A Modular Energy System Controller

Dhirendra Singh
RMIT University
Melbourne, Australia
dhirendra.singh@rmit.edu.au

Geoff James
CSIRO
Sydney, Australia
geoff.james@csiro.au

July 23, 2010

1 Introduction

Energy storage helps to enable increasing levels of renewable energy in our electricity system, and the rapidly maturing supply chains for several battery technologies encourages electricity utilities, generators, and customers to consider using large battery systems. Large battery systems usually comprise multiple modules and in many installations these may be controlled independently. Modules may be operated in synchrony but often there are strategic reasons to keep some modules in a different state to others. For example, if it is undesirable to change the direction of power flow between charging and discharging too frequently, a subset of modules may be used for each direction until it is necessary to change their roles. Also, some technologies have specific requirements, such as the zinc-bromine flow battery for which a complete discharge at regular intervals is desirable to strip the zinc plating and ensure irregularities never have an opportunity to accumulate. Where they exist these requirements place further constraints on module control.

Given, then, an input signal which is the requested rate of charging and discharging for a large battery installation, as a function of time, we would like a control algorithm for the set of component modules that implements the requested rate as the sum over the module rates of charging and discharging. The input signal will be different every day but will have many features that are diurnal or nearly so, due to typical variations of electricity demand and solar and wind energy generation sources, and the repetitive patterns that may be seen over several days of the input signal suggest that a learning algorithm may be appropriate. Our problem is to develop a method for on-line learning that will result in a useful control regime for a modular battery system, when installed at a new site and provided with an input signal derived from the electricity demand and renewable supply at that site.

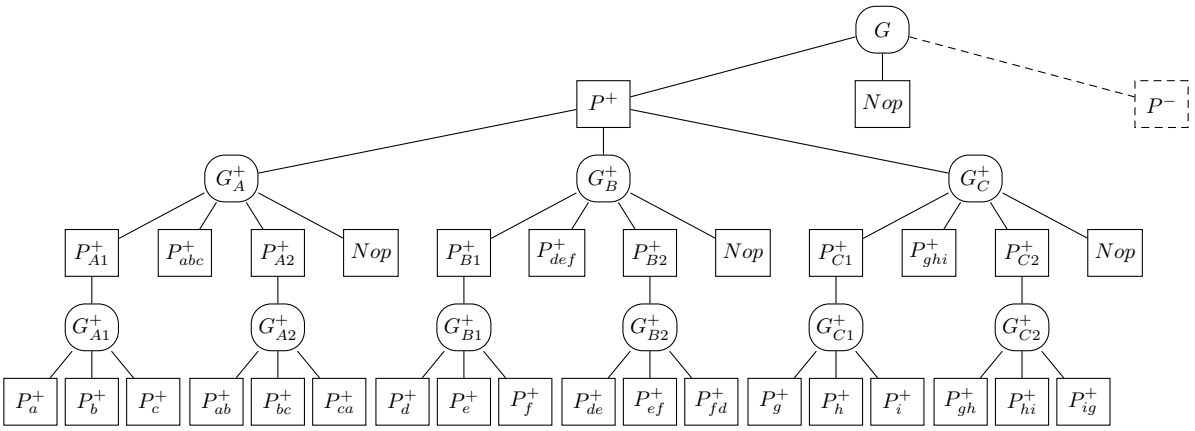

Figure 1: A BDI goal-plan hierarchy for the battery system with nine modules $[a, b, \ldots i]$. The structure specifies a hierarchy of control decisions where the outcome is a combination that specifies which modules to charge or discharge for the given period. Symbols A, B, and C specify module sub-groups. Goal G^+_{B2} for instance, is a goal to enable charging in a group of two modules. Plan P^+_{fd} then, is one plan that may achieve this by enabling charging on modules f and d. The P^- sub-tree is similar in structure to P^+ and is not shown for brevity.

2 Approach

Figure 1 shows a possible BDI goal-plan hierarchy design for a battery system with nine modules $[a, b, \ldots i]$. The top level plan P^+ describes a charging strategy while plan P^- specifies a discharging strategy. The structure of plan P^- is similar to P^+ and may be imagined by replacing the $+$ symbol with $-$ throughout. Plan Nop is an empty plan that performs no action for that particular period. Goals are indicated by a prefix G while plans have a prefix P. The nine modules are divided into three sub-groups A, B, and C. Goal G^+_{B2} for instance, is a goal to enable charging in a group of two modules. Plan P^+_{fd} then, is one plan that may achieve this by enabling charging on modules f and d. Notice that the hierarchy constrains the possibilities for the change in module states. For instance, for the given period, the hierarchy only specifies which combination of modules should start charging, or which ones should start discharging, but never a combination of both.

Leaf plans in the hierarchy may fail when selected if associated module constraints are violated for that period. For instance, plan P^+_g might fail because it is only allowed to change directions once every four periods say, and charging it in this period will violate that constraint. Similarly, plan P^-_{bc} might fail because one of the modules in the group (i.e. b or c) may have already been fully discharged and so further discharge is not possible in this period. Finally, plan P^+_a might fail because it has a requirement to be fully discharged once every day, and charging it in this period will violate that constraint. Since these checks are performed prior to taking any action, then BDI failure recovery may be performed to select a different plan until all internal constraints are satisfied.

In a given period, a possible resolution of the top level goal G may start with the selection of plan P^+. The charging strategy P^+ succeeds when each one of it's three sub-goals are resolved in sequence. Let's say this was achieved by the selection sequence $G^+_A[P^+_{A2}] \cdot G^+_{A2}[P^+_{bc}] \cdot G^+_B[P^+_{def}] \cdot G^+_C[P^+_{C1}] \cdot G^+_{C1}[P^+_i]$. Here we assume that no individual module constraints were violated, or in any case that any violation was addressed by picking a different plan through the BDI failure recovery mechanism. Plan P^+ will then execute this charging strategy on the battery modules, and at the end of that period determine success of failure status by ... measuring performance?

3 Notes

- The input to the system is a signal profile – a sequence of 3-tuples of the form $(period, amplitude, direction)$ – each representing the external request. The variables in the 3-tuple will be discrete valued, and the discretisation will be selected to ensure suitable richness for learning purposes. Input data would be a typical set (for several days or weeks) of control requests for the whole system.
- The aim is to select an appropriate battery system configuration in response to the state of the world i.e. the value of the 3-tuple input and charge in the internal modules, while maintaining module constraints as much as possible. This equates to making a decision about charging/discharging each module for the given period (eg. hour or half hour).
- Each module has fixed energy and power capacities. A module's internal state may be described by a discrete value $Charge \in [0, 1]$. Each module might have a constraint to require one full discharge per day or two days (if a flow battery). Such constraints would be captured by a domain expert writing the hierarchy.
- The output of the system will be a "configuration" - that specifies each internal module's "state" – $Charging$, $Discharging$ or $Unchanged$. The flow battery in Newcastle has 10 internal modules, so with three possible states that represents $3^{10} = 59049$ possible configurations.
- We will create a hierarchy of control decisions with each level having 2-4 choices. Larger systems (with more that 10 modules) would have more choices per level as well as have deeper hierarchies. A sequence of actions is desirable and could be constructed as a separate choice per module.
- Each non-leaf choice in the hierarchy would comprise of a number of subsidiary choices. This gives the opportunity for recursion although not specified currently.
- It would be nice to reassign modules in the hierarchy periodically (e.g. physically – akin to rotating batteries in a battery holder) to remove any bias.
- The measure of success in each time period should be some numeric measure constructed as a sum of the internal "health" (how well the internal constraints were met) and an external "flow" (or how well the system request was met).
- Keep on learning.

- Failure recovery. If a plan selection means that a module will change direction in less than 3 periods, then the plan will plan. Or if a module is asked discharge below a minimum charge. Also if asked to charge above a maximum charge?
- ***Gotcha***: Currently, there is no recursion specified in the hierarchy but it may be possible to add it later. Similarly, at this stage there is no parameterisation of goals. These are both features highly emphasised in the RAS paper [3] and may have to be included for this application to have impact.
- ***Gotcha:*** Would an online constraint solver be more appropriate for this problem rather than a learning solution? Probably not, because while the SAT solver may produce the best configuration to satisfy the immediate input, the learner will likely select a configuration that maximises the reward for the entire day/week, rather than the immediate period. This may be more desirable.

References

[1] S. Airiau, L. Padgham, S. Sardina, and S. Sen. Enhancing Adaptation in BDI Agents Using Learning Techniques. In *International Journal of Agent Technologies and Systems*, 2009.

[2] D. Singh, S. Sardina, L. Padgham, S. Airiau. Learning Context Conditions for BDI Plan Selection. In *Proceedings of Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2010.

[3] D. Singh, S. Sardina, L. Padgham. Extending BDI Plan Selection to Incorporate Learning from Experience. Robotics and Autonomous Systems, 2010.