Scheduling of frame-based Embedded Systems with Rechargeable Batteries

1. Introduction

In this paper there is a study of scheduling in real time systems that rely on rechargeable battery sources with battery and deadline constraints

2. Model

Scheduling a set of tasks in a single processor with variable voltage and frequency.

2.1. Tasks

→ N tasks τ_i have to execute within a frame and all have a common deadline D

→ Each task τ_i takes $t_i = C_i \cdot S_i$ time to run:

C_i = worst case number of cycles req to execute
S_i = speed at which the task runs.

→ Task τ_i consumes energy at a constant rate p_i

energ consumed by task i when it runs between t_i^1 and t_i^2 is given by

$$e_i = \int_{t_i^1}^{t_i^2} p_i \cdot dt = p_i \cdot t_i$$

$$t_i^2 = t_i^1 + t_i$$

Note than even if task i is preempted still the

$$e_i = \int_2 p_i \cdot dt = p_i \cdot t_i$$

2.2 Battery

→ The system runs on a battery whose energy level is $\{E_{min}, E_{max}\}$ $\quad \Delta E = E_{max} - E_{min}$

→ Recharging rate is constant at r

if fully charged E_{max}

if fully discharged E_{min}

→ The consumption rate is given by p_i

$\boxed{p_i' = r - p_i}$ is the instantaneous replenishment rate of energy of system while task i runs.

- if $p_i' > 0$, then task i consumes less than recharging rate
- if $p_i' < 0$ it consumes more

→ Tasks are divided into 2 groups → recharging tasks
→ dissipating tasks

Recharging tasks $CR = \{\tau_i \mid p_i' > 0\}$

dissipating tasks $CD = \{\tau_i \mid p_i' < 0\}$

$$|R| = \sum_{i \in R} p_i \cdot t_i \qquad |D| = -\sum_{i \in D} p_i \cdot t_i$$

2.3 Problem definition

To find a schedule which is able to execute all the tasks within deadline D, starting with a fully charged battery ending at the same battery level as we started.

Also to reduce the cycle time of the system.

3. Fixed Speed Processor

Case 1: All tasks are run at full speed of processor.

- The goal of the algorithm is to execute all the tasks within deadline

```
→ loop
      tidle = 0
→     if |D| > |R|
            tidle = (|D| - |R|)/r
            tidle ∪ R ; pidle = 0
      endif
→     if Σ_{i=0}^{N} ti + tidle > D
            {Failure}
      endif
→ while (D != 0) && (E <= Emin)
        if (E != Emin)
        {
            Schedule D = zi
        }
        else
            preempt zi
        endif
  end while
→ while (R != 0 && E ≥ Emax)
  {
      if (E != Emax)
      {
          sched (R = zi)
      }else
          preempt zj
  end while
end loop.
```

Illustration of Algorithm

Tasks	t_i
τ_1	3
τ_2	4
τ_3	5

$E_{max} = 6$

$E_{min} = 0$

$D = 15$

$t_{idle} = 1$

$\Sigma t_{task_i} = 13$

$13 < D \Rightarrow$ True.