The Power Prefetch project uses an Energy Storage Device (ESD) hierarchy similar to the Memory hierarchy, in order to improve the battery life time, to ensure instantaneous burst of power/energy demand, and to optimize the total energy consumption cost.

This software was developed following a pattern of Object Oriented Design (OOD) and Test Driven Development (TDD) under Matlab. The production\_code folder contains all of the application code, and the test\_code folder contains the test suite.

Main ideas:

* Flywheel acts as the cache, battery as the main memory, renewable energy as the disk.
* Renewable energy can charge battery and flywheel, battery can charge flywheel and discharge, flywheel can only discharge.
* Battery discharge rate is slower than flywheel, so use flywheel will ensure the burst of power demand.
* Battery charge and discharge rate (Depth of discharge or DOD) will significantly affect total battery life time, which most of current applications don’t take into account; whereas flywheel charge and discharge rate will not. And flywheel’s total life time can be considered as static and fixed.
* Battery can only charge and discharge at a certain rate, while flywheel can discharge at any rate.
* Using amortized annual cost idea, and a Mixed Integer Linear Program (MILP), we can get optimal charge and discharge cycles for both battery and flywheel for a given time period, to minimize the total amortized cost, as well as the total battery life time.

Parts needed before running:

To run the app, make sure the following are properly installed in your computer (this app was fully tested under windows and Linux, but not yet under mac OS):

* Matlab 2014a or later. intlinprog function is a new feature of Matlab 2014a, and was used in this app.

Inputs from the user:

* Energy demand pattern
* Renewable energy availability
* Battery cost in terms of $/kWh
* Battery capacity in terms of kWh
* Flywheel cost in terms of $/kWh
* Flywheel capacity in terms of kWh
* Flywheel life time in terms of years

Outputs:

* Optimized charge and discharge cycles (usages)for both battery and flywheel
* Battery’s actual depth of discharge for each time period.
* Total amortized cost per day with and without using flywheel

Run the app:

Run test\_batch.m under test\_code folder

Since this project is still ongoing, so the UI part might not be perfect yet, one can try to run it using the test\_batch.m to see how it works for the time being.

Modeling of the ESD hierarchy:

Assumptions:

1. Time series are numOfIntervals, and time step is 1 interval
2. Renewable energy pattern for each minute is given, as G(t)
3. Energy/power Demand for each minute is given, as D(t), in the unit of kWh, and for now it is constant at each time step.
4. There are three ESDs in the hierarchy, from top to bottom are: renewable(G), battery(B), and flywheel(F)
5. At the beginning of the time series, the energy stored in battery(B) and flywheel(F) are half full, equal to their corresponding max capacities.
6. The way this ESD hierarchy works is: G can charge B and F, as well as satisfy D; B can charge F, and D; F can only satisfy D.
7. The self-discharge of B(Loss rate of B) can be ignored, whereas that of F cannot.
8. If, at each time step, G cannot be used fully, it is wasted(cannot be used for later time steps)

Variables that can be controlled, for each time step t:

1. B\_g(t): the amount of green energy charged into B, variable 1 to

numOfIntervals

1. F\_g(t): the amount of green energy charged into F, numOfIntervals

+ 1 to 2 \* numOfIntervals

1. D\_g(t): the amount of green energy to satisfy D, 2 \* numOfIntervals + 1 to 3 \* numOfIntervals
2. F\_b(t): the amount of battery energy to charge F, 3 \* numOfIntervals + 1 to 4 \* numOfIntervals
3. D\_b(t): the amount of battery energy to satisfy D, 4 numOfIntervals + 1 to 5 \* numOfIntervals
4. DoD\_b(t): the Depth of Discharge of B, 5 \* numOfIntervals + 1 to 6

\* numOfIntervals

1. D\_f(t): the amount of flywheel energy to satisfy D, 6 \*

numOfIntervals + 1 to 7 \* numOfIntervals

1. DoD\_f(t): the Depth of Discharge of F, 7 \* numOfIntervals + 1 to 8

\* numOfIntervals

1. E\_b(t): the amount of energy stored in B, 8 \* numOfIntervals + 1

to 9 \* numOfIntervals

1. E\_f(t): the amount of energy stored in F, 9 \* numOfIntervals + 1

to 10 \* numOfIntervals

1. B\_bin(t): mutual exclusive binary variables for battery, 10 \* numOfIntervals + 1 to 11 \* numOfIntervals
2. F\_bin(t): mutual exclusive binary variables for flywheel, 11 \* numOfIntervals + 1 to 12 \* numOfIntervals
3. All these variables have the unit of kWh, except #6 and #8, which are percentages, and 11, 12 are binary numbers.

Objective function:

1. maximize the expected life time of the battery: Period\_Of\_Peak\_Power \* Life\_Cycle \* (DoD\_max\_b / DoD\_b), in the unit of year

The period of peak power is assumed to be 1 minute, the life cycle of the battery is 2 (2000 numbers of discharge), DoD\_max is 0.8

1. Minimize the discharge of battery
2. Maximize the battery storage at each time step.

These three objective functions can be represented using different weights.

Constraints:

1. D\_b(t) + F\_b(t) <= r\_b \* (Max\_Capa\_B / battery\_rate), discharge rate : charge rate of the battery is r\_b
2. B\_g(t) <= Max\_Capa\_B / battery\_rate,, charge rate of B is bounded, fully charge in 20 hours or 1200 minutes, THIS SHOULD BE
3. E\_b(t) <= Max\_Capa\_B
4. E\_f(t) <= Max\_Capa\_F
5. (1 - DoD\_b(t)) \* Max\_Capa\_B <= E\_b(t)
6. (1 - DoD\_f(t)) \* Max\_capa\_F <= E\_f(t)
7. Given (F\_b(t) +D\_b(t) > 0), B\_g(t) = 0: battery cannot be charged and discharged at the same time
8. Given D\_f(t) > 0, F\_b(t) + F\_g(t) = 0: flywheel cannot be charged and discharged at the same time
9. F\_g(t) + D\_g(t) + B\_g(t) <= G(t)
10. DoD\_b(t) <= DoD\_max\_b
11. DoD\_f(t) <= DoD\_max\_f

1. D\_g(t) + D\_b(t) + D\_f(t) = D(t)
2. E\_b(t) = E\_b(t-1) + efficiency\_b \* B\_g(t-1) - (F\_b(t-1) + D\_b(t-1)), the energy stored in B for each time step
3. E\_f(t) = E\_f(t-1) + efficiency\_f \* ( F\_g(t-1) + F\_b(t-1)) - D\_f(t-1) - self\_discharge\_rate\_f \* E\_f(t-1)
4. D\_b(t) + F\_b(t) <= E\_b(t), battery cannot discharge more than it current has
5. D\_f(t) <= E\_f(t), flywheel cannot discharge more than it currenlty holds
6. All variables are greater than 0

Data used:

* <http://egauge5729.egauge.cs.umass.edu/>
* Optimize over certain period of time
* <https://developer.forecast.io/>, predict next day’s/hour’s solar energy availability.
* Predict next day’s/hour’s power consumption requirements.

References:

* Henrik Bindner, Tom Cronin, Per Lundsager, James F. Manwell, Utama Abdulwahid and Ian Baring-Gould, *Lifetime Modelling of Lead Acid Batteries* , Risø-R-1515 Report 2005
* Di Wang, Chuangang Ren, Anand Sivasubramaniam, Bhuvan Urgaonkar, and Hosam Fathy, *Energy Storage in Datacenters: What, Where, and How* much?, SIGMETRICS 2012
* *Technique manual for Sun Xtender Batteries*. Sunxtender Inc. 2014