Interim Report

COMP3003

Eskil Sulen Gjerde

Novel Approaches for Optimal Control of Battery

1 Contents

2 Introduction

This project will focus on an optimization based real-time control of BESS (battery energy storage system) and configuration optimization which is of great interest to Pixii. They are a company that produces modular energy storage solutions using advanced bi-directional power convertors (Pixii, 2020). Since their systems are modular it means that they are very scalable. A configuration optimizer would therefore be useful before installation to find the optimal size for the modular system. It can also be useful to find out when and how the system could be extended if there is a change in the load profile, the prices of products or other factors. The real time controller is beneficial for optimally utilizing their system to reduce the cost as much as possible.

This semester the focus was on understanding the problem and finding and comparing approaches. Next semester will be focused on expanding on the selected approach.

3 Motivation

One of the major challenges Pixii sees for BESS “behind the meter” is that the payback time often is longer than the life-time of the system. Another challenge is the fact that designing good control strategies for BESS is very hard due to the vast variation in complexity and billing structures. Utilizing the full economical potential of a BESS therefore becomes too complicated for a home user due to the need to properly configure the system.

As a result, one of the biggest motivations for this project is to create an algorithmic framework that can be used for three different things:

- Finding the optimal configuration (Pre-installation)
- Controlling the BESS during operation
- Alerting the user if the optimal configuration changes (Post-installation)

Most people who currently invest into BESS do so out of non-economic reasons and not monetary ones (Gretz, 2016). The key non-economical reason are the environmental benefits, since environmentally friendly grids with a large portion of renewable energy (PV/Wind) need storage capacity to be stable (Aurecon, 2020).

There are many people who wish to be environmentally friendly where the BESS making a profit is not one of their main concerns, but their initial investment must break even. For this incentives from governments have been shown to be effective (Gretz, 2016). Pixii’s system is modular and since the prices of battery and power modules are expected to decrease, while tariff prices increase, it is likely that adding modules in the future will then make it a profitable investment. Therefore, finding the optimal configuration and charge/discharge plan is important in order to make the initial investment break even.

There are also other additional benefits to society as a whole. One problem on the horizon is that considerable reinforcements of the grid backbone and distribution grid are needed to cope with increased demand caused by the electrification of society. For home application EVs are the biggest concern (Catapult, 2018). To mitigate this complex problem, grid owners will have to introduce new tariffs designed to optimize the grid infrastructure (Catapult, 2018). Having a smart BESS automatically adapting to these new tariffs could potentially contribute to solving this future societal challenge.

Another motivating factor for this project and one of Pixii's wishes is to get an algorithm that improves on the current state of the art and that can still run on their hardware. This means that there will be some requirements placed on any algorithm used for this purpose.

1. The algorithm needs to be compact enough (memory size).
2. The run-time of the algorithm needs to be good enough so that it finishes between the time intervals of the controller.

Although these requirements are important, they can be seen more as guidelines since the focus of the project is to get a proof of concept and not the final implementation used on the battery system.

4 Problem Description

4.1 Background

In this project there are two main things that should be achieved. A configuration optimizer and an optimisation-based controller. Configuration optimization is where you try to find the best configuration of for example, size of battery, number of solar panels and converter size. The optimisation-based controller can be used for real-time control of a BESS and since it will work with forecasted data, it needs to be able to handle uncertainty.

Both the configuration optimizer and the real-time controller build upon the same problem of finding an optimal charge/discharge schedule for minimizing cost, based on several factors:

- Load profile
- Generation profile
- Tariffs (cost of energy)
- Battery capacity
- Charge/discharge capacity
- Degradation of the battery
- Loss during charge/discharge
- Cost of the system

Tariffs (billing structures) vary from region to region and there are often also variations between customers within the same region. Since tariff structures vary so greatly it is crucial to be able to represent them using a generic notation that can describe any type of tariff.

Billing structures are normally divided into a combination of a fixed part, energy amount-based and peak power based. The energy amount and peak power-based vary depending on time, e.g. peak hour, day of the week and month of the year.

4.2 Complexity

4.2.1 Piece-wise Linear Cost Function

One of the complexities with this problem is that some tariff structures result in piece-wise linear cost functions, one such example can be seen below.

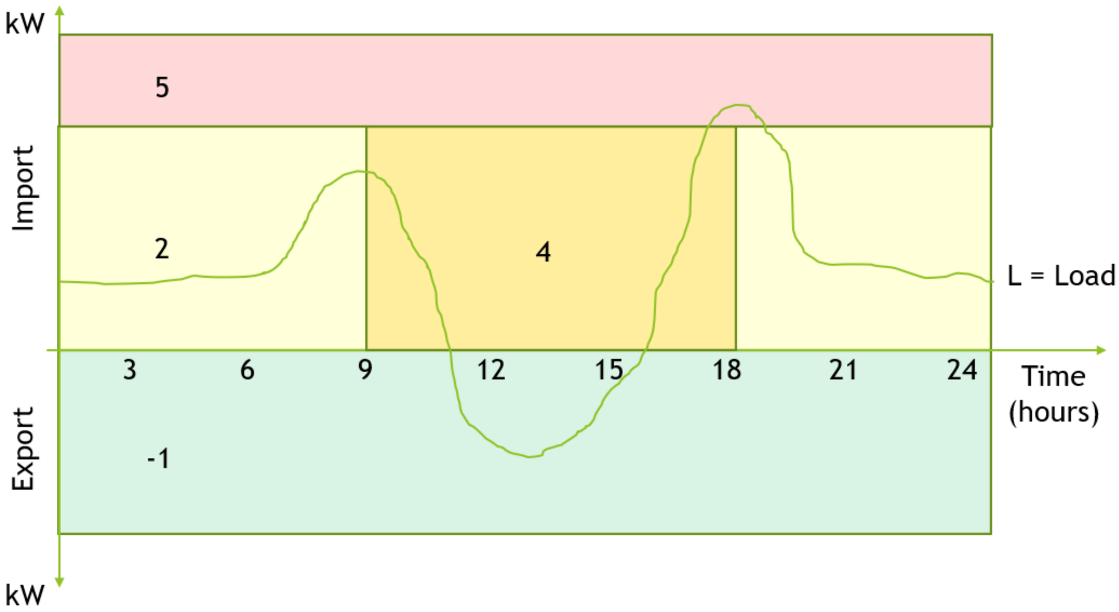

Figure 1. Example of a type of tariff structure

Figure 1 shows that if the system is exporting it gives a profit of 1 all hours of the day, while importing up to a certain threshold costs 2 except during peak hours between 9 and 18, which costs 4. Then there is a threshold where if the load exceeds it at all hours of the day it costs 5. The load at a given time and max BESS charge/discharge capacity determines if the cost function is linear or piece-wise linear. Looking at hour 6 and assuming a load of 60, with a maximum charge/discharge capacity of 10 results in a linear cost function.

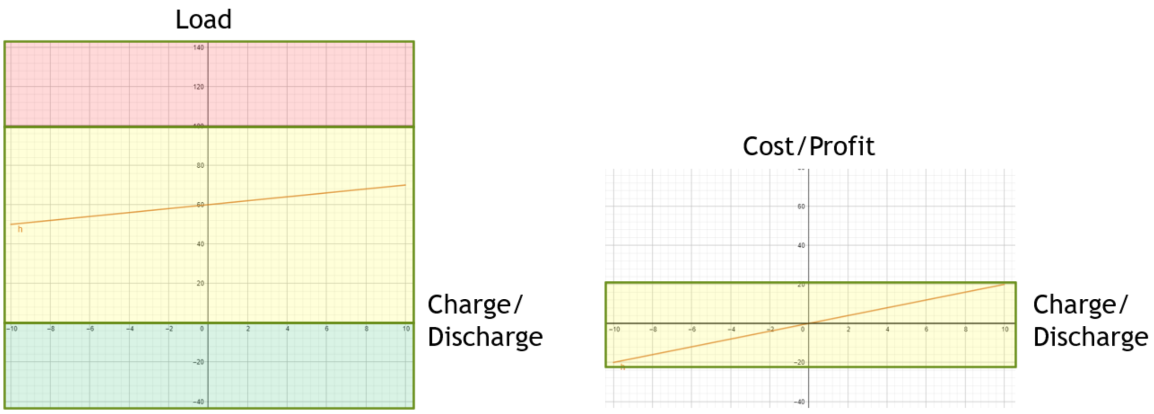

Figure 2. Linear cost function

In the left figure above, you can see the possible range of values (50-70). The important thing to note is that since all these values are within the same tariff price, the resulting cost function is linear.

The problem with the non-linearity in the cost function occurs when the range of possible values covers multiple tariff prices/areas. Using the same example, but with a load of 95 instead of 60 produces the figure below.

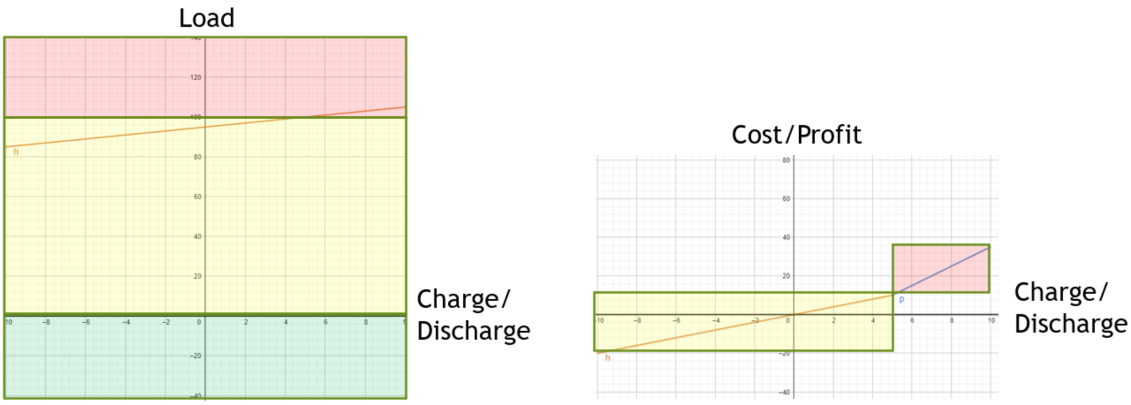

Figure 3. Piece-wise linear cost function

Here the range of values on the left spans across two regions and the corresponding cost function on the right is therefore piece-wise linear.

4.2.2 Search space

To illustrate the complexity of the problem we can look at how the search space can be modelled as a tree. The depth of the tree would be how many time intervals you are trying to find a schedule for and the number of children a node has would be determined by the max charge/discharge capacity. The number of nodes in the tree would be:

$$\text{Number of nodes} = \frac{(maxCD + 1)^{N+1} - 1}{(maxCD + 1) - 1} \qquad (1)$$

maxCD = The maximum amount you can charge/discharge per time interval
N = The total number of time intervals also representing the depth of the tree (counting from 0)

Equation 1. Total number of nodes in search tree

Due to the search space growing exponentially with the number of time intervals it can very quickly become infeasible to solve. This means that solving the problem through brute force is not sensible and performing adequate pruning is therefore required.

4.2.3 Uncertainty

When using the optimisation-based controller for real-time control, there is bound to be uncertainty since it acts on forecasted data. The further into the future the schedule is made for the greater the uncertainty. This can be exploited in order to reduce the computational time of the algorithm. Since having a detailed plan far into the future is redundant, due to the high probability of it changing when new measurements are made. This means that in the short-term making the duration of the time intervals as short as possible is worthwhile, while as you move further and further away the size of the time intervals can increase (Richards and How, 2005). This decreases the number of decisions that have to be made, reducing the overall computational time.

5 Related Work

5.1 Industry Practise

5.1.1 Configurator

There are currently some tools on the market for finding the optimal configuration based on the historical load profile, for example HOMER Grid (HOMER-Grid, 2020). The main focus of tools like HOMER Grid is to find the optimal configuration before installation. It needs to be done manually and by someone with the correct technical knowledge. Although Pixii would want something like this as well, they also want to focus on the after sales market.

5.1.2 Real-time control

There are currently four major BESS control strategies in the market today for "behind the meter applications" which exploit different aspects of tariff structures:

1. **Peak shaving**: discharge when power drawn from the grid exceed a limit that result in some high "penalty" cost. Charge the battery when the power is below the limit. The **SOC** (level of charge of an electrical battery) reference in this strategy is set to high (**100%**).
2. **Time of use:** Charge when the energy cost from the grid is low, and discharge when the cost is high. The **SOC** reference in this strategy is a function of time SOC(t).
3. **Increase self-consumption**: Charge the battery when there is more production than consumption. Discharge as soon as energy is imported from the grid. The **SOC** reference in this strategy is set to low (**0%**).
4. **Time scheduled:** Time scheduled changing between the three above (peak shaving, Increase self-consumption and time of use).

The first three control strategies above have their own benefits and seek to reduce the cost in different ways. The cost reduction achieved from employing only one of the strategies is not enough to pay back the cost of the system. Which is why combining the strategies in an optimal way is so important. Combining them is not straightforward however, since the different strategies have their own SOC reference. For example, if the system is prepared for peak shaving then the SOC reference is at 100% while for increasing self-consumption it is at 0%. The time scheduled control strategy uses all the strategies, but this is not an optimal combination of them since it simply runs one of them for a certain period.

5.2 Scientific

Since BESS can be used for many different purposes the research around it reflects that. Even though the focus related to BESS is often to create an optimal charge/discharge plan, what this optimal plan is varies greatly on the focus of the optimization. There have also been many different types of methods applied to try and solve the problem.

One paper proposes a way of creating an optimal BESS charge/discharge plan using Linear Programming (Chouhan *et al.*, 2016). The issue here is that in order to make the problem linear they have simplified peak shaving and self-consumption. They have created constraints where the load cannot be above a limit or below 0, which means that improving self-consumption is not possible.

MIQP has been used in (Dongol, Feldmann and Bollin, 2018) for reducing the PV feed-in to the grid and grid power consumption. The model tries to minimize an error function where penalties are found using squared error. This means that Quadratic programming is not needed for minimizing cost.

In (Tang and Eghbal, 2018) MILP was used for minimizing cost. It concluded that in all scenarios installing a BESS was not financially beneficial. This paper only looked at TOU tariffs opting to leave out other ways the BESS can reduce cost. The paper also did not consider if the configurations used were optimal.

Another approach (Maly and Kwan, 1995) uses Dynamic Programming to get a charge/discharge schedule minimizing cost by performing peak shaving, load redistribution and prolonging battery life. This algorithm also does not account for PV or self-consumption.

For real-time control, one paper used MPC (Ehsan Raoufat, Asghari and Sharma, 2017) to make the schedule more adaptive. Combining their charge/discharge schedule optimization algorithm which focused on improving PV-utilization with MPC, they saw a 60-80% increase in the utilization. The demand charge savings went down approximately 2% on average when compared to simple rule-based controllers, however. As this paper focuses on improving PV-utilization along with the cost, it is difficult to use this as any indication for BESS being economically feasible or not.

There has also been research done on finding the optimal configuration of PV and energy storage for prosumers (Achiluzzi *et al.*, 2020). The focus here is only on finding the assets with the best price and not how the system would be controlled.

5.2.1 Contributions

The gaps this project sets out to fill in the current existing literature is if combining existing methods can make a BESS economically feasible. There are a lot of papers that focus on topics such as peak shaving, optimizing time of use, increasing self-consumption and configuration optimization, but these papers only focus on one or a few of them at a time. The papers also often focus on other aspects than minimizing cost, such as reducing the power peaks or maximizing self-consumption. Focusing purely on reducing cost by exploiting tariff structures and combining existing methods can therefore give interesting insight into the feasibility of BESS, since existing literature do not include all relevant aspects.

6 Approaches

This section covers two different approaches for solving the charge/discharge schedule problem. The Dynamic Programming approach was used during the initial phase of the project, where the focus was on getting familiarized with the problem, because it is an intuitive way to solve the task. The MILP approach was chosen since it was successfully applied to similar problems in literature and since the configuration optimizer was added to the scope of the project.

6.1 Dynamic Programming

"*Dynamic programming is an optimization approach that transforms a complex problem into a sequence of simpler problems; its essential characteristic is the multistage nature of the optimization procedure*" (Böhme and Frank, 2017)

For the problem of creating a charge/discharge plan, the sub-problems are deciding at a given time interval what the optimal decision is to reach each of the different states. Different states represent different levels of energy in the battery (SOC, state of charge). For example, a BESS with capacity 60 kWh with one state per kWh will have 61 different legal states (0-60). How many states are reachable from a given state is determined by the maximum charge/discharge capacity (power rating of the converter in the system) for a time interval. Below is a figure that illustrate some of these points using a max capacity of 4 and max charge/discharge capacity of 2.

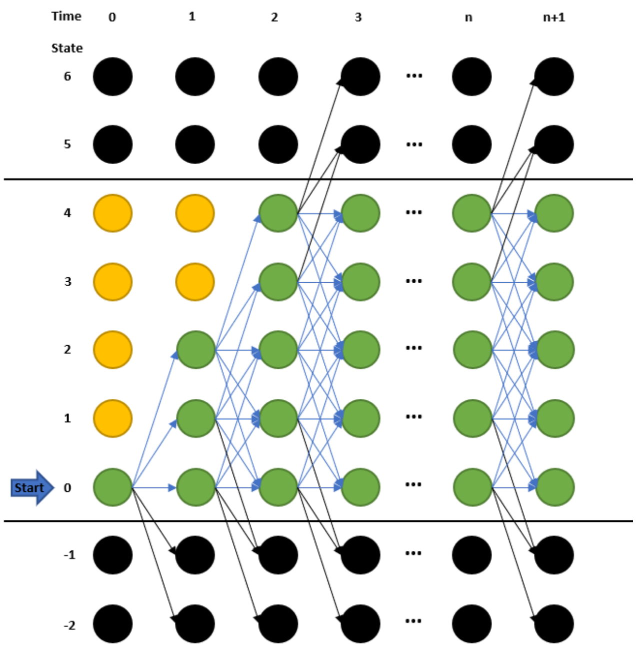

Figure 4. Dynamic Programming approach

All states/nodes between the two black lines (representing the smallest and largest amount of energy in the BESS) are legal states and coloured in green. The yellow states represent legal, but "unreachable" states, which occurs when the algorithm starts. When it begins a start point is given at time 0, which in this case is 0. This means that since the maximum number of states it can move up/down is 2, the reachable states are (-2, -1, 0, 1, 2), meaning states 3 and 4 are not reachable yet. States such as (-2, -1, 5, 6) are coloured in black since they are illegal as they exceed the boundaries of the battery in the BESS.

In the diagram you can also see the number of computations needed for a sub-problem illustrated by the arrows between time interval n and n+1. The number of computations between time intervals remains the same (except the sub-problems at the start with unreachable states), since the number of legal states and the number of states reachable from the different states remains the same. This is a very efficient form of pruning of the brute-force search tree. Here an entire sub-tree is discarded if one of its nodes is an illegal node. Due to this pruning the computational time grows linearly with the number of time intervals. Figure 5 explains in detail how the optimal path to a state is found for a single sub-problem/time interval.

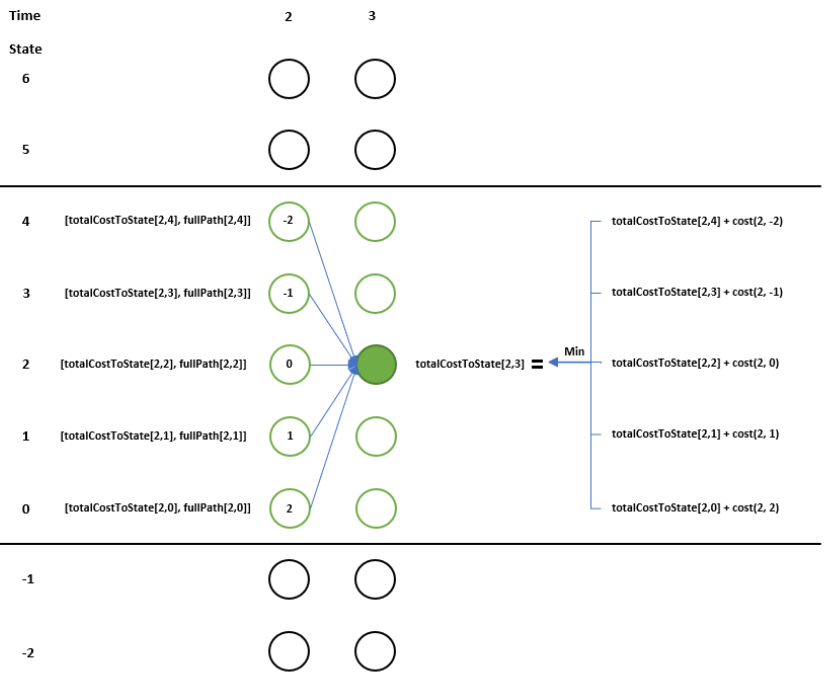

Figure 5. How optimal path is chosen for a state

One important thing to stress is the fact that the minimal cost and optimal path is stored for all legal states in a time interval. The example above shows which values/paths are considered when picking the minimal cost for reaching state 2 at time 3. To explain the figure further it is important to first explain the elements in the diagram:

- **totalCostToState[T, S]:** Contains the smallest possible total cost for reaching state S at time T. These are the solutions to the previous sub-problems where the solutions of T – 1 is used to solve sub-problem T.
- **cost(T, A):** The cost of discharging/charging amount A at time T. Things like tariffs and load/production will vary with T and affect the cost.
- **fullPath[T, S]:** Contains the optimal path to state S at time T. The optimal path can be represented by for example, the amount discharged/charged per time interval.

Since there are in this case 5 different ways of reaching state 2 at time 3, then it means we have 5 values we need to calculate and compare. The values inside the states represent how much should be charged/discharged in order to reach state 2. The total costs are found by taking the total cost of reaching the state you're coming from plus the cost of going from that state at time 2 to state 2 at time 3. After all these values have been found the smallest one is selected and set as the total cost to state 2 at time 3. This process is repeated for all the states at time 3. You would then move to the next time interval and repeat the same process.

```
initialize the best routes and best costs for all states and all time intervals to empty (null/0)
initialize starting SOC, max capacity of battery, max charge/discharge per time interval, tariffs
for i :=0  to numberOfTimeIntervals
    set temp best routes and temp best costs to empty (null/0) for all states
    for j := 0 to number of states
        if the state is not reachable yet continue
        for k:= -max charge to max charge
            tempState := j + k
            if the state is illegal continue
            bestOldStateCost = smallest total cost to reach state j in time interval i-1
            transitionCost := cost going from state j in time interval i-1 to tempState in time interval i based on j, k and tariffs for i
            tempCost := bestOldStateCost + transitionCost
            if tempCost is better than any other route found to tempState in time interval i or no route has been checked so far
                the temp best cost to tempState in time interval i is updated
                the temp best route it updated to be best route to state j at time interval i-1 plus adding j->j+k at the end
    for m := 0 to number of states
        if state not reachable yet continue
        best route and best cost for state m at time interval i is updated to the temp best cost and temp  best route found
```

Figure 6. Dynamic Programming Pseudo-code

Above is the pseudocode for the Dynamic Programming algorithm.

6.2 MILP

MILP is a way of algebraically formulating a problem in order to find the best possible solution. Below is the algebraic formulation for a MILP model intended to find the optimal charge/discharge schedule in addition to finding the optimal battery size and max charge/discharge capacity for the BESS.

$$\begin{array}{lllr}
\text{Min} & Z = \sum_{i=0}^{N}\sum_{j=0}^{PL}(T_{ij} * x_{ij}) + BSCost * bS + CDCCost * cDC & & (1) \\
\text{subject to} & bS \leq MaxBatterySize & & (2) \\
& cC \leq MaxChargeCapacity & & (3) \\
& x_{ij} \leq Limit_j & i = 0, ..., N \;\; j = 1, ..., PL & (4) \\
& x_{i1} \leq |Load_i| & i = 0, ..., N & (5) \\
& d_i \leq cDC & i = 0, ..., N & (6) \\
& d_i \geq -cDC & i = 0, ..., N & (7) \\
& -x_{i0} + x_{i1} + x_{i2} = Load_i + d_i & i = 0, ..., N & (8) \\
& StartCharge + \sum_{j=0}^{i} d_i \geq 0 & i = 0, ..., N & (9) \\
& StartCharge + \sum_{j=0}^{i} d_i \leq bS & i = 0, ..., N & (10) \\
& x_{ij} \geq 0 & i = 0, ..., N \;\; j = 0, ..., PL & (11) \\
& d_i \in \mathbb{Q} & i = 0, ..., N & (12) \\
& bS, cC \in \mathbb{N} & & (13)
\end{array}$$

N = Number time intervals
PL = Number of price levels (tariff levels)
bS = Battery size
BSCost = The cost of increasing battery size
cDC = Capacity of Discharge/Charge per time interval
CDCCost = The cost increasing charge/discharge capacity
StartCharge = Starting charge in the battery
MaxBatterySize = Max battery size possible for the model to find
MaxChargeCapacity = Max discharge/charge possible for each time interval that the model can pick
Limit = An array containing the limits where boundaries where a new tariff starts
Load = An array containing the load + production for all time intervals

Figure 7. MILP model

One notable thing to mention about this model is how it gets around the problem that the cost functions can be piece-wise linear. This is an issue since MILP models can only represent linear relationships. The problem was circumvented by having one decision variable per price level in each time interval. The decision variables are then paired with a coefficient corresponding to the cost/profit of that range.

Due to the introduction of the extra decision variables, there are some additional requirements needed, with the most crucial one being (8). It ensures that all the x decision variables add up to the load plus the BESS charge/discharge (d) amount for that time interval.

In order for the proposed model to work there are certain assumptions made:

- Tariff prices for the different levels are in ascending order.
- The value earned from exporting is less than the smallest value it costs to import.

These are assumptions made for the current model to keep it as simple as possible, but they can be worked around by including some additional requirements later if necessary.

One of the main reasons why a MILP approach is being used is that it works well for finding the best configuration. As can be seen in the objective function (1), there are currently no configurations being chosen from, but instead there is a cost for the size of the battery and maximum BESS charge/discharge capacity. Changing this to be separate configurations instead of two decision variables is trivial and can be done by introducing some binary decision variables and some other minor adjustments. It is not included as of yet due to the fact that data (prices, lifetime etc) need to be gathered first. There are also other things that will potentially be added to the model in the future, for example controllable loads and multiple power options.

6.3 Testing

The implementations for the two approaches were tested using a set of different varied datasets. The optimal cost savings were then found by both approaches on the different datasets. There is then an assumption made that if both the approaches obtain the same optimal cost savings, then they are both implemented correctly. This is a safe assumption to make as they differ so greatly in how they work, and it is therefore very unlikely that they both reported the same wrong values.

One thing to note is the fact that even though they report the same optimal cost reduction, there will be multiple optimal solutions.

6.4 Comparison

6.4.1 Computational time

One key comparison to do between the DP and MILP approaches is how scalable they are in terms of runtime. Below are two graphs showing how the runtime in ms scales as the number of time intervals increases.

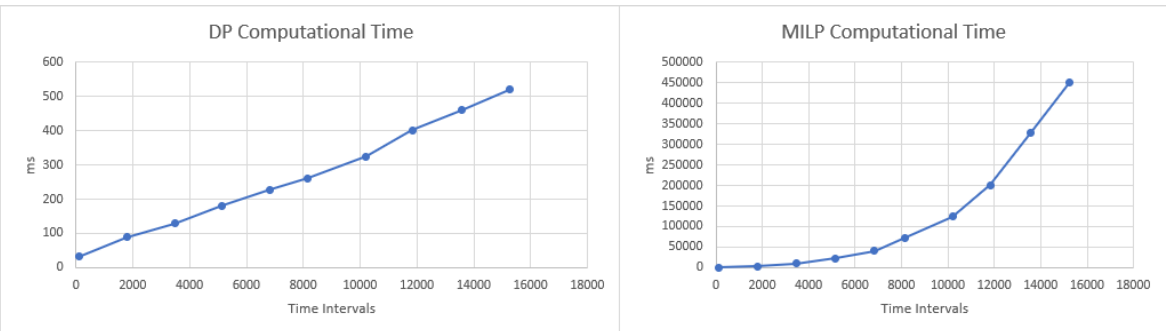

Figure 8. Computational time for DP and MILP

As can be observed from the graphs the computational time grows linearly with the number of time intervals for the DP approach. This is expected based on reasons previously explained. For the MILP approach computational time seems to be growing exponentially with the number of time intervals/decision variables. Currently the number of decision variables in the simple MILP model found in figure 7 increases by four for each additional time interval.

To put the time interval values into perspective, the computational time needed for different numbers of time intervals is paired with how far into the future the plan lasts for different time interval lengths.

Time Intervals	DP Computational Time	MILP Computational Time	15 Seconds	1 Minute	15 Minutes	1 Hour
1000	~0.05 Seconds	~0.5 Seconds	~5 Hours	~17 Hours	~10 Days	~41 Days
6000	~0.2 Seconds	~40 Seconds	~1 Day	~4 Days	~62 Days	~250 Days
15 000	~0.5 Seconds	~7 Minutes 30 Seconds	~2 Days 12 Hours	~10 Days	~156 Days	~625 Days

Table 1. Computational time vs schedule length

As can be seen in the table the DP algorithm is able to find the optimal schedule for 15 000 time intervals in 0.5 seconds, something that takes around 7 and a half minutes for the MILP approach. The complexity of the MILP will also increase in the future introducing more decision variables, which will further increase the computational time.

6.4.2 Configuration

Due to the nature of how pruning is performed in the DP approach, integrating configuration optimization into it is not possible. The reason for this is that even though a more expensive configuration might be better in the long run, if a cheaper alternative performs just as well in the short term, then all solutions with the more expensive option will be discarded and not explored.

This means that for configuration optimization MILP is the favoured choice, since it can be integrated into the model, while for DP you would have to run the same algorithm multiple times, once per configuration.

6.4.3 Continuous vs Discrete decisions

The DP approach works by having a set number of discrete states the battery can be in, representing different levels of energy. This means that when the battery is charging/discharging it needs to be exactly enough to end up in a legal state. Meaning if the distance between states is 1 kWh, then it is not possible to charge/discharge 0.5 kWh only multiples of 1. This is not an issue with the MILP approach as decision variables in this case are continuous and not discrete.

The advantage with having continuous values is that the charge/discharge schedule will be slightly better since it has more control over the battery. Although this difference can be made insignificant by having an adequate number of states, there is a trade-off however, where the computational time grows with the number of states.

6.4.4 Extendibility & Maintainability

An important aspect when choosing which approach to use is how easy it is to maintain and extend. Since a MILP model is represented by compact algebraic formulations an understanding of how it works can quickly be reached. The fact that implementing the model is trivial once the model is defined is another great benefit.

For the DP approach there is a lot more code that must be understood and there is no clear innate model to look at. Extending the DP algorithm is also more difficult than the MILP model, since many things would result in the algorithm not being able to guarantee optimality due to how the pruning works. The MILP model on the other hand is easier to extend as you can add new and modify existing requirements.

6.5 Conclusion

Two initial approaches for solving the problem have now been compared in order to find the most appropriate one for further work. Based on initial testing the DP approach is the faster alternative creating a charge/discharge schedule for 15 000 time intervals 900 times faster. It has it's own clear disadvantages however, where it is not possible to integrate configuration, it is less extendable and it is restricted to a set of discrete values for charging/discharging. If the real-time controller requires the computational time to be this fast, then the drawbacks are manageable.

The main advantages of the MILP approach are that it can be used to solve the configuration problem and can easily be extended. There is also the benefit that the continuous decision variables allow for greater control of the BESS. Based on the fact that the BESS normally controls energy in a one hour time period, it means that the dynamic response time required of the real-time controller is low. Therefore, the additional computational time needed by the MILP approach is not likely to be of consequence. Although reducing the computational time might not be necessary, it can be done and it is worth focusing on.

The conclusion is therefore to continue improving and extending the MILP model to fulfil the project objectives, where the Dynamic Programming approach will be a backup in case unforeseen problems arise with the MILP approach.

7 Future Directions

The research field around BESS is vast and there are a lot of things that can be done that further expands on the work that will be done in this project.

One key problem that can be looked at is how to deal with uncertainty in forecasted data. There are a multitude of ways to approach this problem and it's important, since without dealing with it, the quality of the charge/discharge schedule will heavily rely on the accuracy of the forecast.

With this project the focus is on a behind the meter application of a BESS for home use. Finding out if the same principles that gouvern how you optimize for this scenario apply for other scales is another interesting idea.

8 Progress & Reflection

This section covers progress made during the first half of the project in addition to future plans and reflection on how things have gone so far.

8.1 Objectives

Pixii's initial aim for the project was to create an algorithm based on tariffs for adaptive real time control that optimizes peak shaving, increased self-consumption and time of use by creating a plan based on a forecast and adapting based on real time data. A configuration optimizer that can find the optimal configuration should also be developed.

To make the project more scientifically interesting and challenging controllable loads and additional power sources have been added as objectives for the configuration optimizer.

Objectives:

1. **Real time controller:**
   - a) Explore and implement methods (e.g. dynamic programming and MILP) for finding the optimal charge/discharge plan based on historical data/forecasts and tariffs. The tariffs should be represented in a generic compact rule-based format. Should also identify the pros and cons of the different methods.
   - b) Create an algorithm that can generate simple forecasts for load and production based on historical data.
   - c) Create algorithm to run a charge/discharge plan on a dataset. Useful for comparing the optimal plan obtained from the actual data to the plan you get from the forecast. Which can be used as baseline for later benchmarking.
   - d) Explore and implement a method for combining a forecast and reacting in real time to measured data, for example MPC or other similar rolling time horizon approaches. Real time data must be included to mitigate the effects of uncertainty in the forecast.
2. **Configuration optimizer** (for example PV, battery size and converter size)
   - a) Expand the charge/discharge plan optimizer algorithm to take different PV configurations as input and selecting the optimal one.
   - b) Expand configuration to include things like battery size and converter size.
   - c) Include an option in the configuration optimizer to take into account how much the customer values environmental benefits in addition to economical ones.
   - d) New: Add additional features such as controllable loads and additional power sources.
   - e) New: Create GUI for demonstration purposes.

8.2 Project Management

This project was designed in such a way that the focus of the first semester was to get proof of concepts working for the two approaches. The reason why getting some simplified versions working was important, was to see if it was possible to use the approaches for solving the problem. They had to be simplified due to time constraints caused by me doing a 70/50 split. It would also have been a big risk to only focus on one approach during the first semester, because if the other approach would not have worked when I later started, I would have had much less time to adjust my plan.

Originally there was an equal amount of importance placed on the real-time controller and configurator. Although the real-time controller is something Pixii wishes, after discussions with my supervisor we came to the conclusion that it was not ambitious enough and not as interesting as the configurator from a scientific perspective. This means that there were some adjustments that had to be made to the original project plan.

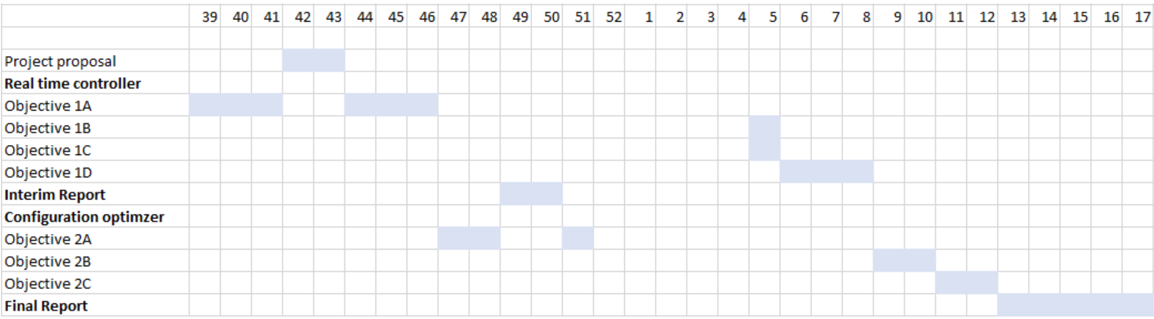

Figure 9. Original project plan

Above is the original project plan and as mentioned above the first semester consisted of creating some proof of concepts. Although some of the project plan changed, this part of the project remained the same, except for some extra time needed for the interim report. I found the workload assigned to this period to be an appropriate amount. As mentioned previously the amount of work I could do during the start of this project was reduced due to the 70/50 split, which means that workload will be greater in the second half.

When it comes to the adjusted project plan, the key things that have been modified are the order of some of the tasks due to priorities changing and the addition of some tasks/objectives.

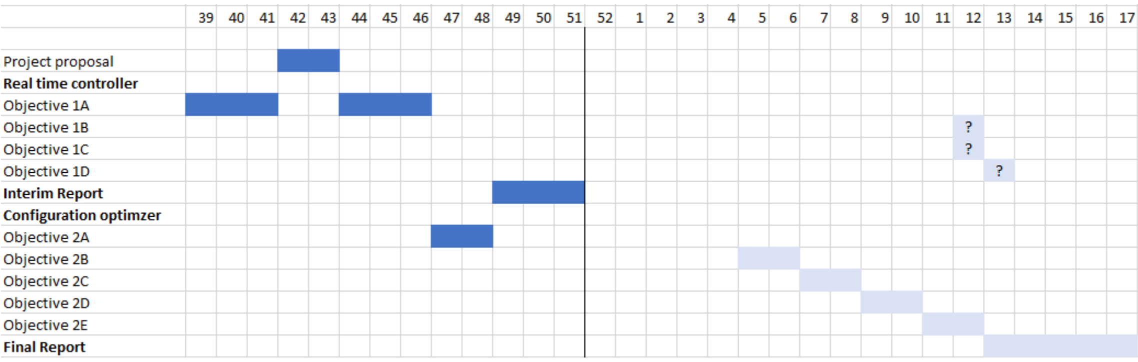

Figure 10. Updated Project Plan with Work Performed

In this updated project plan the new tasks are additional ways of extending the configurator and a GUI for demonstration purposes. These new extensions for the configurator were added in order to make it more scientifically interesting and since more of the project will now focus on the configurator, time that was previously allocated to the real-time controller was freed up for other purposes.

Another reason why these changes were made, was to mitigate potential risks in the project. In order to set up a BESS with real-time control, assistance from Pixii HW/SW engineers is needed. To do this efficiently I had planned to visit Pixii's lab to get access to hardware and help. Due to COVID-19 this can be challenging. The goal is still to get some work done on the real-time controller, but it will only be done if time allows it, which is why those objectives have question marks in the project plan.

8.3 Reflection

Although some modifications were needed for the project plan, the work I set out to accomplish has been completed. There is a lot of work that still needs to be done, but I have a clear plan of what I want to achieve during the second half of the project. Where after comparing the DP and MILP approaches, I concluded that the MILP approach is the most suitable one for creating a configurator and will therefore be used for future work.

For myself, this project is great opportunity to work with a real-world problem from industry. It is also an opportunity for me to put knowledge I have gained from the Linear and Discrete Optimization module to practise. I'm also learning a lot about optimization which is an area of computer science that greatly interest me, and I'm excited to continue work on the project during the next semester.

9 Bibliography

Achiluzzi, E. *et al.* (2020) 'Optimal asset planning for prosumers considering energy storage and photovoltaic (PV) units: A stochastic approach', *Energies*, 13(7), pp. 1–20. doi: 10.3390/en13071813.

Aurecon (2020) *Battery energy storage's role in a sustainable energy future*. Available at: https://www.aurecongroup.com/markets/energy/hornsdale-power-reserve-impact-study.

Böhme, T. J. and Frank, B. (2017) 'Dynamic programming', *Advances in Industrial Control*, (9783319513157), pp. 199–214. doi: 10.1007/978-3-319-51317-1_6.

Catapult, E. S. (2018) 'Preparing UK Electricity Networks for Electric Vehicles Report', pp. 1–29. Available at: https://es.catapult.org.uk/wp-content/uploads/2018/10/Preparing-UK-Electricity-Networks-for-Electric-Vehicles-FINAL.pdf.

Chouhan, S. *et al.* (2016) 'DER optimization to determine optimum BESS charge/discharge schedule using Linear Programming', *IEEE Power and Energy Society General Meeting*, 2016-Novem. doi: 10.1109/PESGM.2016.7741576.

Dongol, D., Feldmann, T. and Bollin, E. (2018) 'A model predictive control based peak shaving application for a grid connected household with photovoltaic and battery storage', *SMARTGREENS 2018 - Proceedings of the 7th International Conference on Smart Cities and Green ICT Systems*, 2018-March(Smartgreens 2018), pp. 54–63. doi: 10.5220/0006685300540063.

Ehsan Raoufat, M., Asghari, B. and Sharma, R. (2017) 'Model predictive BESS control for demand charge management and PV-utilization improvement', *arXiv*.

Gretz, A. (2016) *Here's Who's Buying Home Batteries And Why*. Available at: https://www.swellenergy.com/blog/here-s-who-s-buying-home-batteries-and-why/.

HOMER-Grid (2020) *Model hybrid energy systems to evaluate multiple options and determine the most economic design*. Available at: https://www.homerenergy.com/products/grid/index.html (Accessed: 23 October 2020).

Maly, D. K. and Kwan, K. S. (1995) 'Optimal battery energy storage system (BESS) charge scheduling with dynamic programming', *IEE Proceedings: Science, Measurement and Technology*, 142(6), pp. 453–458. doi: 10.1049/ip-smt:19951929.

Pixii (2020) *How does it work?* Available at: https://www.pixii.com/this-is-how-the-technology-works/ (Accessed: 23 October 2020).

Richards, A. and How, J. (2005) 'Mixed-integer programming for control', *Proceedings of the American Control Conference*, 4, pp. 2676–2683. doi: 10.1109/acc.2005.1470372.

Tang, D. H. and Eghbal, D. (2018) 'Cost optimization of battery energy storage system size and cycling with residential solar photovoltaic', *2017 Australasian Universities Power Engineering Conference, AUPEC 2017*, 2017-Novem, pp. 1–6. doi: 10.1109/AUPEC.2017.8282386.

10 Key Terms

- BESS (Battery Energy Storage System)
- PV (Photovoltaic)
- SOC (State Of Charge)
- LP (Linear Programming)
- MILP (Mixed Integer Linear Programming)
- MIQP (Mixed Integer Quadratic Programming)
- MPC (Model Predictive Controller)