Power management for energy harvesting based systems

Alain Pegatoquet
and... Andrea Castagnetti, Michel Auguin, Cécile Belleudy
1. Introduction and motivation

2. Related works

3. System modeling

4. Power management for energy harvesting WSN nodes
   a) Open-loop energy neutral power manager
   b) Closed-loop: OL plus negative energy power manager

5. Simulation results

6. Conclusion
1. Introduction and Motivations
1. Introduction and motivations

- Wireless Sensor Networks (WSNs) consist of distributed autonomous sensors that can operate without a pre-established network infrastructure.

- Energy limitation arises from the fact that sensor nodes are equipped with **small batteries**.

- Energy harvesting allows **on-site charging** of rechargeable batteries thus increases the lifetime of the nodes.

- With proper **power management**, the lifetime can be extended almost indefinitely (subject to hardware failure).

- Parameters handled by the power manager:
  - Scavenged energy (prediction and/or measure)
  - Consumed energy by the node for different activities (measures)
1. Introduction and motivations

- The objective of the power manager is to balance the harvested and the consumed energy (autonomous objects → GRECO).

- Power management algorithms are in charge of setting the wake-up period of the platform and/or the duty-cycling.
  - In battery powered systems, the lowest tolerable duty cycle is typically chosen in order to extend the achievable lifetime to its maximum.
  - In a harvesting system, the goal is to choose a duty cycle such that the averaged consumed power is set to its highest value allowed for energy neutral operation. This will allow operating at the best possible performance...

- Power management requirements
  - Efficient
  - Reduced overhead
2. Related Works
2. Related Works

- **Kansal et al.** [Kansal 2007] propose a simple solar energy prediction algorithm based on an Exponentially Weighted Moving-Average (EWMA) filter to support an harvested-energy management approach.

- **Recas et al.** [Recas 2009] present a solar prediction algorithm, named Weather-Conditioned Moving Average (WCMA) that is capable of exploiting the solar energy more efficiently than state-of-the-art energy prediction algorithms (e.g. EWMA).

- **Moser et al.** [Moser 2010] used similar prediction as [Kansal 2007] in their proposed adaptive power management framework.

- **Ali et. al** [Ali 2010] have evaluated prediction accuracy of different solar harvested-energy prediction algorithms using multiple real solar-panel data-set. They also have implemented prediction algorithm on real hardware, so that computation overhead was measured. They suggested guidelines to simplify parameters tuning for different working conditions.

- **Ferry et al.** [Ferry 2011] proposed to develop a power and energy estimator to determine the node autonomy duration of WSNs (CAPNET project with a representative scenario of fire brigade intervention).
Kansal et al. were the first to propose a simple solar energy prediction algorithm to support their harvested-energy management approach.

The predictor is based on the observation that energy generation during a given timeslot of day is similar to that generated at the same instant on previous days and it can be estimated using an exponentially weighted moving average (EWMA) of historical data.

\[
\bar{x}(i) = \alpha \bar{x}(i - 1) + (1 - \alpha) x(i)
\]

- Historical average value maintained for slot \(n\) (day \(i\))
- Historical average value maintained for slot \(n\) on previous day (\(i-1\))
- Value of real energy observed at the end of slot \(n-1\)

\(\alpha\): a weighting factor \(\epsilon [0, 1]\)

Choice of the \(\alpha\) parameter through error evaluation

\(\Rightarrow \alpha = 0.5\) minimizes the error

[Kansal 2007]
- Windows size duration = 24 hours (a day)
- The day is divided into slots.
- Each slot is 30 minutes long, as the variation in generated power level is assumed to be small within a 30 minute duration.
- Number of slots per day: \( N_w = 48 \)

\[
\bar{x}(i) = \alpha \bar{x}(i - 1) + (1 - \alpha) x(i)
\]
Kansal et al.

\[ \bar{x}(i) = \alpha \bar{x}(i - 1) + (1 - \alpha)x(i) \]

- Harvested energy measured over 67 days using a Mica2 based energy harvesting sensor node (energy sampled at a period of 10 seconds).

- Error performance of the EWMA based prediction method: average error in each time slot averaged over that slot on all 67 days.

Error is larger during the daytime. The maximum prediction error during daytime is 20mW (13% vs. energy level close to 150mW during daytime).

[Kansal 2007]
- Optimization problem to maximize the data-rate based on the harvested energy prediction
  - optimizing the adaptation of the duty-cycling between active and low power modes.
- In order to maintain energy neutral operation, duty cycles in the future time slots will be reduced or increased to account for the prediction error made in the current time slot.

- So 2 cases are considered:
  - When more energy is received than predicted, the excess energy (received in last time slot) is used to increase the duty-cycle in future time slots.
    - Slots having low predicted energy will have a high duty-cycle.
  - When less energy is received than predicted, duty-cycle in future time slots is reduced in order to compensate energy from subsequent slots.
    - Slots having high predicted energy will have a low duty-cycle.
Recas et al. propose a prediction algorithm named **Weather-Conditioned Moving Average (WCMA)** to take into account both the current and past-days weather conditions.

**Drawback of EWMA:**

![Diagram showing real vs EWMA energy over time](image)

When the sunny and cloudy days alternate, the EWMA produces a significant error in its prediction, due to the high impact of the solar conditions of previous day in the predicted value.

To avoid this effect, Recas et al. propose a prediction algorithm that takes into account not only the solar conditions at a specific time of the day, **but also the weather conditions in the current day**.

[Recas 2009]
• WCMA algorithm uses an **E** matrix of size **DxN** that stores N energy values for each D past days.

\[ E(d, n+1) = \alpha \cdot E(d, n) + GAP_k \cdot (1 - \alpha) \cdot M_D(d, n+1) \]

- **E** matrix of size **DxN**
- **M_D** (mean)
- **V** = \((v_1, v_2, ..., v_k)\)
- **P** (weighting)

[Recas 2009]
The predicted value is related to the previous sample in the same day and the mean value of the past samples (at the same hour of the day).

\[ E(d, n+1) = \alpha \cdot E(d, n) + GAP_k \cdot (1 - \alpha) \cdot M_D(d, n+1) \]

- \( v_k > 1 \) means sunny day, \( v_k < 1 \) means cloudy day.
- The factor \( GAP_k \) measures the solar conditions in the present day relative to the previous days.

\[ GAP_k = \frac{V \cdot P}{\sum P} \]

\[ M_D(d, n) = \frac{\sum_{i=d-1}^{d-D} E(i, n)}{D} \]

\[ v_k = \frac{E(d, n - K + k - 1)}{M_D(d, n - K + k - 1)} \]

[Recas 2009]
To optimize the WCMA’s parameters, an error function is used to find optimal values for:

- The size of the E(DxN) matrix
- The $\alpha$ factor
- The number $K$ of past samples to weight

To optimize these values, the energy available has been recorded from a solar panel every minute during 45 consecutive days.

$$Err = \frac{1}{N} \sum_{i=1}^{N} \text{abs} \left( 1 - \frac{E_{\text{Real}}}{E_{\text{Pred}}} \right)$$

- The following parameters allows minimizing the prediction error:
  - $D = 4$ days, $N = 48$ samples/day, $K = 3$ past samples, and $\alpha = 0.7$

[Recas 2009]
Comparison of WCMA vs. EWMA

- Over all 45 days of the collected solar panel data, EWMA gives an average error of 28.6% compared to 9.8% obtained with WCMA algorithm.

[Recas 2009]
Conclusion

- **EWMA-based algorithm** (Kansal approach) is accurate for consistent weather conditions, but when cloudy and sunny days are mixed, recent days energy values introduce significant prediction errors.

- In other hand, **WCMA based approach** (Recas et al.) takes into account weather changes but introduces a non significant CPU overhead and memory footprint.

- None of these approaches takes into account the State of Charge (**SoC**) of the battery for optimizing the power manager decisions and avoiding a complete discharge of the battery...

- Moreover, a wake-up period of 30 minutes for the power manager seems to be too slow for a good system reactivity.
3. System Modeling
Global View of the System

- Generic Solar Harvesting Systems

![Diagram of energy harvesting system](image)

- Battery Centric Modeling

\[ \text{SoC}(\alpha, \beta, t) \]

\[ \text{SoC}_{\text{max}}, \text{SoC}_{\text{min}} \]

Ecole Thématique EcoFac 2012 – La Colle sur Loup, 21-25 mai 2012
A Task-level Platform Load Model

- $Q_{\text{sense}}$: charge consumed for a sensing operation [Coulomb].
- $Q_{\text{tx}}$: charge consumed for a RF transmission [Coulomb].
- $Q_{\text{rx}}$: charge consumed for a RF transmission [Coulomb].
- $T_{\text{wi}}$: wake-up period for sensing [seconds].
- $T_{\text{tx}}$: RF transmission period (multiple of $T_{\text{wi}}$) [seconds].
- $T_{\text{fwd}}$: Forwarding transmission period (multiple of $T_{\text{wi}}$) [seconds].

- Rate of discharge of the battery

\[ \alpha = \frac{Q_i}{T_i} \text{[C/s] or [A]} \]
A Battery and Energy Harvesting Integrated Model

- Rate of recharge of the battery
  - The $\beta$ parameter models both the efficiency of the energy harvester and the efficiency of the voltage regulator and the charge circuit that are used to recharge the battery.
  - $\beta$ is expressed in Ampere and indicates the rate at which the energy harvester can recharge the battery under a fixed amount of energy available from the environment.
- Rate of recharge for different light conditions
State of Charge (SoC) Estimation

- The SoC depends upon:
  - The harvested energy ($\beta$).
  - The platform current consumption ($\alpha = \frac{Q_i}{T_{wi}}$).
  - The leakage current (battery self discharge and low-power mode current consumption).

The State of Charge (SoC) of a battery for $n$ wake-up periods ($T_{wi}$)

$$SoC(t + nT_{wi}) = SoC(t) + [\beta - (\alpha_s + \alpha_{Tx} \frac{T_{Wi}}{T_{Tx}} + \alpha_{fwd} \frac{T_{Wi}}{T_{fwd}})] \times nT_{Wi} - K_{leak} \times nT_{Wi}$$

Conditions

$$SoC(t = 0) = SoC_{max}.$$
$$SoC_{min} \leq SoC(t + n T_{wi}) \leq SoC_{max}.$$
$eta$, $\alpha_s$ and $\alpha_{Tx}$ are constants on $[t, t + n T_{wi}]$.  

[Castagnetti 04-2012]
Model Validation

Case Study: Model Validation Using the EZ430 platform

- Texas Instruments EZ430 solar energy harvesting platform
  - MSP430 microcontroller (cadenced at 16MHz) and a CC2500 RF transceiver.
  - 2.25in x 2.25in solar panel.
  - Two lithium thin-film 50 µAh rechargeable batteries.
Case Study: Model Validation Using the EZ430 platform

- **EZ430 Power Supply Architecture**

  The solar panel is optimized for operating indoor under low-intensity fluorescent light.
  
  The boost converter raises the voltage to the necessary level needed to recharge the battery or power the system.
  
  The charger circuit is controlled by a control circuit that can disconnect it from the boost converter if the output voltage falls below the level needed to recharge the battery.
  
  A protection circuit is placed at the output of the two Lithium thin-film 50 μAh rechargeable batteries, in order to protect them from fully discharging.
  
  A 1000 μF capacitor is connected to the output of the batteries to mitigate the effect of the pulse discharge current.
  
  The application board is equipped with an MSP430 microcontroller and a CC2500 RF transceiver.
  
  Two Lithium thin-film 50 μAh rechargeable batteries.
Case Study: Model Validation Using the EZ430 platform

- **Application**
  - The End Device (ED) application periodically sends a packet through the air to an Access Point (AP).
  - During the idle period both the microcontroller and the radio chip are put into a sleep state.
Model Validation

Case Study: Model Validation Using the EZ430 platform

- Platform load characterization

\[
Q = [(2,8mA*3,4ms)+(13,8mA*1ms)+(23,4mA*1,4ms)+(2,8mA*7ms)+(5mA*10,6ms)]
= 9,52 \mu As + 13,8 \mu As + 32,76 \mu As + 19,6 \mu As + 53 \mu As
= 128,68 \mu As
= 0,0357 \mu Ah
\]

- The platform load characterization is performed by measuring the current consumed for the different tasks (TX, RX, CPU…) of an end-device.
Case Study: Model Validation Using the EZ430 platform

- Experimental results: lifespan prediction (in minutes)

<table>
<thead>
<tr>
<th>$T_{wi}$ [sec]</th>
<th>$\alpha$ [(\mu\text{A})]</th>
<th>0 lux</th>
<th>0 lux</th>
<th>0 lux</th>
<th>0 lux</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(exp)</td>
<td>(model)</td>
<td>(exp)</td>
<td>(model)</td>
</tr>
<tr>
<td>1</td>
<td>128.52</td>
<td>28</td>
<td>29</td>
<td>50</td>
<td>38</td>
</tr>
<tr>
<td>2</td>
<td>64.26</td>
<td>54</td>
<td>59</td>
<td>140</td>
<td>112</td>
</tr>
<tr>
<td>3</td>
<td>42.84</td>
<td>83</td>
<td>88</td>
<td>312</td>
<td>332</td>
</tr>
<tr>
<td>4</td>
<td>32.13</td>
<td>107</td>
<td>117</td>
<td>/</td>
<td>$+\infty$</td>
</tr>
<tr>
<td>6</td>
<td>21.42</td>
<td>162</td>
<td>176</td>
<td>/</td>
<td>$+\infty$</td>
</tr>
<tr>
<td>10</td>
<td>12.85</td>
<td>290</td>
<td>294</td>
<td>/</td>
<td>$+\infty$</td>
</tr>
<tr>
<td>20</td>
<td>6.43</td>
<td>585</td>
<td>587</td>
<td>/</td>
<td>$+\infty$</td>
</tr>
</tbody>
</table>

- Error margins:
  - 16.8% error margin when $\beta = \sim 200$ lux
  - 5.5% error margin when $\beta = 0$

- $\beta \geq \alpha$
4. Power Management for Energy Harvesting WSN Nodes
Adapting the performance and the power consumption of the system to the available energy (i.e. $\beta$).

Performance scaling is achieved by varying the wake-up period ($T_{wi}$) of the sensor node (i.e. $\alpha$).

The objective is to operate in Energy Neutrality

$\Rightarrow$ balancing the $\alpha$ and $\beta$ parameters.

Hypothesis:

- The power manager must measure/estimate $\beta$.
- The values of $\alpha$ can be computed on-line if $Q_{\text{sense}}$, $Q_{\text{Tx}}$ ... are known ($Q_i$ off-line profiling).

Consequence:

- In energy neutrality, SoC($t$) is constant over time: $\text{SoC}(t) = \text{SoC}(t + n \ T_{wi})$
Balancing the Harvested and Consumed Energy

A Generic Power Manager

- **Architecture**

  - The power management algorithms are in charge of setting the $T_{wi}$ of the platform.
  - Different algorithms can be used depending on the energy availability and the present state of charge of the battery.

  3 main blocks:
  - Energy harvesting sensor
  - Several power managers algorithms
  - SoC predictor

  - Architecture diagram:
    - Energy Harvesting Sensor
    - Battery recharge model
    - Energy neutral Power Manager
    - SoC Predictor
    - Operating Mode
    - Negative Energy Power Manager
    - Positive Energy Power Manager
A Generic Power Manager

- Three types of operations are considered:
  - **Energy-neutral**: the system works only with the energy coming from the environment. However, a portion of the harvested energy can also be used to recharge the battery.
  - **Negative-energy**: as soon as there is no energy to harvest, the operation is only supported by the energy stored in the battery.
  - **Positive-energy**: if the harvested energy exceeds the required energy to operate at peak performance, the recharge rate of the battery can be improved.
Balancing the Harvested and Consumed Energy

An Open-Loop Energy Neutral Power Manager (OL-PM)

- Architecture

<table>
<thead>
<tr>
<th>Power Manager SW Timer</th>
<th>Power Manager SW Timer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tpm</td>
<td>Tpm</td>
</tr>
</tbody>
</table>

- Energy Harvesting Sensor
- Battery recharge model
- Energy neutral Power Manager
- PM core
- App DB

Equations

- Power Management software timer
  \[ T_{pm} = n \cdot T_{wi} \]
- Energy Neutral wake-up period
  \[ \text{SoC}(t) = \text{SoC}(t + n \cdot T_{wi}) - Q_{pm} \]

\[ T_{wi} = \left[ \frac{Q + Q_{pm}/n}{\beta - K_{\text{leak}}} \right] \]

[Castagnetti 02-2012]
5. Simulation Results
The values of $Q_{\text{sense}}$ and $Q_{\text{Tx}}$ have been measured with the TI EZ430 platform.

$\beta$ is estimated from light intensity measurements taken in an office (indoor conditions) at 5-second intervals during 5 days using a lux-meter.

Harvested energy
Evaluation Metrics

- **Average data-Rate** ($<Rd>$): the average throughput is computed over the five days, including the periods of time where the battery is fully discharged.

\[
<Rd> = \frac{\text{Packet}_{\text{payload}}}{T_{wi}}
\]

Packet$_{\text{payload}}$ = 33 bytes

- **Maximum and minimum data-rate** ($R_{d_{\text{max}}}$, $R_{d_{\text{min}}}$): peak and minimal achievable performance of a node using a given power manager.

- **Average SoC** ($<SoC>$): to assess if the power management algorithm drifts in energy-neutral condition.

- **Battery failures** ($B_f$): a value of 0 means that the battery is never fully discharged and the node is always operational. Otherwise, it exists at least a $t^* \mid \text{SoC}(t^*) < \text{SoC}_{\text{min}}$
Performance Analysis and Comparison

- The OL-PM has been simulated over the five-day data-set.
- A state of the art power manager ([Kansal 2007]) for solar energy harvesting WSN has been implemented and simulated over the same data-set.

Power Managers comparison:

<table>
<thead>
<tr>
<th></th>
<th>( \langle \text{Rd} \rangle ) [bits/s]</th>
<th>( \text{Rd}_{\text{max}} ) [bits/s]</th>
<th>( \text{Rd}_{\text{min}} ) [bits/s]</th>
<th>( \langle \text{SoC} \rangle ) [( \mu \text{Ah} )]</th>
<th>( B_f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kansal</td>
<td>29.55</td>
<td>132</td>
<td>0</td>
<td>65.81</td>
<td>9</td>
</tr>
<tr>
<td>OL-PM</td>
<td>44.61</td>
<td>132</td>
<td>2.2</td>
<td>88.8</td>
<td>0</td>
</tr>
</tbody>
</table>

- The average data-rate is improved by 51% using the OL-PM.
- The OL-PM always provide a minimum QoS (\( \text{Rd}_{\text{min}} \neq 0 \)).
- Remark: Simulating the five-day data-set takes about 2 minutes using a laptop PC equipped with an Intel Core-i5 CPU cadenced at 2.5 GHz and 3.8 GB of RAM.
- The Open-Loop power manager avoids battery failures but does partially exploits the battery capacity...
A Closed-Loop Energy Power Manager (CL-PM)

- Architecture
  - The CL-PM is composed of same building blocks as the OL-PM, plus the negative-energy power manager, the Zero-Energy-Interval (ZEI) predictor and the SoC predictor.

- Two power management strategies are available: the energy-neutral and the negative-energy techniques.

- Closed-loop is used as the output signal (SoC(t + nTwi)) is fed back into the input of the negative-energy power manager block.

[Castagnetti 05-2012]
### Block Diagram

- **ZEI**: a Boolean that indicates if the incoming $\beta \leq \beta_{th}$. An hysteresis comparator is used to prevent oscillations.

- **$D_{ZEI}$**: an integer value expressed in seconds used to estimate the ZEI duration. This value is computed as the average of the last four measured $D_{ZEI}$. If less than four measures are available, a default value of 14 hours (50400 seconds) is used.
Behavior of the ZEI predictor during the 5-day data-set

The hysteresis of the comparator prevents undesired oscillation of the ZEI signal and thus provides an accurate estimation of the ZEI duration.
Problem: finding the wake-up period $T_{wi}$ that prevents a complete discharge of the battery.

Equations

- If we call $t^*$ the start of a ZEI, the condition that must be respected is:
  \[ SoC(t^*) - (\alpha + K_{\text{leak}})D_{\text{ZEI}} \geq SoC_{\min} + M \]

\[ T_{wi} \geq \frac{Q D_{\text{ZEI}}}{SoC(t^*) - K_{\text{leak}} D_{\text{ZEI}} - (SoC_{\min} + M)} \]

where
- $SoC(t^*)$ is the state of charge of the battery at the beginning of the ZEI,
- $D_{\text{ZEI}}$ is an estimation of the duration of the ZEI,
- $M$ is the battery discharge margin
Performance Analysis and Comparison

- The CL-PM has been simulated over the five-day data-set.
- **Power Managers comparison:**

<table>
<thead>
<tr>
<th></th>
<th>$\langle R_d \rangle$</th>
<th>$R_{d_{\max}}$</th>
<th>$R_{d_{\min}}$</th>
<th>$\langle SoC \rangle$</th>
<th>$B_f$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>[bits/s]</td>
<td>[bits/s]</td>
<td>[bits/s]</td>
<td>[$\mu$Ah]</td>
<td></td>
</tr>
<tr>
<td>Kansal [8]</td>
<td>29.55</td>
<td>132</td>
<td>0</td>
<td>65.81</td>
<td>9</td>
</tr>
<tr>
<td>OL-PM</td>
<td>44.61</td>
<td>132</td>
<td>2.2</td>
<td>88.8</td>
<td>0</td>
</tr>
<tr>
<td>CL-PM</td>
<td>45.87</td>
<td>132</td>
<td>0.37</td>
<td>69.27</td>
<td>0</td>
</tr>
</tbody>
</table>

- The Closed-Loop Power Manager:
  - slightly improves the average data-rate
  - Does not lead to battery failures
  - Decreases the $<SoC>$...
Execution Traces

- The Closed-Loop power manager better exploits the battery capacity
- It provides better QoS during the night...
The CL-PM behavior during ZEIs (where the negative-energy strategy is used) can be tuned through the adaptation of the safety margin (M) parameter.

This parameter prevents the battery from fully discharging, thus avoiding battery failures.

This parameter counterbalances the inaccuracies on the SoC and $D_{ZEI}$ estimations.

<table>
<thead>
<tr>
<th>M [μAh]</th>
<th>2</th>
<th>3.15</th>
<th>6.3</th>
<th>12.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_d%$</td>
<td>3.17%</td>
<td>5%</td>
<td>10%</td>
<td>20%</td>
</tr>
<tr>
<td>$\langle SoC \rangle$ [μAh]</td>
<td>69.27</td>
<td>70.37</td>
<td>71.89</td>
<td>78.94</td>
</tr>
<tr>
<td>$SoC_l$ [μAh]</td>
<td>37.5</td>
<td>38.99</td>
<td>41.82</td>
<td>48.45</td>
</tr>
<tr>
<td>$\langle Rd \rangle$ [bits/s]</td>
<td>45.87</td>
<td>45.82</td>
<td>45.73</td>
<td>44.73</td>
</tr>
</tbody>
</table>

Since M is used in the negative-energy strategy, only the data-rate during ZEIs is affected by this parameter.
6. Conclusions
Conclusions

Modeling
- We have proposed an high-level modeling approach for energy harvesting WSN nodes.
- Using the $\alpha$ and $\beta$ model, the system can be described in a compact form.

Power Management
- Up to 30% data-rate improvement compared to a state of the art power manager.
- System robustness is improved by avoiding the battery to discharge too deeply.
- A global approach is needed for an efficient wake-up period adaptation (canal traffic, TX power, etc.)
References


Thank you!

Any questions?