analysis
  - How much can we accelerate? i.e. rate of decrease in $T_1^*$
  - Does the traditional response time test work?

- “Deceleration” analysis
  - Can we even decelerate? i.e. rate of increase in $T_1^*$

We provide initial answers to these questions.
One Rhythmic Task and One Periodic Task
The Maximum Possible $C_1$

Lemma 1: Given one rhythmic task $\tau_1$, represented by $(C_1, T_1)$, and one lower-priority periodic task $\tau_2$, represented by $(C_2, T_2)$, both tasks are schedulable by RMS if the following inequality is satisfied.

$$C_1 \leq \max \left( \frac{T_2 - C_2}{\lceil T_2/T_1 \rceil}, T_1 - \frac{C_2}{\lceil T_2/T_1 \rceil} \right)$$
Outline of the Proof

■ Case 1: \( C_2 \) completes at \( T_2 \)

The worst-case response time of \( \tau_2 \)

\[
\begin{align*}
C_1^* & \quad C_2 & \quad C_1^* & \quad C_2 & \quad C_1^* & \quad C_2 \\
\hline
(1) & \quad T_1 & \quad T_1 & \quad T_2 \\
\end{align*}
\]

■ Case 2: \( C_2 \) completes before \( T_2 \)

The worst-case response time of \( \tau_2 \)

\[
\begin{align*}
C_1^* & \quad C_2 & \quad C_1^* & \quad C_2 & \quad C_1^* & \quad C_2 \\
\hline
(2) & \quad T_1 & \quad T_1 & \quad T_2 \\
\end{align*}
\]

■ Find the maximum value of these two cases.
A Rhythmic Task with a Periodic Task

- Given $\tau_2$ with $(C_2 = 6, T_2 = 14)$
- Two interesting sets of points
  - Harmonic points (both task periods are harmonic)
  - Flexion points (task periods are antagonistic)
Utilization Variations as a function of $T_1$ and $C_2$
Relations among Variables

- \( T_1 = \frac{T_2}{i} \), where \( i \) is an integer, leads to harmonic points when the total utilization of the two tasks is maximum (1).
Relations among Variables

- Minimum $U$ points occur when $T_1^* = \frac{C_2}{i(i-1)} + \frac{T_2}{i}$ and

$$C_1^* = \frac{T_2-C_2}{i}$$
“Acceleration”

- **Period decreases after every instance**
  - \( \alpha \): acceleration ratio
    - \( \alpha = 0.5 \rightarrow \) period halves after each instance
  - \( n_\alpha \): the maximum acceleration duration

- **Acceleration example**
  - \( \tau_2 \): (6,14) is a normal periodic task
  - \( \tau_1^* \): (2,5) at 0 with \( \alpha = 0.3 \) and \( n_\alpha = 1 \) as a Constant Computation Rhythmic Task (CCRT)

- \( \tau_1^* \)'s new period starts
- \( \tau_2 \) meets its deadline.
Acceleration Analysis

- The number of preemptions based on the period changes should be taken into account
- One of the following two inequalities should be met:

1. \[ \sum_{t=0}^{n_p^\alpha-1} \left\{ f_c^\alpha(T_1^*,i) \right\} + C_2 \leq T_2 \]

2. \[ \sum_{t=0}^{n_p^\alpha-2} \left\{ f_c^\alpha(T_1^*,i) \right\} + C_2 \leq \sum_{t=0}^{n_p^\alpha-2} T_1^*,i \]
Deceleration Analysis

- A similar approach to acceleration analysis can be applied with care
  - Since the task period is sustainable, CCRT will always work while decelerating if the CCRT is originally schedulable

- One of the following two inequalities should be met:

1. \[ \sum_{t=0}^{n_p-1} \left\{ f_c^* (T_1^{*,i}) \right\} + C_2 \leq T_2 \]

2. \[ \sum_{t=0}^{n_p-2} \left\{ f_c^* (T_1^{*,i}) \right\} + C_2 \leq \sum_{t=0}^{n_p-2} T_1^{*,i} \]
One Rhythmic Task and Many Periodic Tasks
The Maximum Possible $C_1$

**Theorem 1**: Let $f_{C_{max}}^*(T_1^*)$ denote the function which returns the maximum possible value of WCET for $C_1^*$ which makes $\Gamma$ schedulable. Then,

$$f_{C_{max}}^*(T_1^*) = \min_{\forall \tau_i \in \Gamma} \left\{ \max \left( \left( T_i - \sum_{j=2}^{i} T_j \right) C_j, T_1^* - \frac{\sum_{j=2}^{i} T_j}{T_1^*} C_j \right) \right\}$$

**Theorem 2**: The slope of $f_{C_{max}}(T_1^*)$ is either 1 or 0

**Theorem 3**: The absolute minimum flexion point lies in the range, $\frac{T_2}{2} + \frac{C_2}{2} \leq T_1^* \leq T_2$
A Rhythmic Task with 3 Periodic Tasks

**Maximum possible value of** \( C_1^* \)

**Maximum total utilization obtained**

\[ f_{C_{\text{max}}}^* \]

\[ \tau_2: (1,7) \]
\[ \tau_3: (1,10) \]
\[ \tau_4: (1,23) \]

**Total Utilization**

\[ T_1^* \]
General Acceleration Analysis

- **Extended** from the two inequalities used for one rhythmic task and one periodic task
- The **number of preemptions** caused by acceleration should be taken into account
- The corresponding execution time should be calculated based on the task classification

### Algorithm 2 Rhythmic-Acc-α(Γ, α, nα)

**Input:** Γ: a taskset including a rhythmic task, α: the acceleration ratio and nα: the duration of rhythmic task acceleration in terms of the number of job releases

**Output:** Schedulability of Γ

```
1: for i = 2 to n do
2: ▶ Calculate the initial condition for each task τi
3: \[ W_i^0 = C_1^* + \sum_{j=2}^{i} C_j \text{ and } W_i^1 = 0 \]
4: \[ k = 0 \]
5: while \[ W_i^{k+1} \neq W_i^k \] do
6: ▶ Pick the maximum number of preemptions for each iteration
7: \[ n_{p,i}(W_i^k) = \text{Num-Preemptions}(T_1^*, \alpha, n_\alpha, W_i^k) \]
8: \[ E_1^* = \text{Execution-Time}(n_{p,i}(W_i^k), C_1^*, \alpha, n_\alpha) \]
9: \[ W_i^{k+1} = C_i + E_1^* + \sum_{h=2}^{i-1} \left\lfloor \frac{W_i^{k+1}}{T_h} \right\rfloor C_h \]
10: Update necessary parameters
11: if \[ W_i^k \leq D_i \] then
12: Mark \( τ_i \) schedulable
13: if all tasks schedulable then
14: return TRUE
15: else
16: return FALSE
```
## Back to Boss

<table>
<thead>
<tr>
<th>Name</th>
<th>Cycle (Hz)</th>
<th>Period (ms)</th>
<th>Average Utilization (%)</th>
<th>Average Demand (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>LocalPlannerTask</td>
<td>10</td>
<td>100</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>roadBlockageDetector</td>
<td>10</td>
<td>100</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>BehaviorTask</td>
<td>100</td>
<td>10</td>
<td>4</td>
<td>0.4</td>
</tr>
<tr>
<td>ControllerTask</td>
<td>100</td>
<td>10</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>MissionPlannerTask</td>
<td>1</td>
<td>1000</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>Planner3DTask</td>
<td>10</td>
<td>100</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>dataLoggerTask</td>
<td>500</td>
<td>2</td>
<td>1</td>
<td>0.02</td>
</tr>
<tr>
<td>dataLoggerTask</td>
<td>500</td>
<td>2</td>
<td>1</td>
<td>0.02</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
</tbody>
</table>

**NOTE**: The utilization varies depending on the processor clock frequency.

<From the driving simulation scenario of RobotCityCharlie, SystemTest-Navigation>
Utilization of a Rhythmic Task on Boss

Use this value as the maximum utilization allowed for the rhythmic task

10% (relative) variation on utilization of a rhythmic task

Utilization

$T_1^*$ (ms)
Back to Engine Control

- RPM varies from 500 to 9000
  - The period varies from 7.5ms to 120ms
- 9 periodic tasks for running control algorithms, reporting the current status to a diagnosis module and managing sensors
  - $\tau_2: (5,120)$, $\tau_3: (20,120)$, $\tau_4: (5,180)$, $\tau_5: (6,200)$, $\tau_6: (8,240)$, $\tau_7: (10,240)$, $\tau_8: (3,300)$, $\tau_9: (1,360)$, $\tau_{10}: (7,400)$
Engine Control Analysis: WCET and Utilization

The ideal value from the analysis

The proposed mode changes for the rhythmic task to maintain schedulability

- Task changes algorithm (to compute less) as engine speeds up.
Engine Control Analysis: Acceleration

- Mode-change requirements can be accurately determined.
- For longer acceleration duration, the acceleration ratio $\alpha$ must (naturally) be controlled.
Outline

- Motivation
- Rhythmic Tasks
- Case Study
- Related Work
- Conclusion
Related Work

- **Dynamic Parameter Adjustment**
  - Elastic task models for dynamic priority scheduling
  - Gravitational task models for non-preemptive job scheduling
  - Q-RAM for maximizing system utility

- **Acyclic Task Model**
  - A task model where a task makes successive invocations *without constraints*

- **Mode Change Protocol**
  - Schedulability test for *discrete* mode changes
Conclusions and Future Work

- Cyber-physical systems can trigger new task execution patterns
  - Task parameters depend on CPS state
    - **Rhythmic tasks**: periods depend on physical environment
  - Mode changes may need to be introduced to reduce utilization and maintain schedulability

- We studied three different configurations
  - Steady-state operation, Acceleration and Deceleration

- Future work
  - Utilization-bound analysis
  - Multiprocessor analysis
  - OS support
Thank You!

Time for a Mode Change....

Questions?