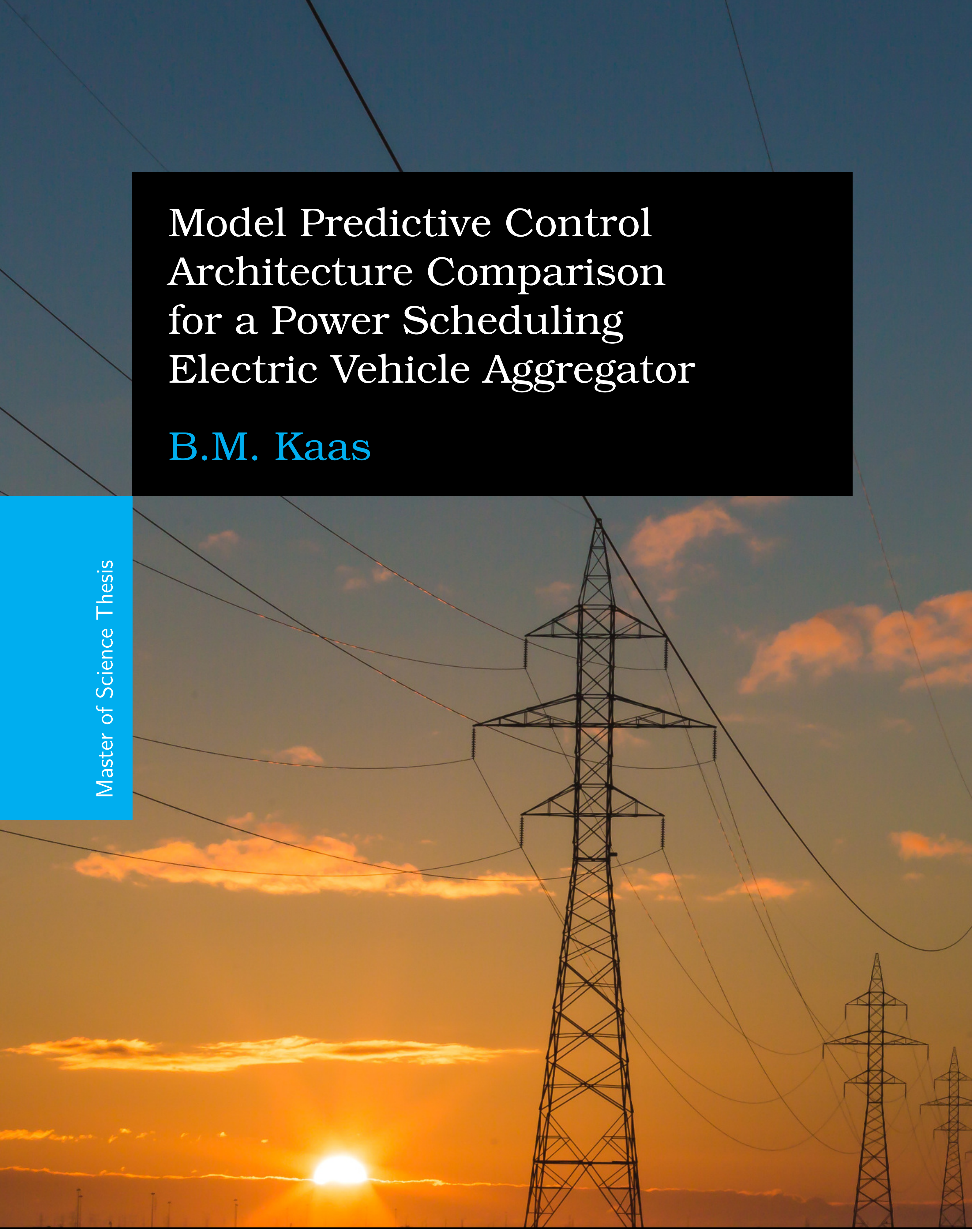


Model Predictive Control Architecture Comparison for a Power Scheduling Electric Vehicle Aggregator

B.M. Kaas

Master of Science Thesis

Delft University of Technology

Delft Center for Systems and Control

Model Predictive Control Architecture Comparison for a Power Scheduling Electric Vehicle Aggregator

Master of Science Thesis

For the degree of Master of Science in Systems and Control at Delft University of Technology

B.M. Kaas

June 24, 2017

Faculty of Mechanical, Maritime and Materials Engineering (3mE) · Delft University of Technology

The work in this thesis is the result of a collaboration with the Netherlands Organisation for Applied Scientific Research TNO.

Abstract

Energy flexibility is the ability to change power production or consumption over time. It is required for a power system to function properly, to balance supply and demand. Currently, the largest providers of energy flexibility in the Netherlands can be found on the supply side and consist mainly out of fossil fueled power plants. As these are set to phase out in the near future and be replaced by mainly variable renewable energy sources, such as wind and solar, the necessity for energy flexibility on the demand side is set to be increased. Therefore, a new role is expected to arise in the power system, namely that of the aggregator. Aggregators will combine small scale energy flexibility providers and provide this aggregated energy flexibility to the power system.

In this MSc thesis, the focus lies with a power scheduling Electric Vehicle (EV) aggregator. Considering that the number of EVs will increase in the near future, the lack of control over charging a large fleet of EVs may result in an overloaded distribution grid. The charging behavior of a large fleet of EVs, connected in a Vehicle to Grid (V2G) setting, is formalized as a Model Predictive Control (MPC) optimization problem. Allowing power consumption to be shifted within a finite prediction horizon. To optimally valorize the energy flexibility of the fleet and respect the limits of the distribution grid, the control problem is extended to include spatial information and network constraints. The goal is to develop multiple control algorithms to solve this control problem using distributed optimization.

The contributions of the work in this MSc thesis are threefold. First, the ability to optimally valorize the energy flexibility is increased by including spatial information in the distribution grid, represented as subsets of the fleet, such that congestion management services can be provided. Secondly, a parallel implementation of a coordinated distributed MPC is developed using resource allocation with feasible iterations for binary on/off input systems. Thirdly, a hierarchical MPC algorithm is developed using virtual batteries to represent the aggregated behavior of a fleet of EVs, for which new tight constraints are derived to better represent the EV fleet.

To conclude, numerical experiments are performed in closed loop to study the behavior of the developed algorithms with respect to a centralized benchmark. The experiments show that for a growing EV fleet, the hierarchical algorithm remains at the same approximate error with respect to the benchmark. This, while the distributed algorithm approaches the benchmark very well, with limited communication and in relatively short computation times with respect to the benchmark. For a growing number of subsets using the same amount of EVs, the hierarchical algorithm is able to come up with a feasible solution reasonably fast. Whereas, the distributed algorithm shows a drastic decrease in computation time as multiple smaller problems are now solved. Both algorithms achieve this at increasing costs. Future work is expected to further improve the hierarchical algorithm such that it will be able to outperform the distributed architecture in practical applications.

Table of Contents

List of Figures

Acknowledgements

I would like to thank my supervisor Dr.ir. T. Keviczky for his guidance during the writing of this thesis. I would also like to thank Farid and Bob as my day to day supervisors. Farid has been able to steer me into clear waters when I was heading into a storm and we have had many discussions providing me with new insights. My TNO supervisor Bob has been able to teach me a great deal with respect to energy flexibility modeling and the challenges related to the future energy system.

Thank you.

Delft, University of Technology
June 24, 2017

B.M. Kaas

Chapter 1

The Power Scheduling Electric Vehicle Aggregator

This chapter motivates the aggregation of energy flexibility of a fleet of Electric Vehicles (EVs) and their control by means of power scheduling. By the aggregation, the charging costs can be minimized and the ability arises to provide congestion management services, therefore value is created. The power scheduling problem for an electric vehicle aggregator is introduced and the chapter concludes with the statement of research objectives and contribution, based on the conclusions drawn in the literature survey, and the outline of the rest of this thesis.

1-1 Changes in the Energy System

In conventional electric power grids, large electricity generation units meet the fluctuating demand by adjusting their power output. With the transition towards variable renewable energy sources, e.g. wind and solar energy, the power grid faces new challenges. As renewable generation is highly dependent on weather conditions, its predictability and controllability is limited and plants are smaller and distributed in general. The result is that the ratio between controllable and uncontrollable generation (i.e. renewables) will therefore decrease. This causes a decrease in supply flexibility, which in turn challenges the balance between supply and demand.

Another trend in the power system is the increasing electric load, with transportation shifting towards electric, demand is expected to increase, which especially challenges the maximum power rating of the lower voltage grids [1, 2].

1-1-1 Creating Value through Aggregating Energy Flexibility

The ability to shift or change power consumption or production over time is defined as *energy flexibility* [3], and simply referred to as flexibility within this thesis. An example of energy

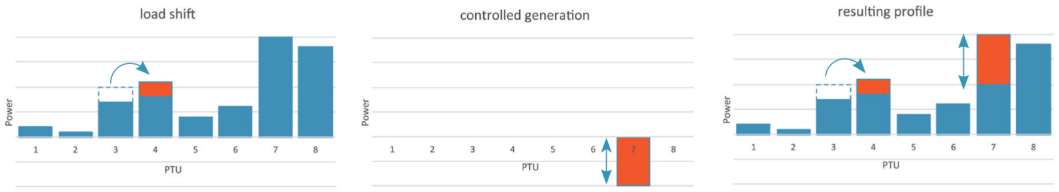


Figure 1-1: Energy flexibility in two forms, 1) a load shift and 2) a temporary discharge into the grid, with the resulting profile on the right, with PTU as Program Time Unit, from [3]

flexibility is shown in Figure 1-1. It can be understood as the flexibility available within the electricity system to match supply and demand, at all times. This balance in the supply and demand is required for an electricity system to function properly. Currently, the largest providers of energy flexibility in the Netherlands can be found on the supply side and consist mainly out of fossil fueled power plants. As these are set to phase out and replaced by mainly variable renewable energy sources, such as wind and solar, the necessity for energy flexibility on the demand side is set to be increased. The latter even further increasing demand energy flexibility due to their limited ability to adapt their power production over time.

The increase of the flexibility demand was studied in [4] and is shown in Figure 1-2 as a function of the participation of variable renewable energy sources in the electricity supply. As a result, a flexibility gap arises. One of the options to introduce new flexibility means is by aggregating large numbers of small flexible resources, of which the energy flexibility is currently not fully utilized.

The aggregation of small demand flexibility, also called demand response or demand side management, may include the power for heating, ventilation, air conditioning, home electricity storages, and many others. By aggregating these small flexible resources, a portfolio of energy flexibility can be created that can be used to act on the arising flexibility gap. It is expected that a new role will arise within the energy system, namely that of the *aggregator* [3]. The aggregator will be responsible for the management of a portfolio of energy flexibility providers. Furthermore, the aggregator will valorize the aggregated flexibility of its portfolio.

The main potential customers of an aggregator are mentioned by the Universal Smart Energy Framework (USEF) in [3] and are shown in Figure 1-3. The flexibility provider is shown on the left, the aggregator in the middle and the potential customers on the right. The aggregator will financially reimburse the providers, here prosumers, for offering their flexibility and potentially sell the aggregated flexibility to the customers. Here, selling flexibility means shifting or changing the original consumption or production schedule to the request of the customer. By aggregating energy flexibility, new flexibility becomes available to the energy system and thus value is created.

The potential customers include Balance Responsible Parties (BRPs), which can use the energy flexibility for trading on spot markets or for their portfolio optimization. Distribution System Operators (DSOs) can use it for grid management. Transmission System Operators (TSOs) can use it for maintaining balance between supply and demand, i.e. frequency management. Since the balance management, which the TSOs are interested in, is typically not dependent on spatial distribution and market based, an aggregator can bid its flexibility into these reserve markets to help the TSO.

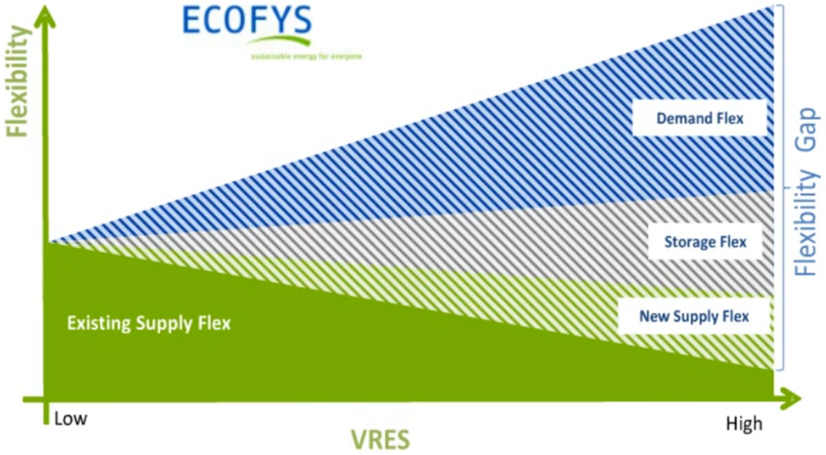


Figure 1-2: With growing variable renewable energy source participation, existing supply flexibility decreases and the necessary flexibility grows, creating a flexibility gap, from [4]

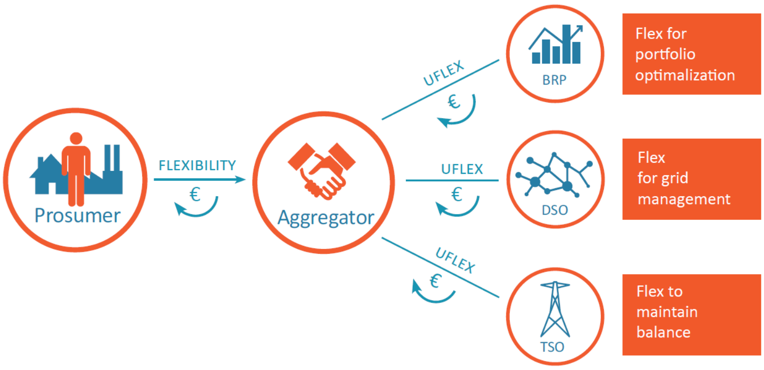
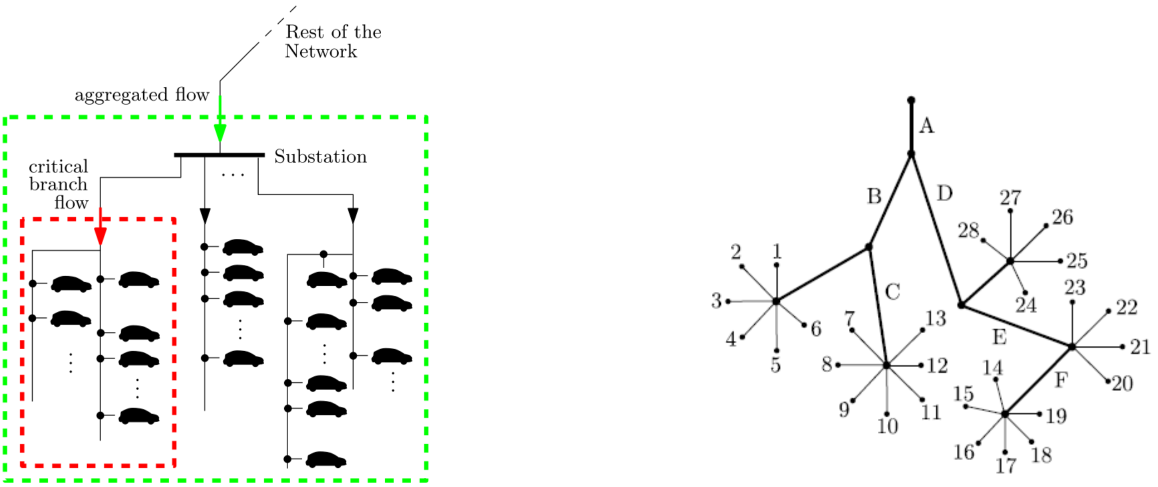


Figure 1-3: The USEF model showing the potential customers of flexibility services, from [3]

1-1-2 Using Energy Flexibility for Congestion Management

The main focus among the potential customers for this thesis lies with the DSOs, which are set with the responsibility to ensure the availability of the distribution grid (medium and low voltage). The low voltage distribution grid is especially expected to be loaded closer to its limits, as a result of the added demand of EV charging, [1, 2]. To ensure the availability of the distribution grid, a DSO needs to prevent congestions from occurring. Which happens when a part of the infrastructure, e.g. a transformer or distribution line, is loaded close to its limits and as such might fail. These will be referred to as *congestion points*. When a congestion point overloads, grid security, e.g. a fuse, will intervene. This is done for safety reasons, and potentially increases the load on the surrounding infrastructure.

To prevent congestions from occurring, DSOs could use *congestion management* by energy flexibility. The goal of congestion management is to prevent congestions from occurring, by using price mechanisms and market forces. Since a power scheduling aggregator can control the power interaction of its portfolio, it can incorporate congestion management constraints



(a) Schematic representation of part of the distribution grid, from [5]

(b) Schematic representation of overview of the distribution grid, from [6]

Figure 1-4: Schematic representations of the distribution grid

within its optimization. Here, power interaction is both the potential power consumption and production of the portfolio.

A schematic representation of the distribution grid is presented in Figure 1-4, with both a detailed figure for a single substation and the larger overview of the distribution grid. In the shown detailed part of the distribution grid, EVs are connected to branches and the branches are connected to the substation, both can act as a congestion point, in which the latter is simply the combination of the branches. In the example, the substation is shown to have a critical branch flow, i.e. this branch is close to congestion. The detailed part is connected to the rest of the network as shown in the overview, in which the numbers indicate the branches, connected to substations. The lower substations are then connected to higher voltage substations. To be able to provide a congestion management service to DSOs, the aggregator should know at which congestion point energy flexibility providers are connected. This is incorporated in the aggregator setting by including subsets. Each subset represents all energy flexibility providers connected to the corresponding congestion point. In the literature, spatial information is often disregarded in aggregator problem formulations [6, 7] and only a single overall network constraint is enforced. Yet, from the viewpoint of an aggregator, especially this spatial information can be valuable, as it can be used to provide congestion management services.

1-2 The Electric Vehicle Aggregator

In the setting of interest, the aggregator has direct control over a fleet of EVs. The EV owners set the constraints, e.g. desired departure time and desired State of Charge (SoC), that have to be met for their EV. Yet, the aggregator decides the power schedule over time by which it will charge the EV. This allows the aggregator to optimally schedule the power demand of the fleet according to varying prices and power availability. By using direct control, the

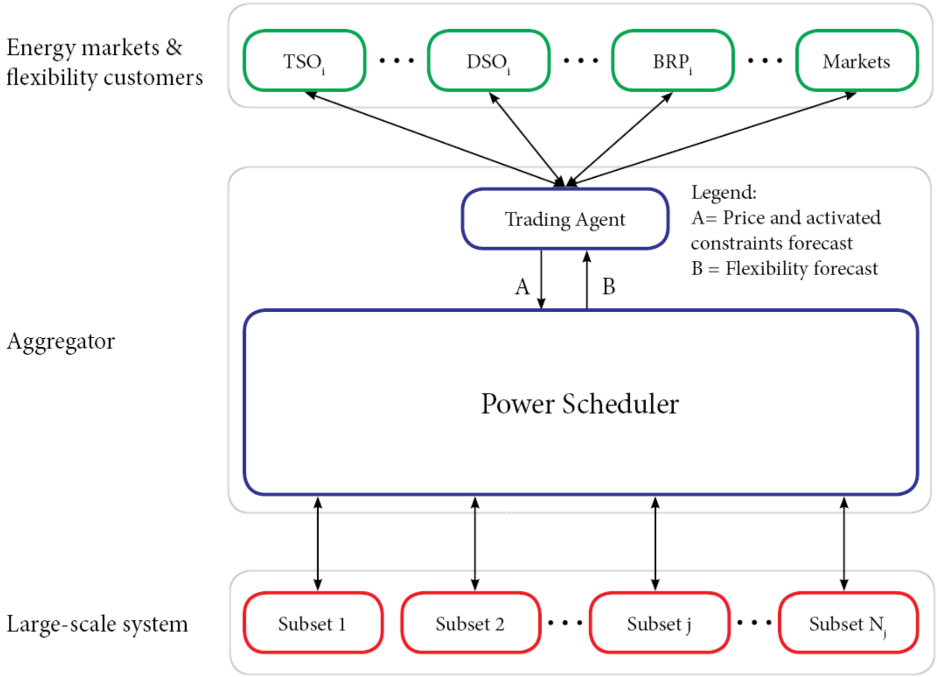


Figure 1-5: Schematic representation of the aggregator setting with respect to potential energy flexibility users and the EV fleet, which are divided over the subsets.

aggregator can also provide forecasts of its power interaction to DSOs and other potential customers over time.

The aggregator can negotiate with the potential customers on how to utilize the available energy flexibility, to optimize its profitability. These complex procedures are out of the scope of this thesis and as such a split is performed within the aggregator. As a result, two coupled agents are formed: the *trading agent* and the *power scheduler*. See Figure 1-5 for a schematic representation of the aggregator.

Trading Agent

The trading agent performs the communication and negotiation about the available energy flexibility with all potential users. Furthermore, internally, it provides the power scheduler with a forecast on the constraints arising from the network and with a forecast on the electricity buy and sell price, based on the negotiations it has had. The network constraints depict the allowable power consumption over time, which can be given per congestion point or any combination of congestion points (e.g. the whole EV fleet). The forecast on the electricity buy and sell price will be used by the power scheduler to minimize the cost for the remaining flexibility.

1-3 Problem Formulation of the Power Scheduler

In this thesis, the perspective of an EV aggregator is taken, which is responsible for the power scheduling of a fleet of EVs connected to a particular part of a distribution network. The control task is to determine the power schedule for each individual EV, such that both the EV and network constraints are respected and the overall cost is minimized. In the following, all constraints related to individual EVs are referred to as *local constraints* and those enforced by the trading agent, related to the network, are referred to as *global constraints*. Typically, the number of global constraints is significantly smaller than the number of EVs under control. Furthermore, it is assumed the global constraints allow feasible power schedules to be found. The control task will be applied in a discrete time setting, with a step size of $\tau = 0.25$ hours, i.e. 15 minutes, which is a common duration of Program Time Units (PTUs) in energy markets.

The framework of Model Predictive Control (MPC) is used to solve the power scheduling problem of the aggregator using receding horizon control with a finite prediction horizon. In MPC, at each sample time t the problem is solved such that an input sequence is obtained over a prediction horizon of N steps. When an optimal input sequence is found, the input for the first step in the horizon is implemented and the process is repeated at the next sample time, using updated information. The PTUs within the prediction horizon are indexed by k, which takes the values $k = \{0, 1, \ldots, N-1\}$.

Consider an environment that contains a constant number of N_j congestion points (indexed by j) and a fleet of EVs (indexed by i). All EVs have an arrival and departure time and a physical distribution network is considered in which each EV can connect to one congestion point. Each EV is, therefore, part of only one subset within the power scheduler. The SoC of EV i, within subset j, is denoted with $E_{j,i}$, with both j and i as subscripts.

Local Constraints

It is assumed that the EVs draw power at fixed rates, which is usually the case in practice [5][1]. In this thesis, single charging and discharging rates will be used, but the framework allows for multiple fixed rates as well. Furthermore, charging and discharging can be interrupted and resumed, but in order to avoid excessive switching, once a control action is applied it must continue for at least the duration of a single PTU.

As direct control is used, the aggregator has the authority to determine the charing behavior for each individual EV. It is assumed as long as each EV i within subset j is connected to the power scheduler, it transmits all its relevant variables to the power scheduler, including the current SoC, $E^{0}_{j,i}$, desired final SoC, $E^{\text{ref}}_{j,i}$, minimum and maximum attainable SoC, $E^{\text{min}}_{j,i}$ and $E^{\text{max}}_{j,i}$ respectively, charging and discharging power, $P^{\text{c}}_{j,i}$ and $P^{\text{d}}_{j,i}$ respectively, charging and discharging efficiency, $\eta^{\text{c}}_{j,i}$ and $\eta^{\text{d}}_{j,i}$ respectively, and the time when the EV is scheduled to be disconnected, $T^{\text{dep}}_{j,i}$. It is assumed that the responsible controller has a forecast of all EVs

[1] This is particularly true in case of connections with low power ratings. More generally, connections compatible with the IEC 61851 standard could operate in a semi-continuous fashion, i.e. with a minimum current output when charging and can then be modulated within a certain band. This requirement results in models containing discrete variables and hence fits the proposed framework. However, to keep the description concise this aspect is not considered.

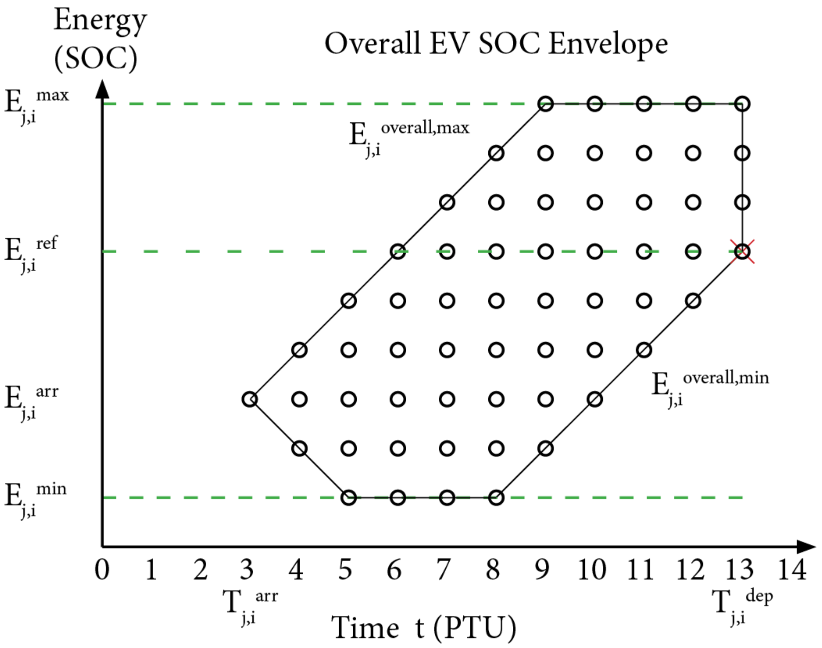


Figure 1-6: Simplified overall EV SoC envelope

Simplified with τ=1, ideal efficiencies ($\eta^{\mathrm{c}}_{j,i}$=$\eta^{\mathrm{d}}_{j,i}$=1) and equal power interacting values, i.e. $P^{\mathrm{c}}_{j,i}$=$P^{\mathrm{d}}_{j,i}$. For this example, the EV arrives at $T^{\mathrm{arr}}_{j,i}$=3 and is set to depart at $T^{\mathrm{dep}}_{j,i}$=13.

which will be connected to it and their associated variables, including their expected arrival time $T^{\mathrm{arr}}_{j,i}$ and SoC upon arrival $E^{\mathrm{arr}}_{j,i}$, i.e. a deterministic setting is used.

In order to interpret these variables intuitively, the SoC envelope is introduced, which represents the allowed energy storage over time for the specified EV variables as stated before. A simplified example of an overall SoC envelope is shown in Figure 1-6. For clarity, it is simplified with ideal efficiencies and equal power interacting values for charging and discharging, to limit the amount of possible SoC values. This simplification is only used for visualizing the SoC envelope and is not taken advantage of in the control problem. The black solid lines represent the overall SoC envelope, which can be found when the extremes are applied. The upper limit, $E^{\mathrm{overall,max}}_{j,i}$, is found when full charging is applied from the very first moment, until $E^{\mathrm{max}}_{j,i}$ is reached and the SoC is set to remain at that level. The lower limit, $E^{\mathrm{overall,min}}_{j,i}$, is found when the EV is set to full discharging until $E^{\mathrm{min}}_{j,i}$ is found, where the SoC remains until it is forced to start charging such that is able to meet $E^{\mathrm{ref}}_{j,i}$ by $T^{\mathrm{dep}}_{j,i}$, which is represented by the red cross. The green dotted lines represent the effect of the SoC constraints and the black circles represent the finite number of possible SoC values, which is a direct result of the fixed charging and discharging rates. The possible values of $E_{j,i}(k)$ within a SoC envelope are also jointly referred to as the *energy flexibility of an EV*.

1-4 Research Objectives

The power scheduling optimization problem, the aggregator is set to solve, is formulated as an Integer Linear Programming problem (ILP). For an ILP it is possible to obtain a global optimum, yet since these belong to the class of NP-Complete problems, the computational cost will grow very quickly when the problem size increases.

The main goal of this thesis is to develop and compare algorithms which enable the ILP to be solved for large scale systems, by reducing the computation time, yet at the cost of a potentially suboptimal solution. For the developed algorithms, the relation between the solution quality with the related computational cost will be studied. The developed algorithms will be compared to a centralized MPC algorithm, which will be used as a benchmark.

An architecture with some kind of coordinator or top node is most likely to arise, due to the communication with the trading agent. Furthermore, as a result of the privacy sensitive information the power scheduler deals with and to limit communicational demands, algorithms with limited information sharing are beneficial. Although the use of relaxation techniques and constraint softening have been proven useful, this thesis focuses on the use of feasible iterations. I.e. any iteration which provides a solution to the problem, respects all constraints and can be implemented in a closed loop setup. As a result, the algorithm can be stopped at any time after the first solution to the problem is obtained, yet performance is most likely to be further improved when more iterations are allowed. Using feasible iterations is also known as *Anytime MPC*.

Since the goal is to study the control architecture of the power scheduler rather than the practical implementation of such an architecture, the following initial assumptions are made:

1. All system states, $E_{j,i}$, are fully observable. This means that all EVs are equipped with the necessary sensors. Disturbances are not taken into account.
2. All responsible controllers have a perfect forecast of all EVs which will be connected to it and their associated variables, including their expected arrival time $T_{j,i}^{\mathrm{arr}}$ and SoC upon arrival $E_{j,i}^{\mathrm{arr}}$, i.e. a deterministic setting is used.
3. All controllers are able to communicate with the coordinator or top node without delays and loss of information.

1-5 Related Literature and Research Contributions

In literature, distributed architectures to solve the power scheduling problem of an aggregator, is receiving more attention over time. In [5] and [6], algorithms have been proposed for the overnight power scheduling of an EV fleet, i.e. one large scheduling problem is solved instead of an MPC implementation. Furthermore, both publications rely on the mathematical basis of the dual problem formulation, in which the coupling global constraints are dualized in the objective function, resulting in infeasible iterations. In [7, 8], a very similar environment has been used and an algorithm has been proposed which solves the optimization problem using distributed MPC with resource allocation. Yet, the algorithm cannot be directly applied to linear cost functions, which is the most natural and simple way to calculate the charging costs and does not incorporate Vehicle to Grid (V2G). None of the algorithms found in literature incorporate the spatial information of the EVs, to be able to better valorize the energy flexibility.

As thesis contribution, congestion points are introduced in the aggregator power scheduling problem to provide congestion management services. More realistic EV properties have been used, incorporating arrival and departures, unideal efficiencies and the derivations are all provided for a V2G setting, which has not been seen often in literature. Furthermore, to be able to better model the aggregated behavior of a fleet of EVs with fixed power rates, the Virtual Battery (VB) constraints on both the power envelope and SoC envelope have been improved from what was found in the literature [9, 10, 11]. Finally, to the best of the authors knowledge, no MPC architecture comparison is available in the literature for fixed rate power scheduling EV aggregators.

1-6 Outline

The remainder of this thesis is organized as follows:

- **Chapter 2** presents the binary input model for single EVs and the detailed formulation of the optimization problem of the system within the framework of MPC as an ILP. The Centralized Model Predictive Control (CMPC) algorithm is presented and discussed.
- Next, **Chapter 3** presents the developed Distributed Model Predictive Control with Resource Allocation (DMPC-RA) algorithm, which is a multi-agent MPC formulation for integer input systems with limited information sharing between nodes with feasible and good iterations, approaching the global optimal system behavior over time. The detailed formulations are given and the nodal MPC problems are formulated as ILPs.
- **Chapter 4** introduces the VB model, to represent the aggregated behavior of a fleet of EVs using very limited information. Furthermore it explains how the optimization problem for the system can be decomposed using VBs in a Hierarchical Decentralized Model Predictive Control (HDe-MPC) architecture with a reference tracking scheme. The architecture consists of a single the top MPC and multiple lower MPCs, of which the problems are formulated as Mixed Integer Linear Programming problems (MILPs).
- In **Chapter 5**, the developed CMPC, DMPC-RA and HDe-MPC algorithms are compared using numerical experiments for both open loop as closed loop behavior and discusses their merits and limitations. Finally, in Chapter 6 the main conclusions of this thesis are presented and recommended future work is discussed.

Chapter 2

Centralized Model Predictive Control

In this chapter, the notation is introduced for the optimization problem of the system. A Centralized Model Predictive Control (CMPC) architecture is assumed, such that all information about the system is available within the large, single CMPC. The CMPC has a limited prediction horizon over which information is available, about both the network constraints and the Electric Vehicle (EV) fleet behavior. The interaction between the trading agent, the customers, EV fleet and the power scheduler were presented in Chapter 1. The communication structure of the CMPC architecture with the EV fleet is shown in Figure 2-1.

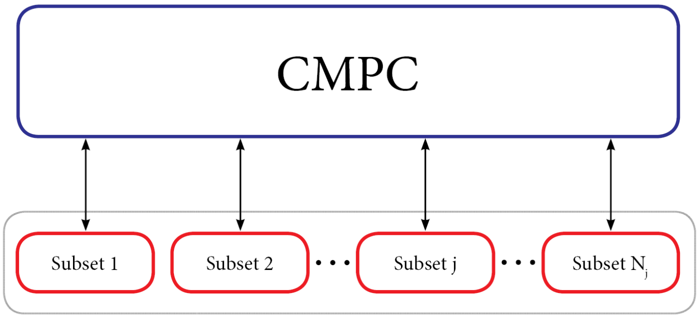


Figure 2-1: Communication structure of the CMPC algorithm

2-1 Formulation of the Optimization Problem of the System

In order to be able to implement the specified control task of the power scheduler, it is formulated as a MPC optimization problem. The constraints that are related to individual EVs are referred to as *local constraints* and those that are enforced by the trading agent, related to the network, are referred to as *global constraints*. Typically, the number of global constraints is significantly smaller than the number of EVs under control.

For the sake of brevity and clarity, the design of the CMPC algorithm is presented for a single sample time. The time step $k{=}0,1,\ldots,N{-}1$ will be used as the step within the prediction horizon, in which N is the length of the prediction horizon in Program Time Units (PTUs). Let the time steps within the horizon be in the set $\mathcal{N}{=}\{0,1,\ldots,N{-}1\}$.

Arrivals and Departures

An environment is used in which EVs are arriving and departing. If EV i is expected to be connected to the aggregator within the current prediction horizon $k{\in}\mathcal{N}$, the first time step is denoted with $k_{j,i}^{\mathrm{arr}}$ and the last time step is denoted with $k_{j,i}^{\mathrm{dep}}$. Both $k_{j,i}^{\mathrm{arr}}$ and $k_{j,i}^{\mathrm{dep}}$ are assumed known to the MPC. If EV i was connected before the current sampling time, then $k_{j,i}^{\mathrm{arr}}{=}0$. Likewise, if the EV is connected until the end of the prediction horizon or even beyond, then $k_{j,i}^{\mathrm{dep}}{=}N{-}1$.

With the first and last step defined for EV i, the set $\mathcal{N}_{j,i}$ consists of that part of the prediction horizon during which the EV is connected. It is defined as,

$$\mathcal{N}_{j,i} = \{k_{j,i}^{\mathrm{arr}}, k_{j,i}^{\mathrm{arr}}{+}1, \ldots, k_{j,i}^{\mathrm{dep}}\}. \tag{2-1}$$

Furthermore, the set $\mathcal{I}_j(k)$ consists of all EVs connected to congestion point j at time step k and is defined as,

$$\mathcal{I}_j(k) = \{i \mid \text{connected to congestion point } j, k{\geq}k_{j,i}^{\mathrm{arr}} \text{ and } k{\leq}k_{j,i}^{\mathrm{dep}},\ \ \forall i\}. \tag{2-2}$$

The initial State of Charge (SoC) within the prediction horizon is assumed known, given as $E_{j,i}^{0}$ and the minimum SoC at the end of the prediction horizon for EV i will be denoted as $E_{j,i}^{\mathrm{MPC,ref}}$.

The variables $k_{j,i}^{\mathrm{arr}}, k_{j,i}^{\mathrm{dep}}$ and $E_{j,i}^{0}, E_{j,i}^{\mathrm{MPC,ref}}$, together with the set $\mathcal{N}_{j,i}$ will be used to denote in which part of the overall SoC envelope an EV is currently, which was introduced in Figure 1-6. The set $\mathcal{I}_j$ will be used for aggregated definitions.

Decision Variables

The control signal the aggregator assigns to EV i, connected to congestion point j, is represented by the binary variables $u_{j,i}^{\mathrm{c}}(k), u_{j,i}^{\mathrm{d}}(k){\in}\{0,1\}$, for time step k.

$$\begin{aligned} u_{j,i}^{\mathrm{c}}(k) &= \begin{cases} 0, & \text{EV is not charging} \\ 1, & \text{EV is charging} \end{cases} \\ u_{j,i}^{\mathrm{d}}(k) &= \begin{cases} 0, & \text{EV is not discharging} \\ 1, & \text{EV is discharging} \end{cases} \end{aligned} \tag{2-3}$$

with which the input vector is defined as,

$$u_{j,i}(k) = \begin{bmatrix} u_{j,i}^{\mathrm{c}}(k) \\ u_{j,i}^{\mathrm{d}}(k) \end{bmatrix} \qquad \forall k{\in}\mathcal{N}_{j,i} \tag{2-4}$$

Only one mode can be activated at the same time, hence a constraint needs to be added as,

$$\begin{bmatrix} 1 & 1 \end{bmatrix} u_{j,i}(k) \leq 1, \quad \forall k \in \mathcal{N}_{j,i}. \tag{2-5}$$

Clearly, the three arising modes are charging $u^{\mathrm{c}}_{j,i}(k){=}1, u^{\mathrm{d}}_{j,i}(k){=}0$, discharging $u^{\mathrm{c}}_{j,i}(k){=}0, u^{\mathrm{d}}_{j,i}(k){=}1$ and idle $u^{\mathrm{c}}_{j,i}(k){=}u^{\mathrm{d}}_{j,i}(k){=}0$. This can be rewritten in set notation, such that $u_{j,i}(k)$ is constrained to lie in $\mathcal{U}_{j,i}(k)$, which is defined as,

$$\mathcal{U}_{j,i}(k){=}\left\{ \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \end{bmatrix} \right\}. \tag{2-6}$$

Dynamics of the System

A binary input model is used to represent the power interaction of the EVs, as in [6] and used in simplified form in [7, 5]. Let $E_{j,i}(k)$ denote the SoC of the i-th EV, connected to congestion point j. It is affected by charging and discharging. In the model, $P^{\mathrm{c}}_{j,i}$ and $P^{\mathrm{d}}_{j,i}$ denote the constant charging and discharging power respectively, and $\eta^{\mathrm{c}}_{j,i}$ and $\eta^{\mathrm{d}}_{j,i}$ represent the charging and discharging efficiency, respectively. The SoCs for the next time step can then be calculated as,

$$E_{j,i}(k{+}1) = E_{j,i}(k) + \begin{bmatrix} \tau P^{\mathrm{c}}_{j,i}\eta^{\mathrm{c}}_{j,i} & -\tau P^{\mathrm{d}}_{j,i}\frac{1}{\eta^{\mathrm{d}}_{j,i}} \end{bmatrix} \begin{bmatrix} u^{\mathrm{c}}_{j,i}(k) \\ u^{\mathrm{d}}_{j,i}(k) \end{bmatrix} \qquad \forall k \in \mathcal{N}_{j,i}, \tag{2-7}$$

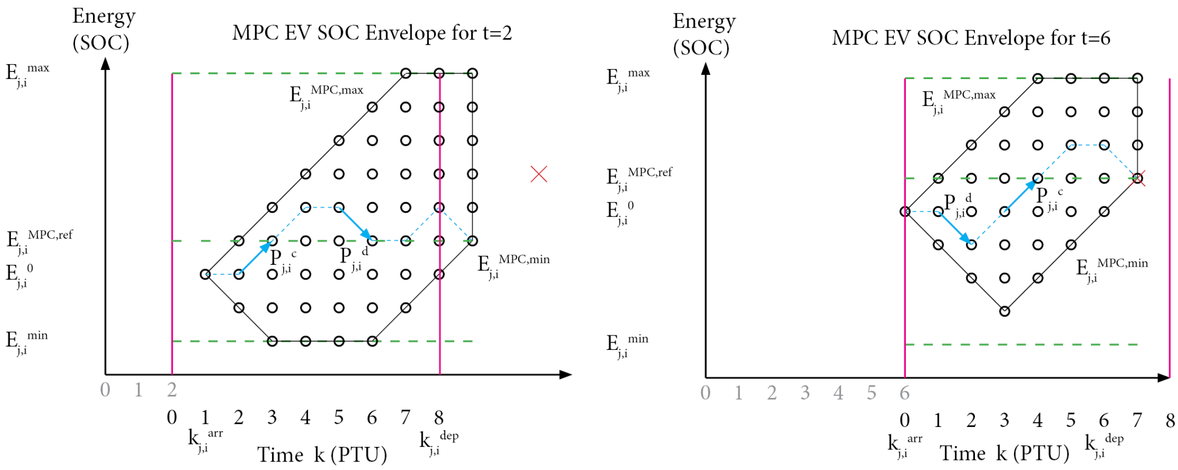
or equivalently,

$$E_{j,i}(k{+}1) = E_{j,i}(k) + B_{j,i}u_{j,i}(k) \qquad \forall k \in \mathcal{N}_{j,i}. \tag{2-8}$$

State of Charge Constraints

The overall SoC constraints are set by the EV user. At each sample time, the energy flexibility options within the prediction horizon are updated, based on the overall constraints and the current SoC, $E^{0}_{j,i}$. The minimum SoC at the end of the horizon, $E^{\mathrm{MPC,ref}}_{j,i}$, has already been introduced. The variables $E^{\min}_{j,i}$ and $E^{\max}_{j,i}$ define the allowable minimum and maximum SoC respectively. When $k^{\mathrm{arr}}_{j,i}{>}0$, the initial value is set to $E^{0}_{j,i}{=}E^{\mathrm{arr}}_{j,i}$, which is the SoC when the EV arrives and connects to the aggregator.

For simplified settings, i.e. $\tau{=}1$ hour, the charging and discharging efficiencies are ideal ($\eta^{\mathrm{c}}_{j,i}{=}\eta^{\mathrm{d}}_{j,i}{=}1$) and $P^{\mathrm{c}}_{j,i}{=}P^{\mathrm{d}}_{j,i}$, then the energy flexibility options of an individual EV within a MPC prediction horizon can be represented as in Figure 2-2 by forming a MPC SoC envelope [9, 10, 11]. Since the SoC values are predicted for the next time step, they are shown for one step out of the prediction horizon. In this example the prediction horizon is $N{=}9$ PTUs and the same EV is used of which the overall SoC envelope was presented in Figure 1-6. The magenta vertical lines depict the current prediction horizon and the black solid lines represent the MPC SoC envelope predictions, starting at the initial SoC. The red cross is not know to the CMPC, but represents the overall $E^{\mathrm{ref}}_{j,i}$ at $T^{\mathrm{dep}}_{j,i}$. The circles represent all finite number of possible SoC values and the blue dotted line represents a feasible power schedule. The derivation of this MPC SoC envelope will be introduced subsequently.



(a) Simplified MPC EV SoC envelope for t=2.

(b) Simplified MPC EV SoC envelope for t=6.

Figure 2-2: Simplified MPC EV SoC envelopes

Simplified with ideal efficiencies ($\eta_{j,i}^{c}$=$\eta_{j,i}^{d}$=1) and equal power values, i.e. $P_{j,i}^{c}$=$P_{j,i}^{d}$. A prediction horizon of N=9 PTUs is used.

If more realistic settings are used for visualizations, yet still using τ=1 hour, the MPC SoC envelope shown in Figure 2-2a would become much more complex, a limited example is shown in Figure 2-3. The latter envelope can be found when a realistic discharging efficiency is used, i.e. $\eta_{j,i}^{d}<1$, or different values for charging and discharging are used, i.e. $P_{j,i}^{c} \neq P_{j,i}^{d}$. In contrast with the ideal settings, a subsequent charging and discharging PTU no longer ends up at the same SoC at which was started. With ideal settings and equal charging and discharging power values, $P_{j,i}^{c}\eta_{j,i}^{c} - P_{j,i}^{d}\frac{1}{\eta_{j,i}^{d}}=0$, yet this does not hold for more realistic settings and in fact when using $\eta_{j,i}^{d}<1$ the result would be, $P_{j,i}^{c}\eta_{j,i}^{c} - P_{j,i}^{d}\frac{1}{\eta_{j,i}^{d}}<0$. As such, new viable SoC values arise over time with respect to the ideal settings.

For clarity, only the first three steps are shown in the realistic MPC SoC envelope example in Figure 2-3. One can see the new viable SoC values arise over time, just below previous viable SoC settings, since typically $\eta_{j,i}^{d}$ still has values between 0.8−0.95. The orange dotted line is the tight set for the MPC horizon. The blue shaded area is unreachable in this example for realistic settings. For the argument of clarity, the simplified MPC SoC envelope will be used for further visualizations.

The dynamics of the system, (2-8), incorporate this difference in behavior between ideal and more realistic settings, therefore the MPC SoC envelope can be defined using these dynamics. Based on the current values of $E_{j,i}^{0}$, $E_{j,i}^{\text{MPC,ref}}$, the MPC SoC envelope can be formed by calculating the respective diagonal slopes of the envelope. For the top of the envelope, starting at $E_{j,i}^{0}$ the energy flexibility grows towards $E_{j,i}^{\max}$ with $E_{j,i}^{\text{MPC,min,slope}}(k+1)$, until it reaches $E_{j,i}^{\max}$. On the bottom of the envelope, $E_{j,i}^{\text{MPC,min,slope}}(k+1)$ starts at $E_{j,i}^{0}$ and decreases until $E_{j,i}^{\min}$ is reached. On the bottom right, $E_{j,i}^{\text{MPC,ref,slope}}(k+1)$ denotes the slope with which the SoC needs to rise, such that $E_{j,i}^{\text{MPC,ref}}$ at $k_{j,i}^{\text{dep}}+1$ is met. Since SoC predictions are constrained,

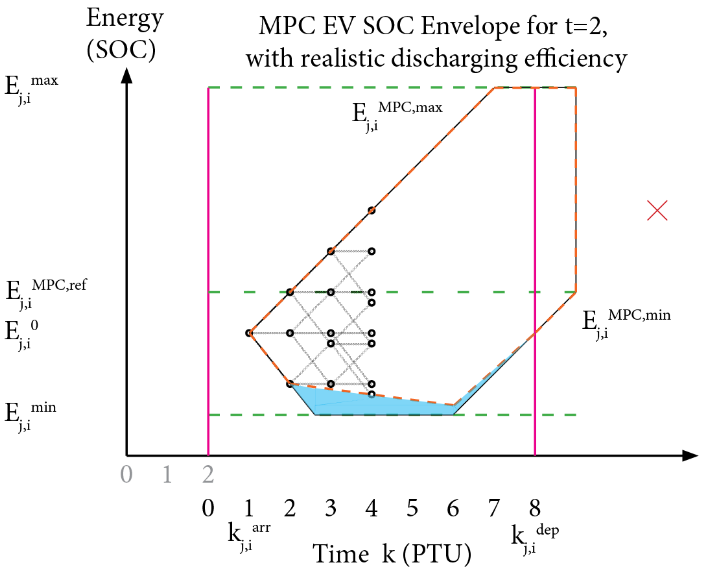


Figure 2-3: Realistic MPC EV SoC envelope, with new viable SoC levels arising over time
SoC envelope with possible SoC values for the first three steps. The envelope can be found when a realistic discharging efficiency is used, i.e. $\eta^{\text{d}}_{j,i}{<}1$, or different values for charging and discharging are used, i.e. $P^{\text{c}}_{j,i}{\neq}P^{\text{d}}_{j,i}$. A prediction horizon of $N{=}9$ PTUs is used.

$k{+}1$ is used and the three slopes are defined as,

$$\begin{aligned} E^{\text{MPC,min,slope}}_{j,i}(k{+}1) &= E^{0}_{j,i}{-}\tau(k{+}1{-}k^{\text{arr}}_{j,i})P^{\text{d}}_{j,i}\frac{1}{\eta^{\text{d}}_{j,i}} && \forall k{\in}\mathcal{N}_{j,i} \\ E^{\text{MPC,max,slope}}_{j,i}(k{+}1) &= E^{0}_{j,i}{+}\tau(k{+}1{-}k^{\text{arr}}_{j,i})P^{\text{c}}_{j,i}\eta^{\text{c}}_{j,i} && \forall k{\in}\mathcal{N}_{j,i} \\ E^{\text{MPC,ref,slope}}_{j,i}(k{+}1) &= E^{\text{MPC,ref}}_{j,i}{+}\tau(k{+}1{-}k^{\text{dep}}_{j,i})P^{\text{c}}_{j,i}\eta^{\text{c}}_{j,i} && \forall k{\in}\mathcal{N}_{j,i}. \end{aligned} \tag{2-9}$$

Using these slopes, the bounds on the energy flexibility for an EV within the horizon are defined by,

$$\begin{aligned} E^{\text{MPC,min}}_{j,i}(k{+}1) &= \max\left(E^{\text{MPC,min,slope}}_{j,i}(k{+}1), E^{\text{min}}_{j,i}, E^{\text{MPC,ref,slope}}_{j,i}(k{+}1)\right) && \forall k{\in}\mathcal{N}_{j,i} \\ E^{\text{MPC,max}}_{j,i}(k{+}1) &= \min\left(E^{\text{MPC,max,slope}}_{j,i}(k{+}1), E^{\text{max}}_{j,i}\right) && \forall k{\in}\mathcal{N}_{j,i} \end{aligned} \tag{2-10}$$

which constrain the value of $E_{j,i}(k{+}1)$ as,

$$E^{\text{MPC,min}}_{j,i}(k{+}1) \leq E_{j,i}(k{+}1) \leq E^{\text{MPC,max}}_{j,i}(k{+}1) \quad \forall k{\in}\mathcal{N}_{j,i} \tag{2-11}$$

The set containing the EV energy flexibility options within the MPC horizon for step $k{+}1$ is then defined as,

$$\begin{aligned} \mathcal{E}^{\text{MPC}}_{j,i}(k{+}1){=}\{E_{j,i}(k{+}1) \mid {} & E_{j,i}(k{+}1){=}E_{j,i}(k){+}B_{j,i}u_{j,i}(k), \\ & E^{\text{MPC,min}}_{j,i}(k{+}1){\leq}E_{j,i}(k{+}1){\leq}E^{\text{MPC,max}}_{j,i}(k{+}1) \text{ and} \\ & E_{j,i}(k^{\text{arr}}_{j,i}){=}E^{0}_{j,i} \;\; \forall u_{j,i}(k){\in}\mathcal{U}_{j,i}(k),\; \forall k{=}\mathcal{N}_{j,i}\}. \end{aligned} \tag{2-12}$$

for which holds that the MPC SoC envelope will always be a mathematical subset of the overall SoC envelope of an EV.

Global Constraints

The global constraints affect the available resources for the subsets and represent the congestion management constraints. Each subset represents a low-end congestion point (e.g. a low voltage station or distribution line) and combinations of subsets can represent a congestion point higher up in the physical architecture (e.g. a medium voltage station), see Figure 1-4. The matrix Λ_k^{CM} consists of the prediction of activated constraints for step k, in which both single subsets as combinations of subsets can be constrained and the vector P_k^{CM} consists of the respective power values by which the combination is constrained. The vector U_k is the combination of all binary input vector for step k and the block diagonal matrix Φ_k^{CM} consists of the power values related to U_k, with a row per subset. The detailed definitions of these variables are given in Appendix A and omitted here for brevity. The global constraints for a single time step k can be jointly represented as,

$$\Lambda_k^{\text{CM}}\Phi_k^{\text{CM}}U_k \leq P_k^{\text{CM}}. \tag{2-13}$$

For example, in an environment with two subsets, with respectively one and two EVs, of which both the sum of the two subsets is constrained by $P_k^{\text{CM}}(1)$ and the first subset is also constrained by $P_k^{\text{CM}}(2)$, then for time step k, this results in,

$$\underbrace{\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}}_{\Lambda_k^{\text{CM}}} \underbrace{\begin{bmatrix} P_{1,1}^{\text{c}} & -P_{1,1}^{\text{d}} & 0 & 0 & 0 & 0 \\ 0 & 0 & P_{2,2}^{\text{c}} & -P_{2,2}^{\text{d}} & P_{2,3}^{\text{c}} & -P_{2,3}^{\text{d}} \end{bmatrix}}_{\Phi_k^{\text{CM}}} \underbrace{\begin{bmatrix} u_{1,1}^{\text{c}} \\ u_{1,1}^{\text{d}} \\ u_{2,2}^{\text{c}} \\ u_{2,2}^{\text{d}} \\ u_{2,3}^{\text{c}} \\ u_{2,3}^{\text{d}} \end{bmatrix}}_{U_k} \leq \underbrace{\begin{bmatrix} P_k^{\text{CM}}(1) \\ P_k^{\text{CM}}(2) \end{bmatrix}}_{P_k^{\text{CM}}} \tag{2-14}$$

Cost Function

The power scheduler receives the internal buy and sell price forecasts, respectively $p^{\text{b}}(k)$ and $p^{\text{s}}(k)$, for all $k \in \mathcal{N}$ from the trading agent. Each EV wants to minimize its cost of charging and the cost function of the aggregator is simply the summation of all the individual charging costs from all EVs. The charging cost of a single EV is calculated as the summation of its costs of charging minus the revenues from discharging. The cost $J_{j,i}$ for a single EV connected to congestion point j is then

$$J_{j,i} = \sum_{k \in \mathcal{N}_{j,i}} f_{j,i}^T(k) u_{j,i}(k), \tag{2-15}$$

in which the EV cost vector $f_{j,i}(k)$ is defined as,

$$f_{j,i}^T(k) = \left[\tau P_{j,i}^{\text{c}} p^{\text{b}}(k) \quad -\tau P_{j,i}^{\text{d}} p^{\text{s}}(k)\right] \tag{2-16}$$

where τ is the duration of a PTU and $P_{j,i}^{\text{c}}$ and $P_{j,i}^{\text{d}}$ are the charging and discharging power respectively. As a result, the cost function of the power scheduler can be written as,

$$\text{minimize} \sum_{k \in \mathcal{N}} \sum_{j=1}^{N_j} \sum_{i \in \mathcal{I}_j(k)} f_{j,i}^T(k) u_{j,i}(k). \tag{2-17}$$

Centralized MPC Optimization Problem

To simplify the notation, Appendix A presents the detailed derivation in which all variables are joined over the prediction horizon, over all the subsets and all EVs. The input vector containing all binary variables over the whole prediction horizon becomes $\mathbf{U}$. All local constraints (2-5), (2-8) and (2-11) are joined and rewritten using substitution, such that they can be written as constraints on the input vector, which is presented in set notation as $\mathbf{U}{\in}\mathcal{U}$. All global constraints (2-13) are joined and can be written as $\mathbf{\Lambda}^{\text{CM}}\mathbf{\Phi}^{\text{CM}}\mathbf{U}{\leq}\mathbf{P}^{\text{CM}}$ and all cost vectors are combined such that they can be written as $\mathbf{f}$. The global constraints will be left out of the set $\mathcal{U}$ to keep a clear distinction between the local and global constraints. Using these more compact definitions, the CMPC optimization problem, solved at each sample time and with a limited horizon of N PTUs, can be formulated as,

$$\begin{aligned} \underset{\mathbf{U}}{\text{minimize}} \quad & \mathbf{f}^T\mathbf{U} \\ \text{s.t.} \quad & \mathbf{U}{\in}\mathcal{U} \\ & \mathbf{\Lambda}^{\text{CM}}\mathbf{\Phi}^{\text{CM}}\mathbf{U}{\leq}\mathbf{P}^{\text{CM}} \end{aligned} \tag{2-18}$$

The cost related to the solution of the optimization problem for the system is also referred to as the *performance*. Optimal performance is achieved when the cost cannot be further minimized for the current sample time.

2-2 Conclusions

In this chapter, the MPC optimization problem of the system is defined in detail as large scale Integer Linear Programming problem (ILP), using binary inputs to represent the fixed charging and discharging rates in (2-18). The local constraints are those related to the individual EVs and put in set notation for brevity. The global constraints are those provided by the trading agent, related to the network. In (2-14), an example is given on how multiple congestion management constraints can be enforced on single subsets or combinations thereof. The energy flexibility MPC SoC envelope has been introduced for a single EV, for both a simplified and more realistic setting in Figure 2-2 and Figure 2-3 respectively.

Although the derived problem in (2-18) can be easy to solve conceptually, it will encounter an enormous computational cost for large scale systems, i.e. for a large fleet of EVs. This motivates the aforementioned goals, as discussed in Chapter 1, to develop the two coordinated distributed approaches, which will be presented in the following chapters.

Chapter 3

Distributed Model Predictive Control using Resource Allocation

In this chapter, a Distributed Model Predictive Control with Resource Allocation (DMPC-RA) algorithm is developed, to solve the power scheduling problem defined in Section 2-1. The DMPC-RA architecture is a multi-agent MPC formulation for integer input systems with limited information sharing between nodes with feasible iterations, approaching the global optimal system performance over time. It comprises of a coordinator role and multiple MPCs, each responsible for the control of one subset.

The communication structure of the developed DMPC-RA algorithm is shown in Figure 3-1. The coordinator optimizes the resource allocation, based on limited information from the subset MPCs. The resource allocation represents the amount of resources made available to each subset, represented by the auxiliary variable γ. Subsequently, the subset MPCs optimize the power schedules for their Electric Vehicles (EVs), using the allocated resources. Since only limited information is available to both the coordinator and subset MPCs, the resource allocation coordination algorithm is solved over a number of iterations such that in each following iteration, resources can be shifted from one subset to another to improve global system performance.

The aim is to achieve global optimal system performance. Yet, due to the discrete jumps in the behavior of the system, it is in general very hard to reach global optimality in a distributed setting, especially when active network constraints are applied and only feasible iterations are used. The goal of this chapter is to derive the DMPC-RA algorithm such that global optimal system performance is approached for an increasing number of iterations. The coordinator only has information on a high-over level, as such it has no detailed information about any of the EVs. At the start of each sampling time, the trading agent provides the coordinator with a forecast of activated network constraints, i.e. the coupling between subsets $\mathbf{\Lambda}^{\text{CM}}$ and the vector of corresponding maximum values $\mathbf{P}^{\text{CM}}$. The coordinator is set with the responsibility to ensure these global constraints are met.

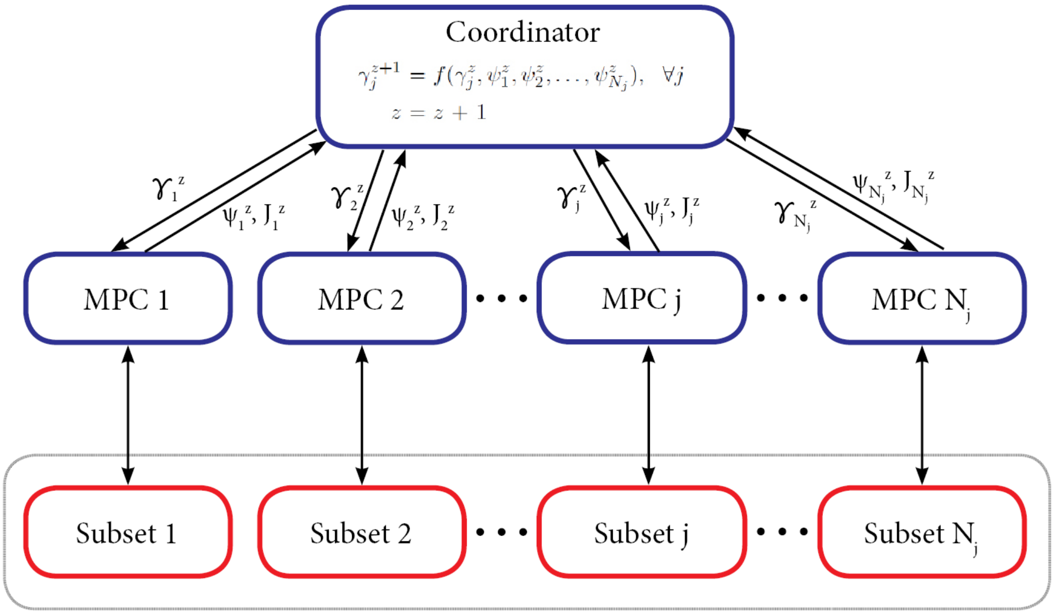


Figure 3-1: Communication structure of the developed DMPC-RA algorithm

The subset MPCs only have local information and have availability over the price signals p^{b} and p^{s}, such that they can optimize the power schedules of the EVs within their subset and minimize their operational cost.

Given an initial resource allocation, represented by $\boldsymbol{\gamma}_j$ for each subset j, the subset MPCs solve their optimization problem. Subsequently, they let the coordinator know to what extent they would benefit from extra resources, represented by $\boldsymbol{\psi}_j$ and their current cost function value, represented by J_j. The coordinator then evaluates if the global system performance has improved and calculates a new resource allocation.

The rest of this chapter is organized as follows. First, an overview of the decomposition method is discussed in Section 3-1. In Section 3-2, the rationale and challenges in resource allocation for binary input systems are discussed. Furthermore, the section presents the derivation of the communicated variables $\boldsymbol{\gamma}_j, \boldsymbol{\psi}_j$ and the necessary iterative procedures to approach global optimal performance. In Section 3-3 the combined algorithm for coordinator and subset MPCs is presented. To conclude, Section 3-4 presents the conclusions of the chapter.

3-1 Decomposing the Optimization Problem of the System

To use a distributed MPC architecture, the optimization problem defined (2-18) has to be decomposed. Since the dynamics, cost functions and local constraints are defined per EV, only the global constraints have to be decoupled. The decoupling of the global constraints is obtained by introducing the auxiliary scalar variable γ_j, representing the maximum available resources for subset j. Given γ_j, each subset MPC j will optimize its power schedule, while not using more than the allocated resources.

The iteration procedure allows for the shifting of resources from one subset to another, such that the global cost is decreased. The coordinator is responsible for calculating the optimal resource allocation and assesses the global performance. Let z be the iteration counter.

As defined previously, Λ_k^{CM} and P_k^{CM} represent the coupling between subsets and their constraining values for k respectively. Let $\gamma_j(k)$ be defined such that,

$$\begin{aligned} \Lambda_k^{\mathrm{CM}} \begin{bmatrix} \gamma_{j=1}(k) \\ \gamma_{j=2}(k) \\ \vdots \\ \gamma_{j=N_j}(k) \end{bmatrix} &= P_k^{\mathrm{CM}} \qquad \forall k \in \mathcal{N} \\ \gamma_j(k) &\geq 0 \qquad \forall j, \forall k \in \mathcal{N}. \end{aligned} \tag{3-1}$$

which will be jointly referred to as the *decoupled global constraints* in the following. As a result of (3-1), the available resources P_k^{CM} will always be divided over the subset MPCs, which can locally decide whether to make use of these resources or not. The subset MPCs only have information on their own $\gamma_j(k)$ over the horizon, represented as $\boldsymbol{\gamma}_j$. While the coordinator has knowledge over the entire resource allocation, denoted by $\boldsymbol{\gamma}$. The resource allocation vectors are defined as,

$$\boldsymbol{\gamma} = \begin{bmatrix} \gamma_{k=0} \\ \gamma_{k=1} \\ \vdots \\ \gamma_{k=N-1} \end{bmatrix}, \gamma_k = \begin{bmatrix} \gamma_{j=1}(k) \\ \gamma_{j=2}(k) \\ \vdots \\ \gamma_{j=N_j}(k) \end{bmatrix}, \boldsymbol{\gamma}_j = \begin{bmatrix} \gamma_{k=0,j} \\ \gamma_{k=1,j} \\ \vdots \\ \gamma_{k=N-1,j} \end{bmatrix}. \tag{3-2}$$

The combined vector for the overall system $\boldsymbol{\gamma}$, will be referred to as the *resource allocation*, i.e. the division of all available resources over all subsets for the entire horizon. The vector $\boldsymbol{\gamma}_j$, will be referred to as the *allocated resources for subset j*. The set of feasible $\gamma_j(k)$ for time step k, $\mathcal{G}(k)$, is defined as,

$$\mathcal{G}(k) = \{\gamma_j(k) \mid \gamma_j(k) \geq 0 \text{ and } \Lambda_k^{\mathrm{CM}} \gamma_k = P_k^{\mathrm{CM}}, \ \ \forall j, \forall k \in \mathcal{N}\} \tag{3-3}$$

Subset MPC Problem

To have a compact notation, the subset variables are combined over the horizon, with the detailed definitions presented in Appendix A. The resulting input vector is represented as $\mathbf{U}_j$. It is constrained to lie in the set $\mathcal{U}_j$, which represents all local EV constraints. The EV cost vectors are joined in $\mathbf{f}_j$ and the power values corresponding to $\mathbf{U}_j$, are joined in Φ_j^{CM}. When reviewing the centralized problem in (2-18), and as a result of (3-1), the effect of the decoupled global constraints on the subset input vector $\mathbf{U}_j$ is given as,

$$\Phi_j^{\mathrm{CM}} \mathbf{U}_j \leq \boldsymbol{\gamma}_j \tag{3-4}$$

which will be left explicit as it is still related to the network and not individual EVs. The subset MPC optimization problem can then be formulated as,

$$\begin{aligned} J_j(\boldsymbol{\gamma}_j) = \text{minimize} \quad & \mathbf{f}_j^T \mathbf{U}_j \\ \text{s.t.} \quad & \mathbf{U}_j \in \mathcal{U}_j \\ & \Phi_j^{\mathrm{CM}} \mathbf{U}_j \leq \boldsymbol{\gamma}_j \end{aligned} \tag{3-5}$$

with $J_j(\boldsymbol{\gamma}_j)$ as the cost for a given amount of $\boldsymbol{\gamma}_j$ resources and in which $\Phi_j^{\mathrm{CM}}\mathbf{U}_j$ gives the amount of resources consumed by subset j per step in the horizon as a column vector.

Coordinator Problem

The aim is to achieve global optimal system performance. Therefore, the related overall control problem can be defined by aggregating the subset control problems and including the decoupled global constraints in (3-1) and using (3-5), the *coordinator problem* can be defined as,

$$\begin{aligned} \underset{\boldsymbol{\gamma}}{\text{minimize}} \quad & \sum_{j=1}^{N_j} J_j(\boldsymbol{\gamma}_j) \\ \text{s.t.} \quad & \boldsymbol{\Lambda}^{\mathrm{CM}}\boldsymbol{\gamma} = \mathbf{P}^{\mathrm{CM}} \\ & \boldsymbol{\gamma} \geq 0 \end{aligned} \tag{3-6}$$

which also holds for multiple congestion management constraints, under the mild condition that $\boldsymbol{\Lambda}^{\mathrm{CM}}$ and $\mathbf{P}^{\mathrm{CM}}$ allow $\boldsymbol{\Lambda}^{\mathrm{CM}}\boldsymbol{\gamma} = \mathbf{P}^{\mathrm{CM}}$. The coordinator only has limited information about the subset MPC problems, which is limited to the value of the cost function $J_j(\boldsymbol{\gamma}_j)$ and an indication whether the subset would benefit from extra resources. The iterative procedure to approach global optimality, referred to as *resource allocation coordination*, is presented in the next section.

3-2 Resource Allocation Coordination with a Single Global Constraint

This section will introduce the general rationale of the resource allocation coordination architecture. For clarity the derivations will be given for a single overall global constraint, provided by the trading agent, as a row vector for each step k, i.e.

$$\Lambda_k^{\mathrm{CM}} \in \{0,1\}^{1\times N_j} \quad \forall k \in \mathcal{N}. \tag{3-7}$$

Furthermore, for now, it is assumed the coordinator is able to find feasible resource allocations $\boldsymbol{\gamma}_j^z$, for all j and z.

3-2-1 Primal Decomposition

First, inspiration is found in the field of convex optimization. Given a convex problem, with continuous input signals, the decomposition as presented in Equations (3-1), (3-5) and (3-6) is commonly known as primal decomposition. In primal decomposition, the coordinator problem can be solved efficiently using a subgradient method. This is a simple iterative method for solving convex optimization problems. More specifically, given a convex problem with decision variable γ, classical subgradient methods search for the solution to the problem by using the following iteration:

$$\gamma^{z+1} = \Pi\left(\gamma^z - \xi^z h(\gamma^z)\right), \tag{3-8}$$

where z is the iteration step, $h(\gamma^z)$ denotes a subgradient of the objective function of the problem at γ^z, ξ^z denotes the step size at z and $\Pi(\cdot)$ denotes the projection onto the constrained solution space. For convex problems it can be derived that in (3-5), a subgradient of $J_j(\boldsymbol{\gamma}_j)$ for step k is given by $-\lambda_j(k)$, with $\lambda_j(k)$ as the Lagrangian Multiplier corresponding to the decoupled global constraint for step k, given as $\Phi^{\text{CM}}_{k,j} U_{k,j} \leq \gamma_j(k)$ [12, 8]. In particular the subgradient method is given as,

$$\gamma_j^{z+1}(k) = \Pi_{\mathcal{G}(k)} \left(\gamma_j^z(k) + \xi^z \left(\lambda_j^z(k) - \frac{1}{N_j} \sum_{h=1}^{N_j} \lambda_h^z(k) \right) \right), \qquad \forall j \tag{3-9}$$

where $\Pi_{\mathcal{G}(k)}$ is the projection onto $\mathcal{G}(k)$, such that for all iterations (3-1) holds. ξ^z is a diminishing step size that satisfies,

$$\xi^z > 0, \lim_{z \to +\infty} \xi^z = 0, \sum_{z=1}^{+\infty} \xi^z = +\infty, \sum_{z=1}^{+\infty} (\xi^z)^2 < +\infty \tag{3-10}$$

to guarantee, for a convex problem, the global optimum is found [12].

3-2-2 Resource Allocation for Binary Input Systems

Since the power scheduling problem as defined in (2-18) is not convex but an Integer Linear Programming problem (ILP), Lagrangian Multipliers are hard to obtain. For that reason, an alternative has been developed. The following sections will present the design of the DMPC-RA algorithm for binary input systems. Global optimality cannot be guaranteed in finite time, but using feasible iterations performance is shown to be approaching global optimality.

To represent the Lagrangian Multiplier behavior for binary input systems, the auxiliary variable $\psi_j(k)$ will be used from here on. The main purpose of $\psi_j(k)$ in this setting is to indicate how much a subset would benefit from extra resources. In [7],[8, Ch7] a definition for $\psi_j(k)$ is presented for use in a DMPC-RA architecture for hybrid systems. Yet, for a linear cost and linear global constraints, this $\psi_j(k)$ definition becomes independent of the calculated input vector. Therefore, it is not suitable for the current setting.

Since no suitable definition for $\psi_j(k)$ is known for linear cost functions, the following paragraphs will introduce the criteria which should be met.

Resource Allocation Update Function and Stopping Criterion Analysis

To come up with a good definition for $\psi_j(k)$, first, the update function for the resource allocation and stopping criterion are presented and analyzed. The update function for the resource allocation as presented in (3-9) can be used, yet now with $\psi_j(k)$ instead of a Lagrangian Multiplier. The update function for $\gamma_j^{z+1}(k)$ then becomes,

$$\gamma_j^{z+1}(k) = \Pi_{\mathcal{G}(k)} \left(\gamma_j^z(k) + \xi^z \left(\psi_j^z(k) - \frac{1}{N_j} \sum_{h=1}^{N_j} \psi_h^z(k) \right) \right). \tag{3-11}$$

A common stopping criterion in resource allocation algorithms is,

$$\sum_{k\in\mathcal{N}}\sum_{j=1}^{N_j}\Big|\ \gamma_j^{z+1}(k)-\gamma_j^{z}(k)\ \Big| < \varepsilon, \tag{3-12}$$

meaning that the absolute value of the change in allocated resources is strictly smaller than the predefined error tolerance ε. To assess how this influences $\psi_j^z(k)$, assume the projection operator is inactive, i.e. a feasible $\gamma_j^{z+1}(k)$ is found when the projection operator is omitted. Then, the stopping criterion can be rewritten as,

$$\sum_{k\in\mathcal{N}}\sum_{j=1}^{N_j}\left|\ \psi_j^z(k)-\frac{1}{N_j}\sum_{h=1}^{N_j}\psi_h^z(k)\ \right| < \frac{\varepsilon}{\xi^z}. \tag{3-13}$$

This stopping criterion only holds if all individual $\psi_j^z(k)$ are close enough to the system wide average, given as $\frac{1}{N_j}\sum_{j=1}^{N_j}\psi_j^z(k)$. If this happens it means that all subsets are equally disadvantaged. Therefore, $\psi_j^z(k)$ should be defined such, that the difference, between $\psi_j^z(k)$ and the system wide average, converges to zero, only when the system approaches the global optimal solution.

Globally Unconstrained Solution

When the decoupled global constraint, $\Phi_j^{\text{CM}}\mathbf{U}_j \leq \boldsymbol{\gamma}_j$, is omitted in the subset MPC problems and infinite resources are assumed, then a direction and distance between the current solution and the globally unconstrained solution can be found, as is done in [7]. Then, the free cost J_j^{free}, and input vector $\mathbf{U}_j^{\text{free}}$, are defined as,

$$\begin{aligned} J_j^{\text{free}} &= \underset{\mathbf{U}_j\in\mathcal{U}_j}{\text{minimize}}\quad \mathbf{f}_j^T\mathbf{U}_j \\ \mathbf{U}_j^{\text{free}} &= \arg \underset{\mathbf{U}_j\in\mathcal{U}_j,\ \mathbf{f}_j^T\mathbf{U}_j=J_j^{\text{free}}}{\text{minimize}}\quad \sum_{i\in\mathcal{I}_j}\left(T_{j,i}^{\text{ref}}\right) \end{aligned} \tag{3-14}$$

in which $T_{j,i}^{\text{ref}}$ is the time it takes for EV i to reach $E_{j,i}^{\text{MPC,ref}}$. Since multiple input vectors might result in the same free cost, the $\mathbf{U}_j^{\text{free}}$ is defined for which the related power schedules of all EVs reach their reference value the fastest. Note that J_j^{free} and $\mathbf{U}_j^{\text{free}}$ are independent of z and as such only need to be calculated once.

Let $U_{k,j}^{\text{free}}$ be part of $\mathbf{U}_j^{\text{free}}$ for step k, then when all subsets are combined, U_k^{free} is defined as,

$$U_k^{\text{free}} = \Big[(U_{k,j=1}^{\text{free}})^T \quad (U_{k,j=2}^{\text{free}})^T \quad \dots \quad (U_{k,j=N_j}^{\text{free}})^T\Big]^T. \tag{3-15}$$

Then $\Phi_k^{\text{CM}}U_k^{\text{free}}$ gives the vector with the optimal free resource usage for step k. In general, when a single global constraint is used, three cases can be defined:

1. $\Lambda_k^{\text{CM}}\Phi_k^{\text{CM}}U_k^{\text{free}} < P_k^{\text{CM}}$
2. $\Lambda_k^{\text{CM}}\Phi_k^{\text{CM}}U_k^{\text{free}} = P_k^{\text{CM}}$
3. $\Lambda_k^{\text{CM}}\Phi_k^{\text{CM}}U_k^{\text{free}} > P_k^{\text{CM}}$

In the first and second case a resource allocation can be found, such that $\Phi_{k,j}^{\text{CM}} U_{k,j}^{\text{free}} \leq \gamma_j^z(k)$ for all j. In the first case, the flexibility available to reach this resource allocation is defined as $P_k^{\text{CM}} - \Lambda_k^{\text{CM}} \Phi_k^{\text{CM}} U_k^{\text{free}}$. Yet, in the second case no such flexibility is available. The third case occurs when active congestion management constraints are applied and resource consumption needs to be shifted to other steps within the horizon, potentially resulting in higher operational costs.

Representing the Lagrangian Multiplier behavior

First, if $\Phi_{k,j}^{\text{CM}} U_{k,j}^{\text{free}} \leq \gamma_j^z(k)$, the corresponding $\psi_j^z(k)$ is set equal to 0 by definition, as the free input vector $U_{k,j}^{\text{free}}$ is attainable for that step k, as done in [7]. If this holds for all $k \in \mathcal{N}$, then the decoupled global constraints do not pose any restriction in this iteration for this subset. For this case, the optimal input vector is $\mathbf{U}_j^{z,*} = \mathbf{U}_j^{\text{free}}$ and $\psi_j^z(k)=0$ for all k.

Then, to summarize, the main criteria for a suitable $\psi_j^z(k)$ definition include:

1. Represent how much a subset would benefit from extra resources
2. Depend on the calculated input vector $\mathbf{U}_j^z$
3. Converge to the system wide average, only when the global optimal solution is approached

To satisfy the first criterion, $\psi_j^z(k)$ will be calculated in the same units as the cost function, by using the cost vector. Furthermore, $\psi_j^z(k)$ will be set to 0 by definition if $\Phi_{k,j}^{\text{CM}} U_{k,j}^{\text{free}} \leq \gamma_j^z(k)$ and is defined with $\psi_j^z(k) \geq 0$. The second criterion is met by using the difference between $U_{k,j}^{\text{free}}$ and $U_{k,j}^z$. To conclude, the third criterion is satisfied by normalizing to a $0-1$ scale.

Let the non-normalized version of $\psi_j^z(k)$ be noted as $\hat{\psi}_j{}^z(k)$, which is defined as,

$$\hat{\psi}_j{}^z(k) = \begin{cases} 0, & \text{if } \Phi_{k,j}^{\text{CM}} U_{k,j}^{\text{free}} \leq \gamma_j^z(k) \\ \max(0, f_{k,j}^T (U_{k,j}^{\text{free}} - U_{k,j}^z)) & \text{if } \Phi_{k,j}^{\text{CM}} U_{k,j}^{\text{free}} > \gamma_j^z(k) \end{cases} \tag{3-16}$$

with $f_{k,j}$ as the cost vector related to k and j. If $\Phi_{k,j}^{\text{CM}} U_{k,j}^{\text{free}} > \gamma_j^z(k)$ and $f_{k,j}^T (U_{k,j}^{\text{free}} - U_{k,j}^z) > 0$, then the subset will benefit from extra resources.

Let $\hat{\boldsymbol{\psi}}_j{}^z$ be the vector joining $\hat{\psi}_j{}^z(k)$ over the horizon, defined as,

$$\hat{\boldsymbol{\psi}}_j{}^z = \begin{bmatrix} \hat{\psi}_{j=1}^z(k{=}0) & \hat{\psi}_{j=1}^z(k{=}1) & \dots & \hat{\psi}_{j=1}^z(k{=}N{-}1) \end{bmatrix}^T. \tag{3-17}$$

then the final vector $\boldsymbol{\psi}_j^z$ is found by normalizing with respect to the largest entry in $\hat{\boldsymbol{\psi}}_j{}^z$. The vector $\boldsymbol{\psi}_j^z$ is defined as,

$$\boldsymbol{\psi}_j^z = \frac{1}{\max\left(\hat{\boldsymbol{\psi}}_j{}^z\right)} \hat{\boldsymbol{\psi}}_j{}^z. \tag{3-18}$$

such that all values in $\boldsymbol{\psi}_j^z$ are normalized to a $0-1$ scale. This normalization also enables the resource allocation to converge to a solution in which all subsets are equally disadvantaged. Other normalization options have been investigated, the presented version has shown the best results.

Initial Resource Allocation

A simple initial resource allocation, denoted as $\gamma_j^{z=1}$, can be to equally divide the available resources over the subsets. Yet, this choice might lead to an increased risk of an infeasible first iteration, since the calculated schedules at the previous sample time might not be feasible anymore.

Warm starting $\gamma_j^{z=1}$ is a better choice generally, based on the best iteration from the previous sample time. This ensures that all calculated schedules in the previous sample time are still feasible. Yet, the resources for $k{=}N{-}1$ still need to be divided. Let $\gamma_j^{z=1}$ be defined as,

$$\gamma_j^{z=1} = \left[\left(\gamma_j^{\text{best,prev}}(k{=}1{:}N{-}1) \right)^T \quad \frac{1}{N_j}\mathbf{P}^{\text{CM}}(k{=}N{-}1) \right]^T, \quad \forall j \tag{3-19}$$

in which $\gamma_j^{\text{best,prev}}$ represents the allocated resources for subset j for the best global solution, found in the previous sample time. Since the current horizon has moved one step since the last sample time, only the second until the last entry are used from $\gamma_j^{\text{best,prev}}$, i.e. $k{=}1{:}N{-}1$. For the last step in $\gamma_j^{z=1}$, an equal division is used.

3-2-3 Updating the Best Resource Allocation

To enable a good any time MPC algorithm, the evaluation of the global performance will be monitored. The coordinator keeps track of the global performance and the associated resource allocation. Due to the discrete jumps in the system behavior, the global cost does not necessarily monotonically decrease within the iteration procedure. It may present a rise in cost before it decreases. This is expected system behavior and directly related to the discrete jumps. Yet, to be able to return to the best iteration so far, the coordinator will save and update the best iteration as z^{best} and the best allocated resources as γ_j^{best} for all j.

To compute the global system performance, all MPCs send the value of their optimal cost function $J_j^{z,*}(\gamma_j^z)$, related to the allocated resources γ_j^z, to the coordinator at the end of each iteration. Subsequently, the coordinator evaluates if the iteration resulted in a better global performance than the previous best iteration. If $\sum_{j=1}^{N_j} J_j^{z,*}(\gamma_j^z) < J^{\text{best}}$, then a new best resource allocation has been found and the variables z^{best}, J^{best} and γ_j^{best} for all j are updated.

3-2-4 Dealing with Infeasible Resource Allocations

An infeasible resource allocation is any resource allocation for which one or multiple MPCs are not able to find feasible power schedules for the EVs in their subset. This is a result of the binary input model, for which discrete jumps occur in the behavior of the system. A resource allocation is only infeasible if one or multiple MPCs have too little resources available for it to find feasible power schedules for all connected EVs. I.e. the *local constraints* cannot be satisfied. As discussed previously, it is assumed the *global constraints* allow for feasible resource allocations to be found, as these are enforced by the trading agent.

If a subset MPC is not able to find feasible power schedules for all EVs in iteration z, it will set the optimal cost $J_j^{z,*}(\gamma_j^z)$, associated with the allocated resources γ_j^z, to infinity, i.e.

$$J_j^{z,*}(\gamma_j^z) = \infty, \quad \text{if } \gamma_j^z \text{ is an infeasible resource allocation for MPC } j. \tag{3-20}$$

Then, a distinction is made between infeasible initial resource allocations, i.e. γ_j^0 is infeasible, and later iterations. If an infeasible initial resource allocation occurs, the coordinator calculates a new resource allocation in which more resources are allocated to the infeasible subset until this results in a feasible initial resource allocation. If an infeasible resource allocation occurs during the iteration process, the coordinator returns to the best known feasible iteration z^{best} and reuses the associated allocated resources γ_j^{best} for all j. Due to the diminishing step size a different result will be calculated for the next iteration

3-3 Overall Distributed Model Predictive Control with Resource Allocation Algorithm

For the numerical experiments, the coordinator and subset MPCs problems are joined in one algorithm, see Algorithm 1. The steps performed by the coordinator are indicated with C and those by a lower level subset MPC with L.

3-4 Conclusions

This chapter provides the detailed derivation of the DMPC-RA algorithm. The algorithm comprises of a coordinator role and multiple MPCs, each responsible for the control of one subset. The presented coordinator in this chapter is able to handle only one global constraint per step k. If a small extension is applied, the algorithm will be able to deal with multiple global constraints, this is part of the proposed future work.

The following conclusions can be drawn:

- The developed DMPC-RA algorithm will approach global optimal performance for an increasing number of iterations, in which the subset MPCs only have local information.
- The DMPC-RA algorithm uses feasible iterations and can be stopped at any time.
- The proposed method to converge to the global optimal performance works for the developed DMPC-RA algorithm, i.e. the ψ_j definition for binary input systems with a linear cost function.

Algorithm 1 DMPC-RA algorithm for problems with binary variables and a single global constraint, open loop setup.

```
Require: $\mathbf{\Lambda}^{\text{CM}}, \mathbf{P}^{\text{CM}}, \gamma_j^{\text{best,prev}}, p^{\text{b}}, p^{\text{s}}, N, \varepsilon, z^{\max}, \xi^0, \alpha^{\text{d}}$ and $\mathcal{U}_j, \mathcal{I}_j$ for all $j$
Set initial values
C: Set $z {\leftarrow} 0$, $\xi^z {\leftarrow} \xi^0$, $J^{\text{best}} {\leftarrow} \infty$
C: Set $\gamma_j^{z+1} {\leftarrow} \left[ \left(\gamma_j^{\text{best,prev}}(k{=}1{:}N{-}1)\right)^T \quad \frac{1}{N_j}\mathbf{P}^{\text{CM}}(k{=}N{-}1) \right]^T$ for all $j$
C: Set $\gamma_j^z {\leftarrow} \gamma_j^{z+1} {+} 2\varepsilon$ for all $j$

Start iteration loop
while $\sum_{j=1}^{N_j} |\gamma_j^{z+1} {-} \gamma_j^z| > \varepsilon \;||\; z \leq z^{\max}$ do
    C: Set $z {\leftarrow} z{+}1$ and send $\gamma_j^z$ to the subset MPCs

    Solve subset MPC problems
    for $j{=}1{:}N_j$ do
        if $z{=}1$ then
            L: Solve $J_j^{\text{free}} = \underset{\mathbf{U}_j \in \mathcal{U}_j}{\text{minimize}} \quad \mathbf{f}_j^T \mathbf{U}_j$
            L: Solve $\mathbf{U}_j^{\text{free}} = \arg \underset{\mathbf{U}_j \in \mathcal{U}_j,\; \mathbf{f}_j^T \mathbf{U}_j = J_j^{\text{free}}}{\text{minimize}} \quad \sum_{i \in \mathcal{I}_j} \left(T_{j,i}^{\text{ref}}\right)$
        end if
        L: Solve $\mathbf{U}_j^{z,*} {=} \arg \underset{\mathbf{U}_j \in \mathcal{U}_j, \Phi_j^{\text{CM}} \mathbf{U}_j \leq \gamma_j^z}{\min} \quad \mathbf{f}_j^T \mathbf{U}_j$ with associated cost $J_j^{z,*}(\gamma_j^z)$
        L: Update $\psi_j^z$ using (3-16)
        L: Save $\mathbf{U}_j^{z,*}$ and send $\psi_j^z, J_j^{z,*}$ to the coordinator
    end for

    C: Update $\xi^z {=} \frac{\xi^{z-1}}{\alpha^{\text{d}}}$
    if $\sum_{j=1}^{N_j} J_j^{z,*}(\gamma_j^z) {=} \infty$ then
        if z=1 then
            C: Set $z {\leftarrow} z{-}1$
            C: Calculate new $\gamma_j^z$ for all $j$, based on the infeasible subsets, i.e. $J_j^{z,*}(\gamma_j^z) {=} \infty$
        else
            C: Update $\gamma_j^{z+1} = \Pi_{\mathcal{G}(k)} \left(\gamma_j^{\text{best}} {+} \xi^z \left(\psi_j^{z^{\text{best}}} {-} \frac{1}{N_j} \sum_{h=1}^{N_j} \psi_h^{z^{\text{best}}}\right)\right)$ for all $j$
        end if
    else
        Update best iteration
        if $\sum_{j=1}^{N_j} J_j^{z,*}(\gamma_j^z) {<} J^{\text{best}}$ then
            C: Update $J^{\text{best}} = \sum_{j=1}^{N_j} J_j^{z,*}(\gamma_j^z)$, $z^{\text{best}} = z$ and $\gamma_j^{\text{best}} = \gamma_j^z$ for all $j$
        end if
        Solve coordinator problem for next iteration
        C: Update $\gamma_j^{z+1} = \Pi_{\mathcal{G}(k)} \left(\gamma_j^z {+} \xi^z \left(\psi_j^z {-} \frac{1}{N_j} \sum_{h=1}^{N_j} \psi_h^z\right)\right)$ for all $j$
    end if
end while
C: Let local controllers know $z^{\text{best}}$ has best global result
```

Chapter 4

Hierarchical Model Predictive Control using Virtual Batteries

In this chapter the developed Hierarchical Decentralized Model Predictive Control (HDe-MPC) algorithm to solve the power scheduling problem stated in Section 2-1 is introduced. It comprises of a single high-level MPC and several lower-level MPCs, one for each subset. The high-level controller solves a system wide MPC optimization problem based on aggregated information and provides the lower-level controllers with a reference signal. The power schedules for the individual Electric Vehicles (EVs) are then calculated by the lower-level controllers by means of a reference tracking setup. Since the lower-level controllers do not communicate amongst each other, the architecture is called Hierarchical Decentralized MPC. The high-level controller uses a Virtual Battery (VB) model for each subset to represent the aggregated behavior of multiple EVs. One of the main merits of the VB model is that the size, i.e. the number of variables and constraints, is independent of the number of EVs it represents. As a result, the computational cost for the high-level controller remains equal when multiple EVs are represented.

The general lay-out of the HDe-MPC architecture is presented in Figure 4-1, in which both the high-level controller and lower-level controllers act on the same time scale and with the same prediction horizon N. The high-level controller ensures the global network constraints are satisfied and the lower-level controllers ensure that all local EV constraints are satisfied and the operational costs of the subsets are minimized. The detailed information of all individual EVs is only available to the lower-level controllers.

The high-level and lower-level controllers interact using a reference tracking setup with absolute error minimization scheme. Since the lower-level controllers, control EVs with discrete jumps in the dynamic behavior, an error is likely to remain. As such, the lower-level controllers provide the high-level controller with a correction vector $\mathbf{P}_j^{\mathrm{cor}}$, which represents the maximum possible error. The high-level controller uses these correction vectors to create a contracted version of the available resources and calculates the reference signals $\mathbf{P}_j^{\mathrm{ref}}$, based on these contracted resources.

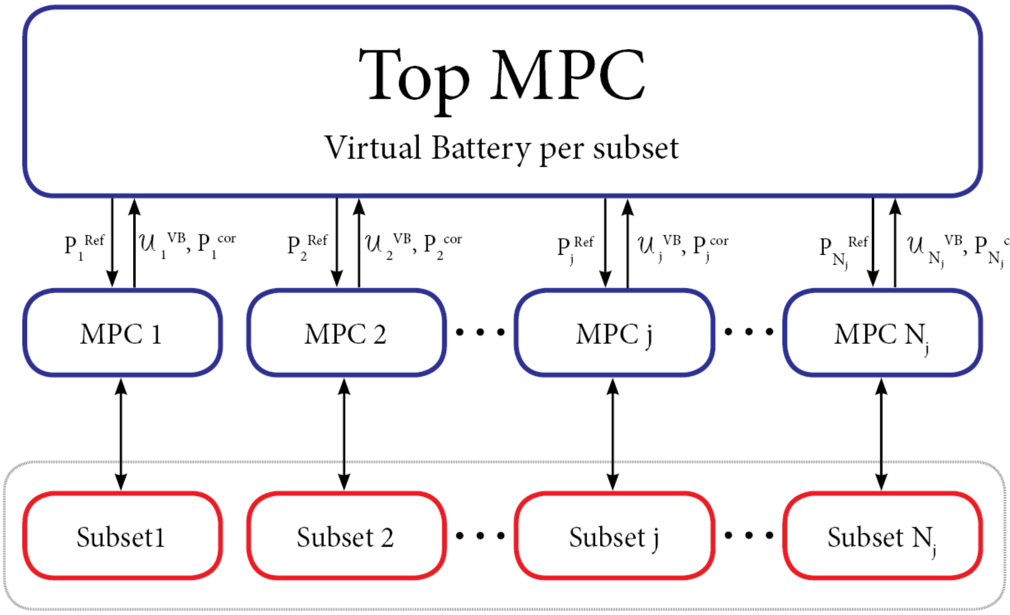

Top MPC
Virtual Battery per subset
P_1^{Ref}
$\mathcal{U}_1^{VB}, P_1^{cor}$
P_2^{Ref}
$\mathcal{U}_2^{VB}, P_2^{cor}$
P_j^{Ref}
$\mathcal{U}_j^{VB}, P_j^{cor}$
$P_{N_j}^{Ref}$
$\mathcal{U}_{N_j}^{VB}, P_{N_j}^{cor}$
MPC 1
MPC 2
MPC j
MPC N_j
Subset1
Subset 2
Subset j
Subset N_j

Figure 4-1: Communication structure of the developed HDe-MPC algorithm

Since the necessary information of local and global constraints is spread out over the system, the communication protocol looks as follows, at every sample time, three steps are performed:

1. The lower-level controllers update the model and constraint variables for the aggregated VB models, represented as $\mathcal{U}_j^{\text{VB}}$, and the correction which needs to be corrected for, represented as $\mathbf{P}_j^{\text{cor}}$, and send this information to the high-level controller.
2. The high-level controller updates the VB models, calculates the power reference signals over the prediction horizon $\mathbf{P}_j^{\text{ref}}$ for all subsets, based on a contracted version of the available resources, and sends this information to the lower-level controllers.
3. The lower-level controllers track the provided power reference signal using an absolute error minimization scheme with a penalty on the absolute error and optimize the operational cost of their associated subset. As standard procedure of MPC, when an optimal schedule is found, the first input of the determined schedule is sent to the individual EVs.

The rest of this chapter is organized as follows. First, the VB model is introduced in Section 4-1 and the update definitions for the aggregated model and constraint variables are presented. In Section 4-3, the optimization problem for the high-level controller is derived. In Section 4-4, the optimization problem for the lower-level controllers is derived and the reference tracking setup is introduced. In Section 4-5, the combined algorithm for the high-level controller and lower-level controllers is presented and discussed. Section 4-6 presents and discusses the results of a numerical experiment for the open loop behavior of the high-level controller, to provide an indication in the use of the VBs. To conclude, Section 4-7 presents the conclusions of the chapter.

4-1 The Virtual Battery Model

Since the VB model needs to represent an entire subset of EVs, it is more extensive than the binary input model used for individual EVs. This section will introduce the details of the model and explain how the necessary variables are obtained from aggregating the individual EV variables. All derivations will be given for subset j.

4-1-1 Decision Variables

The model has three decision variables: $P_j^{\mathrm{VB}}(k)$, $P_j^{\mathrm{VB,d}}(k)$ and $\delta_j^{\mathrm{VB,d}}(k)$, respectively the continuous power reference for subset j, the negative part of the power reference and a binary variable which is set to 1 when $P_j^{\mathrm{VB}}(k)$ is negative. The latter two variables are necessary to be able to incorporate efficiencies in the VB model and use different values for the buy and sell prices. The decision variables for VB j for step k are joined in the input vector $U_{k,j}^{\mathrm{VB}}$ as,

$$U_{k,j}^{\mathrm{VB}} = \begin{bmatrix} P_j^{\mathrm{VB}}(k) \\ P_j^{\mathrm{VB,d}}(k) \\ \delta_j^{\mathrm{VB,d}}(k) \end{bmatrix}, \tag{4-1}$$

with $\{P_j^{\mathrm{VB}}(k), P_j^{\mathrm{VB,d}}(k)\}{\in}\mathbb{R}$ and $\delta_j^{\mathrm{VB,d}}(k){\in}\{0,1\}$. The mutual relations cause the VB model to be non-linear, since $P_j^{\mathrm{VB,d}}(k){=}\delta_j^{\mathrm{VB,d}}(k)P_j^{\mathrm{VB}}(k)$. Yet, this relation can be rewritten as six linear mixed integer inequalities, as presented in the following section.

Mixed Logical Dynamical System Constraints

In this section, the procedure presented in [13] and [14] is used to rewrite the non-linear relation $P_j^{\mathrm{VB,d}}(k){=}\delta_j^{\mathrm{VB,d}}(k)P_j^{\mathrm{VB}}(k)$ into a combination of six linear mixed integer inequalities. Assume that $P_j^{\mathrm{VB}}(k)$ is upper and lower bounded by $P_j^{\mathrm{VB,max}}(k)$ and $P_j^{\mathrm{VB,min}}(k)$, respectively. The variable $\delta_j^{\mathrm{VB,d}}(k)$ is a binary variable that is equal to 1 whenever $P_j^{\mathrm{VB}}(k)$ is negative. Therefore,

$$P_j^{\mathrm{VB}}(k) \leq 0 \iff \delta_j^{\mathrm{VB,d}}(k) = 1 \tag{4-2}$$

which is true, if and only if

$$\begin{aligned} P_j^{\mathrm{VB}}(k) &\leq P_j^{\mathrm{VB,max}}(k)\left(1{-}\delta_j^{\mathrm{VB,d}}(k)\right) \\ -P_j^{\mathrm{VB}}(k) &\leq -\left(P_j^{\mathrm{VB,min}}(k){-}\epsilon\right)\delta_j^{\mathrm{VB,d}}(k){-}\epsilon, \end{aligned} \tag{4-3}$$

in which the small tolerance ϵ is needed to transform a strict constraint into a non-strict constraint, as presented in [13]. Similarly, $P_j^{\mathrm{VB,d}}(k){=}\delta_j^{\mathrm{VB,d}}(k)P_j^{\mathrm{VB}}(k)$ can be represented as,

$$\begin{aligned} P_j^{\mathrm{VB,d}}(k) &\leq P_j^{\mathrm{VB,max}}(k)\delta_j^{\mathrm{VB,d}}(k) \\ P_j^{\mathrm{VB,d}}(k) &\geq P_j^{\mathrm{VB,min}}(k)\delta_j^{\mathrm{VB,d}}(k) \\ P_j^{\mathrm{VB,d}}(k) &\leq P_j^{\mathrm{VB}}(k){-}P_j^{\mathrm{VB,min}}(k)(1{-}\delta_j^{\mathrm{VB,d}}(k)) \\ P_j^{\mathrm{VB,d}}(k) &\geq P_j^{\mathrm{VB}}(k){-}P_j^{\mathrm{VB,max}}(k)(1{-}\delta_j^{\mathrm{VB,d}}(k)). \end{aligned} \tag{4-4}$$

Hence, $P_j^{\text{VB}}(k)$ can have both positive and negative values, while $P_j^{\text{VB,d}}(k)$ is always a non-positive real number. As shown above, the relations between the three variables $P_j^{\text{VB}}(k)$, $P_j^{\text{VB,d}}(k)$ and $\delta_j^{\text{VB,d}}(k)$ can be represented by six linear mixed integer inequalities. These inequalities can be written as follows,

$$\begin{bmatrix} 1 & 0 & P_j^{\text{VB,max}}(k) \\ -1 & 0 & \left(P_j^{\text{VB,min}}(k){-}\epsilon\right) \\ 0 & 1 & -P_j^{\text{VB,max}}(k) \\ 0 & -1 & P_j^{\text{VB,min}}(k) \\ -1 & 1 & -P_j^{\text{VB,min}}(k) \\ 1 & -1 & P_j^{\text{VB,max}}(k) \end{bmatrix} U_{k,j}^{\text{VB}} \leq \begin{bmatrix} P_j^{\text{VB,max}}(k) \\ -\epsilon \\ 0 \\ 0 \\ -P_j^{\text{VB,min}}(k) \\ P_j^{\text{VB,max}}(k) \end{bmatrix}. \tag{4-5}$$

In the following, this equation will be compactly denoted as,

$$A_{k,j}^{\text{MLD}} U_{k,j}^{\text{VB}} \leq b_{k,j}^{\text{MLD}}, \tag{4-6}$$

in which both $A_{k,j}^{\text{MLD}}$ and $b_{k,j}^{\text{MLD}}$ depend on the constraints for $P_j^{\text{VB}}(k)$. The constraint variables, $P_j^{\text{VB,min}}(k)$ and $P_j^{\text{VB,max}}(k)$, are presented and defined in the following sections.

4-1-2 Dynamics of the Virtual Battery Model

This section introduces the dynamics of the VB model. The model uses time dependent variables to be able to best grasp the properties of the EVs it represents. These properties change over time due to arrivals and departures. The dynamics of the VB model can be represented as,

$$\begin{aligned} E_j^{\text{VB}}(k{+}1) = E_j^{\text{VB}}(k){+}\tau\eta_j^{\text{VB,c}}(k)P_j^{\text{VB}}(k){+}\tau\left(\eta_j^{\text{VB,c}}(k){-}\frac{1}{\eta_j^{\text{VB,d}}(k)}\right)P_j^{\text{VB,d}}(k) \\ {+}E_j^{\text{VB,arr}}(k{+}1){-}E_j^{\text{VB,dep}}(k{+}1), \quad \forall k{\in}\mathcal{N}, \end{aligned} \tag{4-7}$$

with $E_j^{\text{VB}}(k)$ as the stored energy in the j-th VB at step k, $E_j^{\text{VB,arr}}(k{+}1)$ and $E_j^{\text{VB,dep}}(k{+}1)$ represent the increase and decrease respectively of the total stored energy as a result of the arrival or departure of EVs and $\eta_j^{\text{VB,c}}(k)$ and $\eta_j^{\text{VB,d}}(k)$ are the charging and discharging efficiencies respectively. The values of $\eta_j^{\text{VB,c}}(k)$ and $\eta_j^{\text{VB,d}}(k)$ are time dependent, because they depend on the efficiencies of the connected EVs. The value of $E_j^{\text{VB,arr}}(k{+}1)$ is available from forecast and the value of $E_j^{\text{VB,dep}}(k{+}1)$ represents the summed State of Charge (SoC) of the EVs which are scheduled to depart in step k. Yet, since it is unknown what the SoC will be of these departing EVs, the minimum attainable SoC values of the departing EVs are used.

The dynamics can be compactly rewritten as,

$$E_j^{\text{VB}}(k{+}1) = E_j^{\text{VB}}(k){+}B_{k,j}^{\text{VB}}U_{k,j}^{\text{VB}}{+}w_{k,j}^{\text{VB}}, \quad \forall k{\in}\mathcal{N}, \tag{4-8}$$

with the time-dependent vector $B_{k,j}^{\mathrm{VB}}$ and variable $w_{k,j}^{\mathrm{VB}}$ defined as,

$$\begin{aligned} B_{k,j}^{\mathrm{VB}} &= \left[\begin{array}{ccc} \tau\eta_j^{\mathrm{VB,c}}(k) & \tau\left(\eta_j^{\mathrm{VB,c}}(k)-\frac{1}{\eta_j^{\mathrm{VB,d}}(k)}\right) & 0 \end{array}\right], \\ w_{k,j}^{\mathrm{VB}} &= E_j^{\mathrm{VB,arr}}(k+1)-E_j^{\mathrm{VB,dep}}(k+1). \end{aligned} \tag{4-9}$$

4-1-3 Aggregation of the EV Variables per Subset

In each sampling time, the time-dependent model and constraint variables of the VBs are updated by the lower level controllers out of the individual EV parameters and send to the high-level controller. The time-dependent model variables are $\eta_j^{\mathrm{VB,c}}(k), \eta_j^{\mathrm{VB,d}}(k), E_j^{\mathrm{VB,arr}}(k+1)$ and $E_j^{\mathrm{VB,dep}}(k+1)$. The time-dependent constraint variables are $E_j^{\mathrm{VB,min}}(k+1), E_j^{\mathrm{VB,max}}(k+1), P_j^{\mathrm{VB,min}}(k)$ and $P_j^{\mathrm{VB,max}}(k)$.

Because $P_j^{\mathrm{VB}}(k)$ is a continuous variable, the constraints on $E_j^{\mathrm{VB}}(k+1)$ and $P_j^{\mathrm{VB}}(k)$ must be tight, such that only attainable values for $E_j^{\mathrm{VB}}(k+1)$ and $P_j^{\mathrm{VB}}(k)$ can be obtained. This means that the constraints need to be more strict than the SoC constraints used for the individual EV SoC envelopes. The binary input model only allowed for three discrete power options per time step, while the VB uses a continuous power signal and as such needs different constraints. The general form of the constraints on $E_j^{\mathrm{VB}}(k)$ and $P_j^{\mathrm{VB}}(k)$ is,

$$\begin{aligned} E_j^{\mathrm{VB,min}}(k+1) \leq E_j^{\mathrm{VB}}(k+1) \leq E_j^{\mathrm{VB,max}}(k+1), &\quad \forall k \in \mathcal{N} \\ P_j^{\mathrm{VB,min}}(k) \leq P_j^{\mathrm{VB}}(k) \leq P_j^{\mathrm{VB,max}}(k), &\quad \forall k \in \mathcal{N} \end{aligned} \tag{4-10}$$

Tight SoC Constraints

As a result of the aggregation, tight constraints on the SoC need to be derived to be able to represent the subset behavior correctly.

A simple constraint definition can be obtained by summing the individual EV SoC constraints as,

$$\begin{aligned} \hat{E}_j^{\mathrm{VB,min}}(k+1) &= \sum_{i \in \mathcal{I}_j(k)} \left(E_{j,i}^{\mathrm{MPC,min}}(k+1)\right) \quad \forall k \in \mathcal{N} \\ \hat{E}_j^{\mathrm{VB,max}}(k+1) &= \sum_{i \in \mathcal{I}_j(k)} \left(E_{j,i}^{\mathrm{MPC,max}}(k+1)\right) \quad \forall k \in \mathcal{N}, \end{aligned} \tag{4-11}$$

with $E_{j,i}^{\mathrm{MPC,min}}(k+1)$ and $E_{j,i}^{\mathrm{MPC,min}}(k+1)$ as determined in (2-10). These constraints will be referred to as *static SoC constraints*, since they are based on theoretical constraint values, not incorporating all available information about the current options of an EV. The EV SoC constraints in (2-10) are not tight as they include regions of the MPC EV SoC envelope which cannot be reached.

An improved constraint definition can be obtained when the unreachable regions are excluded and only attainable values are incorporated in the definition. The improved constraint definitions will be referred to as *dynamic SoC constraints*. The difference between the static and dynamic SoC constraints are visualized in Figure 4-2. The static SoC envelope is given by

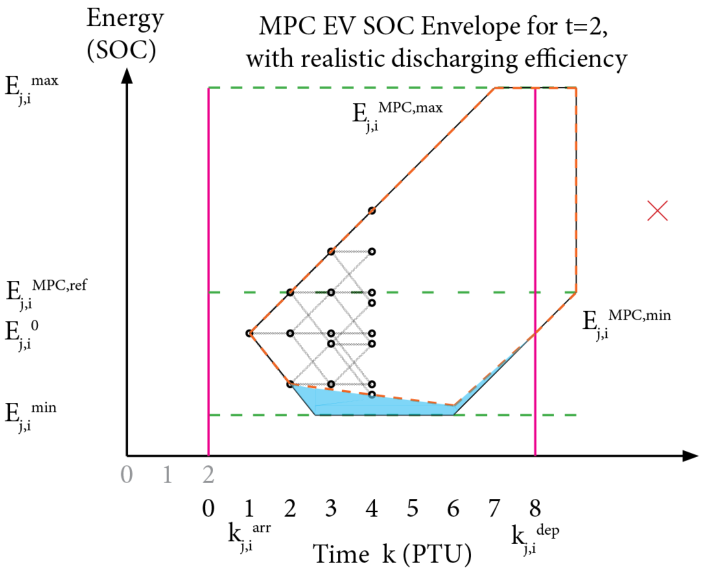


Figure 4-2: Difference between the static and dynamic MPC EV SoC envelope

The static SoC envelope is given by the solid black lines, the dynamic envelope is given by the dotted orange lines. The difference between the two envelopes for this example is colored in blue. The red cross is not know to the controller, but represents the overall $E_{j,i}^{\text{ref}}$ at $T_{j,i}^{\text{dep}}$.

the solid black lines, the dynamic envelope is given by the dotted orange lines. The difference between the two envelopes for this example is colored in blue. As such, the blue region represents the unreachable region. If simple SoC constraints are used, adding all these unreachable regions for all individual EVs would result in a large mismatch between the VB behavior and the actual behavior of the EVs in the subset.

The dynamic SoC constraints are defined using the set $\mathcal{E}_{j,i}^{\text{MPC}}(k{+}1)$. This set was defined in (2-12) and contains all attainable values for the SoC of EV i for step $k{+}1$. As such, $E_j^{\text{VB,min}}(k{+}1)$ and $E_j^{\text{VB,max}}(k{+}1)$ are defined as,

$$\begin{aligned} E_j^{\text{VB,min}}(k{+}1) &= \sum_{i\in\mathcal{I}_j(k)} \left(\min\left(\mathcal{E}_{j,i}^{\text{MPC}}(k{+}1)\right)\right) \quad \forall k\in\mathcal{N} \\ E_j^{\text{VB,max}}(k{+}1) &= \sum_{i\in\mathcal{I}_j(k)} \left(\max\left(\mathcal{E}_{j,i}^{\text{MPC}}(k{+}1)\right)\right) \quad \forall k\in\mathcal{N} \end{aligned} \tag{4-12}$$

Tight Power Constraints

As a result of the aggregation, tight constraints on $P_j^{\text{VB}}(k)$ need to be derived to be able to represent the subset behavior correctly.

A simple constraint definition can be obtained by summing the individual EV charging and discharging power values as,

$$\begin{aligned}\hat{P}_j^{\mathrm{VB,min}}(k) &= \sum_{i\in\mathcal{I}_j(k)} \left(P_{j,i}^{\mathrm{c}}\right) \quad \forall k\in\mathcal{N}\\ \hat{P}_j^{\mathrm{VB,max}}(k) &= \sum_{i\in\mathcal{I}_j(k)} \left(P_{j,i}^{\mathrm{d}}\right) \quad \forall k\in\mathcal{N}.\end{aligned} \tag{4-13}$$

These constraint definitions will be referred to as *static power constraints*, since they are based on theoretical values, not incorporating the information about the actual charging and discharging options of an EV. For example, when the departure time of an EV approaches, the EV might be forced to keep on charging until his departure. This information is not incorporated in the static power constraints.

Based on the initial SoC of an EV, it can be determined what charging and discharging options the EV has over the prediction horizon using the set $\mathcal{E}_{j,i}^{\mathrm{MPC}}(k)$. The constraint definitions taking these available charging and discharging options into account will be referred to as *dynamic power constraints.* Within the MPC EV SoC envelope, five non-intersecting sets can be identified. The dynamic power constraints will be defined based on in which set the initial SoC $E_{j,i}^0$ is for that sampling time. The five set definitions, using strict inequalities, are defined as,

$$\begin{aligned}\mathcal{E}_{j,i}^1 = \{E_{j,i}(k) \mid\ & E_{j,i}(k)\in\mathcal{E}_{j,i}^{\mathrm{MPC}},\ E_{j,i}(k)<E_{j,i}^{\mathrm{ref}} \text{ and}\\ & E_{j,i}(k)<E_{j,i}^{\mathrm{ref,slope}}(k)+\tau P_{j,i}^{\mathrm{c}}\eta_{j,i}^{\mathrm{c}},\ \forall k\in\mathcal{N}_{j,i}\}\\ \mathcal{E}_{j,i}^2 = \{E_{j,i}(k) \mid\ & E_{j,i}(k)\in\mathcal{E}_{j,i}^{\mathrm{MPC}}\backslash\{\mathcal{E}_{j,i}^1\},\ E_{j,i}(k)<E_{j,i}^{\mathrm{ref}}+\tau P_{j,i}^{\mathrm{d}}\frac{1}{\eta_{j,i}^{\mathrm{d}}} \text{ and}\\ & E_{j,i}(k)<E_{j,i}^{\mathrm{ref,slope}}(k)+\tau P_{j,i}^{\mathrm{c}}\eta_{j,i}^{\mathrm{c}}+\tau P_{j,i}^{\mathrm{d}}\frac{1}{\eta_{j,i}^{\mathrm{d}}},\ \forall k\in\mathcal{N}_{j,i}\}\\ \mathcal{E}_{j,i}^3 = \{E_{j,i}(k) \mid\ & E_{j,i}(k)\in\mathcal{E}_{j,i}^{\mathrm{MPC}}\backslash\{\mathcal{E}_{j,i}^1,\mathcal{E}_{j,i}^2\} \text{ and}\\ & E_{j,i}(k)<E_{j,i}^{\mathrm{min}}+\tau P_{j,i}^{\mathrm{d}}\frac{1}{\eta_{j,i}^{\mathrm{d}}},\ \forall k\in\mathcal{N}_{j,i}\}\\ \mathcal{E}_{j,i}^4 = \{E_{j,i}(k) \mid\ & E_{j,i}(k)\in\mathcal{E}_{j,i}^{\mathrm{MPC}} \text{ and } E_{j,i}(k)>E_{j,i}^{\mathrm{max}}-\tau P_{j,i}^{\mathrm{c}}\eta_{j,i}^{\mathrm{c}},\ \forall k\in\mathcal{N}_{j,i}\}\\ \mathcal{E}_{j,i}^0 = \{E_{j,i}(k) \mid\ & E_{j,i}(k)\in\mathcal{E}_{j,i}^{\mathrm{MPC}}\backslash\{\mathcal{E}_{j,i}^1,\mathcal{E}_{j,i}^2,\mathcal{E}_{j,i}^3,\mathcal{E}_{j,i}^4\},\ \forall k\in\mathcal{N}_{j,i}\}\end{aligned} \tag{4-14}$$

in which the \ is used as set exclusion, i.e. $\mathcal{E}_{j,i}^{\mathrm{MPC}}\backslash\{\mathcal{E}_{j,i}^1\}$ is the set of elements in $\mathcal{E}_{j,i}^{\mathrm{MPC}}$ but not in $\mathcal{E}_{j,i}^1$. The variable $E_{j,i}^{\mathrm{ref,slope}}(k)$ was introduced in Equation (2-9). The five sets are visualized in Figure 4-3.

Let $P_{j,i}^{\mathrm{min}}(k)$ and $P_{j,i}^{\mathrm{max}}(k)$ denote the minimum and maximum power value attainable for EV i at step k. Then, if the initial SoC of EV i is within $\mathcal{E}_{j,i}^0$, the dynamic power constraints will be equal to the static power constraints, i.e. $P_{j,i}^{\mathrm{min}}(k)=-P_{j,i}^{\mathrm{d}}$ and $P_{j,i}^{\mathrm{max}}(k)=P_{j,i}^{\mathrm{c}}$ for all $k\in\mathcal{N}_{j,i}$.

Yet, if the the initial SoC of EV i is within one of the other four sets, $P_{j,i}^{\mathrm{min}}(k)$ or $P_{j,i}^{\mathrm{max}}(k)$ will be limited. A distinction can be made whether the initial SoC is within $\mathcal{E}_{j,i}^1$ or $\mathcal{E}_{j,i}^2$, then the power options for the remainder of the horizon are affected, or within $\mathcal{E}_{j,i}^3$ or $\mathcal{E}_{j,i}^4$, then only the first time step k=0 is affected. For example, if the initial SoC is within $\mathcal{E}_{j,i}^3$ and the EV charges in the first time step, then the second time step, k=1, allows for all three power options: charging, idle and discharging.

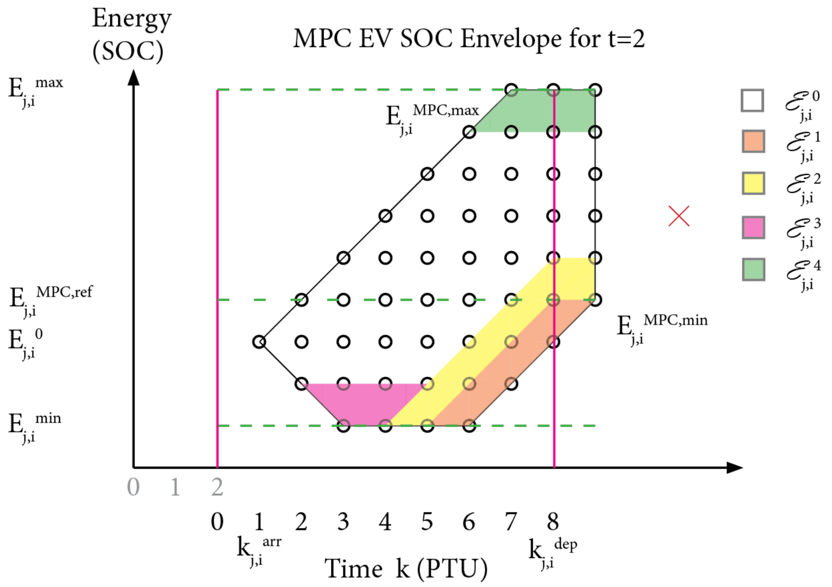


Figure 4-3: Schematic representation of the set definitions based on initial conditions within the MPC EV SoC envelope for t=2.

Simplified visualization with the legend indicating the mutually exclusive sets. The red cross is not know to the controller, but represents the overall $E^{\text{ref}}_{j,i}$ at $T^{\text{dep}}_{j,i}$.

The minimum and maximum attainable power values for EV i are defined as,

$$\begin{aligned}
P^{\text{min}}_{j,i}(k) &= \begin{cases} -P^{\text{d}}_{j,i}, & \text{if } E^0_{j,i}{\in}\{\mathcal{E}^0_{j,i},\mathcal{E}^4_{j,i}\} \\ 0, & \text{if } E^0_{j,i}{\in}\{\mathcal{E}^2_{j,i},\mathcal{E}^3_{j,i}\} \\ P^{\text{c}}_{j,i} & \text{if } E^0_{j,i}{\in}\{\mathcal{E}^1_{j,i}\} \end{cases} && k = k^{\text{arr}}_{j,i} \\
P^{\text{max}}_{j,i}(k) &= \begin{cases} 0, & \text{if } E^0_{j,i}{\in}\{\mathcal{E}^4_{j,i}\} \\ P^{\text{c}}_{j,i}, & \text{if } E^0_{j,i}{\in}\{\mathcal{E}^0_{j,i},\mathcal{E}^2_{j,i},\mathcal{E}^3_{j,i},\mathcal{E}^4_{j,i}\} \end{cases} && k = k^{\text{arr}}_{j,i} \\
P^{\text{min}}_{j,i}(k) &= \begin{cases} -P^{\text{d}}_{j,i}, & \text{if } E^0_{j,i}{\in}\{\mathcal{E}^0_{j,i},\mathcal{E}^3_{j,i},\mathcal{E}^4_{j,i}\} \\ 0, & \text{if } E^0_{j,i}{\in}\{\mathcal{E}^2_{j,i}\} \\ P^{\text{c}}_{j,i} & \text{if } E^0_{j,i}{\in}\{\mathcal{E}^1_{j,i}\} \end{cases} && \forall k{\in}\mathcal{N}_{j,i}\backslash\{k^{\text{arr}}_{j,i}\} \\
P^{\text{max}}_{j,i}(k) &= P^{\text{c}}_{j,i} && \forall k{\in}\mathcal{N}_{j,i}\backslash\{k^{\text{arr}}_{j,i}\}
\end{aligned} \tag{4-15}$$

with which the dynamic MPC VB power constraints can be defined as,

$$\begin{aligned}
P^{\text{VB,min}}_j(k) &= \sum_{i\in\mathcal{I}_j(k)} P^{\text{min}}_{j,i}(k) \quad \forall k{\in}\mathcal{N} \\
P^{\text{VB,max}}_j(k) &= \sum_{i\in\mathcal{I}_j(k)} P^{\text{max}}_{j,i}(k) \quad \forall k{\in}\mathcal{N}.
\end{aligned} \tag{4-16}$$

The values for $P^{\text{VB,min}}_j(k)$ and $P^{\text{VB,max}}_j(k)$ can be directly incorporated in the Mixed Logical Dynamical system (MLD) constraints, which were presented in (4-5)

Model Variable Updates

The charging and discharging efficiencies of VB j are calculated as the mean weighted average of the individual EV efficiencies, with the corresponding EV charging and discharging power values as weights respectively. As such, $\eta_j^{\mathrm{VB,c}}(k)$ and $\eta_j^{\mathrm{VB,d}}(k)$ are defined as,

$$\begin{aligned} \eta_j^{\mathrm{VB,c}}(k) &= \frac{\sum_{i\in\mathcal{I}_j(k)}\left(\eta_{j,i}^{\mathrm{c}}P_{j,i}^{\mathrm{c}}\right)}{\sum_{i\in\mathcal{I}_j(k)}\left(P_{j,i}^{\mathrm{c}}\right)} \quad \forall k\in\mathcal{N} \\ \eta_j^{\mathrm{VB,d}}(k) &= \frac{\sum_{i\in\mathcal{I}_j(k)}\left(\eta_{j,i}^{\mathrm{d}}P_{j,i}^{\mathrm{d}}\right)}{\sum_{i\in\mathcal{I}_j(k)}\left(P_{j,i}^{\mathrm{d}}\right)} \quad \forall k\in\mathcal{N}. \end{aligned} \tag{4-17}$$

The initial SoC for E_j^{VB} is calculated as the sum of the initial individual EV SoC as,

$$E_j^{\mathrm{VB}}(0) = \sum_{i\in\mathcal{I}_j(0)}\left(E_{j,i}^{0}\right). \tag{4-18}$$

The changes in SoC, $E_j^{\mathrm{VB,arr}}$ and $E_j^{\mathrm{VB,dep}}$ related to the arrival and departure of EVs respectively, are defined as,

$$\begin{aligned} E_j^{\mathrm{VB,arr}}(k{+}1) &= \sum_{i\in\mathcal{I}_j(k)}\left(E_{j,i}^{\mathrm{arr}}\right)|^{k+1=k_{j,i}^{\mathrm{arr}}} && \forall k\in\mathcal{N}\backslash\{N{-}1\} \\ E_j^{\mathrm{VB,dep}}(k{+}2) &= \sum_{i\in\mathcal{I}_j(k)}\left(E_{j,i}^{\mathrm{MPC,min}}(k{+}1)\right)|^{k=k_{j,i}^{\mathrm{dep}}} && \forall k\in\mathcal{N} \end{aligned} \tag{4-19}$$

with $k_{j,i}^{\mathrm{arr}}$ and $k_{j,i}^{\mathrm{dep}}$ as the arrival and departure time within the MPC horizon, assumed known to the responsible lower-level controller and $E_{j,i}^{\mathrm{MPC,min}}$ as defined in (2-10). The value of $E_j^{\mathrm{VB,arr}}(k{+}1)$ is defined for the next time step as it is used in the calculation of the prediction of the SoC for the next time step. The value of $E_j^{\mathrm{VB,dep}}(k{+}2)$ is obtained by using $E_j^{\mathrm{VB,min}}(k{+}1)|^{k=k_{j,i}^{\mathrm{dep}}}$, since $k_{j,i}^{\mathrm{dep}}$ indicates the last step for which a control input is calculated and the VB SoC is only affected after the EV has left, so with two steps delay with respect to $k_{j,i}^{\mathrm{dep}}$. For example, if an EV has a $k_{j,i}^{\mathrm{dep}}{=}0$, the last SoC will be predicted for $k{=}1$ and this will affect the VB in step $k{=}2$ as only then the EV has departed. As such the first two steps of the horizon are not affected by departing EVs as this change is already incorporated in the initial SoC and the set $\mathcal{I}_j(k)$. The first two steps of $E_j^{\mathrm{VB,dep}}$ are set to 0,

$$\begin{aligned} E_j^{\mathrm{VB,dep}}(0) &= 0 \\ E_j^{\mathrm{VB,dep}}(1) &= 0. \end{aligned} \tag{4-20}$$

4-1-4 Virtual Battery Variable Update by Lower-Level Controllers

At the start of each sampling time, the lower-level controllers update the VB variables and send the set $\mathcal{U}_j^{\mathrm{VB}}$ to the high-level controller. To this end, the vector U_j^{VB} is defined over the prediction horizon as,

$$U_j^{\mathrm{VB}} = \begin{bmatrix} U_{k=0,j}^{\mathrm{VB}} \\ U_{k=1,j}^{\mathrm{VB}} \\ \vdots \\ U_{k=N-1,j}^{\mathrm{VB}} \end{bmatrix}. \tag{4-21}$$

The local constraints for VB j consist of the MLD constraints in (4-6) and the constraints in (4-12), of which the constraints on the power $P_j^{\mathrm{VB}}(k)$ can be directly incorporated in the MLD constraints. These constraints can be written, using substitution for the system dynamics as presented in (4-8) and (4-18), as constraints on the the vector U_j^{VB} in set notation as,

$$U_j^{\mathrm{VB}} \in \mathcal{U}_j^{\mathrm{VB}}. \tag{4-22}$$

The detailed derivation of the set $\mathcal{U}_j^{\mathrm{VB}}$ is presented in Appendix A.

4-2 Decomposing the Optimization Problem of the System

To use a HDe-MPC architecture, the optimization problem defined (2-18) has to be decomposed. Since the dynamics, cost functions and local constraints are defined per EV, only the global constraints have to be decoupled. This decoupling is provided by the high-level controller, which ensures the global constraints are satisfied and provides the lower-level controllers with a power reference signal over the horizon. The high-level controller uses a VB model per subset to model the aggregated behavior of the subsets.

The lower-level controllers are set with the responsibility to track the provided power reference using an absolute error minimization scheme. As such, the lower level controllers only have local information and due to the discrete jumps, a tracking error will most likely remain. The maximum value of this error can be approximated beforehand and corrected for using a resource contraction within the high-level controller. The value of the error which needs to be corrected for is denoted as $P_{k,j}^{\mathrm{cor}}$, for subset j at time step k, and can be calculated as,

$$P_{k,j}^{\mathrm{cor}} = \frac{1}{2} \max_{i \in \mathcal{I}_j(k)} P_{j,i}^{\mathrm{c}}. \tag{4-23}$$

which is calculated by the lower-level controllers. Since the error is minimized, the maximum value of the error is half of the largest connected charging power $P_{j,i}^{\mathrm{c}}$. This is a conservative contraction, especially when multiple subsets are jointly constrained. The vector $\mathbf{P}_j^{\mathrm{cor}}$ is defined by joining $P_{k,j}^{\mathrm{cor}}$ over the horizon,

$$\mathbf{P}_j^{\mathrm{cor}} = \begin{bmatrix} P_{k=0,j}^{\mathrm{cor}} & P_{k=1,j}^{\mathrm{cor}} & \dots & P_{k=N-1,j}^{\mathrm{cor}} \end{bmatrix}^T. \tag{4-24}$$

4-3 MPC Optimization Problem Formulation for the High-Level Controller

This section will derive the high-level controller MPC optimization problem. At the start of each sampling time, the high-level controller receives an update from the lower-level controllers including $\mathcal{U}_j^{\mathrm{VB}}$ and $\mathbf{P}_j^{\mathrm{cor}}$. The input vector $\mathbf{U}^{\mathrm{high}}$ is defined by joining the VB input vectors as,

$$\mathbf{U}^{\mathrm{high}}{=}\begin{bmatrix} U_{k=0}^{\mathrm{high}} \\ U_{k=1}^{\mathrm{high}} \\ \vdots \\ U_{k=N-1}^{\mathrm{high}} \end{bmatrix}, U_k^{\mathrm{high}}{=}\begin{bmatrix} U_{k,j=1}^{\mathrm{VB}} \\ U_{k,j=2}^{\mathrm{VB}} \\ \vdots \\ U_{k,j=N_j}^{\mathrm{VB}} \end{bmatrix}, \tag{4-25}$$

with $U_{k,j}^{\mathrm{VB}}$ as the input vector for VB j per step k, defined previously.

Given the resource contraction $\mathbf{P}_j^{\mathrm{cor}}$ for all subsets, the entries are rewritten as,

$$P_k^{\mathrm{cor}} = \begin{bmatrix} P_{k,j=1}^{\mathrm{cor}} & P_{k,j=2}^{\mathrm{cor}} & \cdots & P_{k,j=N_j}^{\mathrm{cor}} \end{bmatrix}^T, \tag{4-26}$$

and subsequently as,

$$\mathbf{P}^{\mathrm{CM,cor}} = \mathbf{\Lambda}^{\mathrm{CM}} \begin{bmatrix} P_{k=0}^{\mathrm{cor}} \\ P_{k=1}^{\mathrm{cor}} \\ \vdots \\ P_{k=N-1}^{\mathrm{cor}} \end{bmatrix}. \tag{4-27}$$

The vector $\mathbf{P}^{\mathrm{CM,cor}}$ consists of the added individual resource corrections, in the same format as $\mathbf{P}^{\mathrm{CM}}$. As such the contracted resource vector $\mathbf{P}^{\mathrm{CM,cont}}$ is defined as,

$$\mathbf{P}^{\mathrm{CM,cont}}{=}\mathbf{P}^{\mathrm{CM}}{-}\mathbf{P}^{\mathrm{CM,cor}} \tag{4-28}$$

Using the contracted resource vector, the global constraints can be written as,

$$\mathbf{\Lambda}^{\mathrm{CM}}\mathbf{\Phi}^{\mathrm{CM,high}}\mathbf{U}^{\mathrm{high}}{\leq}\mathbf{P}^{\mathrm{CM,cont}} \tag{4-29}$$

in which $\mathbf{\Phi}^{\mathrm{CM,high}}$ is used for the coupling between $\mathbf{\Lambda}^{\mathrm{CM}}$ and $\mathbf{U}^{\mathrm{high}}$.

The cost vector for each VB at step k is defined as,

$$f_{k,j}^{\mathrm{high}}{=}\begin{bmatrix} p^{\mathrm{b}}(k) \\ p^{\mathrm{b}}(k){-}p^{\mathrm{s}}(k) \\ 0 \end{bmatrix} \tag{4-30}$$

which is in this form as a result of the MLD reformulation.

Compact Form of the High-Level Controller Problem

To simplify the notation, Appendix A presents the detailed derivation in which the remaining variables are joined over the VBs and over the prediction horizon, which will be omitted here for brevity. The cost vectors are joined such that the cost vector for the high-level controller can be written as $\mathbf{f}^{\mathrm{high}}$. The VB constraints sets are joined rewritten, such that they can be written as $\mathbf{U}^{\mathrm{high}}{\in}\mathcal{U}^{\mathrm{high}}$ and the detailed variable definitions for $\mathbf{\Phi}^{\mathrm{CM,high}}$ is provided in Appendix A. With these compact notations, the high-level controller optimization problem can be formulated as Mixed Integer Linear Programming problem (MILP) as,

$$\begin{aligned} \underset{\mathbf{U}^{\mathrm{high}}}{\text{minimize}} \quad & (\mathbf{f}^{\mathrm{high}})^T\mathbf{U}^{\mathrm{high}} \\ \text{s.t.} \quad & \mathbf{U}^{\mathrm{high}}{\in}\mathcal{U}^{\mathrm{high}} \\ & \mathbf{\Lambda}^{\mathrm{CM}}\mathbf{\Phi}^{\mathrm{CM,high}}\mathbf{U}^{\mathrm{high}}{\leq}\mathbf{P}^{\mathrm{CM,cont}}. \end{aligned} \tag{4-31}$$

The power reference $\mathbf{P}_j^{\text{ref}}$ for the lower-level controllers will be derived from the resulting optimal solution of the high-level optimization problem $\mathbf{U}^{\text{high},*}$. The individual power values per VB j at time step k, $P_j^{\text{VB}}(k)$, will be joined per j to form $\mathbf{P}_j^{\text{ref}}$ as,

$$\mathbf{P}_j^{\text{ref}} = \begin{bmatrix} P_j^{\text{VB}}(k{=}0) & P_j^{\text{VB}}(k{=}1) & \dots & P_j^{\text{VB}}(k{=}N{-}1) \end{bmatrix}^T. \tag{4-32}$$

4-4 MPC Optimization Problem Formulation for the Lower-Level Controllers

This section will derive the lower-level controller MPC optimization problem. Each subset has a corresponding lower-level controller and some of the derivations and variables presented in (3-5) can be reused, which are defined in detail in Appendix A.

After the high-level controller has calculated the power reference $\mathbf{P}_j^{\text{ref}}$ for subset j, the lower-level controller will minimize the absolute error between this reference and the power interaction of the subset. The resulting error is defined as $\boldsymbol{\theta}_j$ and will be penalized and added to the cost function in order to minimize it. The input vector for subset j is represented by $\mathbf{U}_j$ and was already used in (3-5).

Absolute error minimization

The absolute error vector $\boldsymbol{\theta}_j$ is defined as,

$$\left|\Phi_j^{\text{CM}}\mathbf{U}_j - \mathbf{P}_j^{\text{ref}}\right| = \boldsymbol{\theta}_j, \tag{4-33}$$

with $|\cdot|$ as the element-wise absolute value and Φ_j^{CM} as used in (3-5), consisting of the charging and discharging power values corresponding to $\mathbf{U}_j$. This can be rewritten as a combination of three inequalities as,

$$\begin{aligned} \Phi_j^{\text{CM}}\mathbf{U}_j - \boldsymbol{\theta}_j &\leq \mathbf{P}_j^{\text{ref}} \\ -\Phi_j^{\text{CM}}\mathbf{U}_j - \boldsymbol{\theta}_j &\leq -\mathbf{P}_j^{\text{ref}} \\ \boldsymbol{\theta}_j &\geq 0. \end{aligned} \tag{4-34}$$

Now, both $\mathbf{U}_j$ and $\boldsymbol{\theta}_j$ are optimization variables. They can be put in set notation as,

$$\mathcal{U}_j^{\text{low,slack}} = \{\begin{bmatrix} \mathbf{U}_j^T & \boldsymbol{\theta}_j^T \end{bmatrix}^T \mid \Phi_j^{\text{CM}}\mathbf{U}_j - \boldsymbol{\theta}_j \leq \mathbf{P}_j^{\text{ref}},\ -\Phi_j^{\text{CM}}\mathbf{U}_j - \boldsymbol{\theta}_j \leq -\mathbf{P}_j^{\text{ref}} \text{ and } \boldsymbol{\theta}_j \geq 0\}. \tag{4-35}$$

The penalty on the absolute error vector is given as $(\mathbf{f}_j^{\text{slack}})^T\boldsymbol{\theta}_j$, this is incorporated in the cost function of the lower-level controller. The penalty vector $\mathbf{f}_j^{\text{slack}}$ can be used as a tuning parameter and set by the user. In order to provide reference tracking, it needs to be greater than the cost vector, i.e. $\mathbf{f}_j^{\text{slack}} > \mathbf{f}_j$.

4-4-1 Compact Form of the Lower-Level Controller Problem

To compactly represent both the local EV and the absolute reference tracking constraints, with a small abuse of notation, they are joined in set notation in $\mathcal{U}_j^{\text{low,final}}$ as,

$$\mathcal{U}_j^{\text{low,final}} = \bigcap\{\mathcal{U}_j, \mathcal{U}_j^{\text{low,slack}}\}, \tag{4-36}$$

with $\mathcal{U}_j$ as used in (3-5). Note that both $\mathbf{U}_j$ and $\boldsymbol{\theta}_j$ are constrained to lie in the set $\mathcal{U}_j^{\text{low,final}}$. The lower-level controller problem can then be compactly defined as,

$$\begin{aligned} \underset{\mathbf{U}_j}{\text{minimize}} \quad & \begin{bmatrix} \mathbf{f}_j^T & (\mathbf{f}_j^{\text{slack}})^T \end{bmatrix} \begin{bmatrix} \mathbf{U}_j^T \\ \boldsymbol{\theta}_j^T \end{bmatrix} \\ \text{s.t.} \quad & \begin{bmatrix} \mathbf{U}_j^T & \boldsymbol{\theta}_j^T \end{bmatrix}^T \in \mathcal{U}_j^{\text{low,final}}. \end{aligned} \tag{4-37}$$

4-5 Overall Hierarchical Decentralized Model Predictive Control Algorithm

For the numerical experiments, the high-level and lower-level control problems are joined in one algorithm, shown in Algorithm 2. The steps performed by the high-level controller are indicated with H and those by a lower-level controller with L.

As a result of the discrete jumps and the allowed error in the reference tracking setup, the lower-level controllers can calculate power schedules which cause the high-level control problem to be infeasible in the next sampling time. If this happens, the high-level controller reduces the conservativeness, represented by $\mathbf{P}^{\text{CM,cor}}$, and tries again.

Algorithm 2 Overall Algorithm of the Developed HDe-MPC Algorithm

```
Require: $\boldsymbol{\Lambda}^{\text{CM}}, \boldsymbol{\Phi}^{\text{CM,high}}, \mathbf{P}^{\text{CM}}, \mathbf{f}^{\text{high}}, N_j$ and $\mathbf{f}_j^{\text{slack}}, \mathbf{f}_j, \Phi_j^{\text{CM}}$ for all $j$
Update the Virtual Battery variables and constraint tightening
for $j$=1:$N_j$ do
    L: Calculate $\mathcal{U}_j^{\text{VB}}$ and $\mathbf{P}_j^{\text{cor}}$ and send to the high-level controller
end for

Solve the high-level controller problem
H: Derive $\mathcal{U}^{\text{high}}$ from the $N_j$ individual $\mathcal{U}_j^{\text{VB}}$ sets
H: Calculate $\mathbf{P}^{\text{CM,cont}}{=}\mathbf{P}^{\text{CM}}{-}\mathbf{P}^{\text{CM,cor}}$ using the $N_j$ individual $\mathbf{P}_j^{\text{cor}}$
H: Solve $\mathbf{U}^{\text{high},*}{=}\arg\min_{\mathbf{U}^{\text{high}} \in \mathcal{U}^{\text{high}}, \boldsymbol{\Lambda}^{\text{CM}} \boldsymbol{\Phi}^{\text{CM,high}} \mathbf{U}^{\text{high}} \leq \mathbf{P}^{\text{CM,cont}}} (\mathbf{f}^{\text{high}})^T \mathbf{U}^{\text{high}}$
H: If no feasible solution can be found, reduce $\mathbf{P}^{\text{CM,cor}}$ and try again.
H: Derive $\mathbf{P}_j^{\text{ref}}$ from $\mathbf{U}^{\text{high},*}$ for all $j$ and send to lower-level controllers

Solve the lower-level controller problems
for $j$=1:$N_j$ do
    L: Derive $\mathcal{U}_j^{\text{low,slack}}$ from $\Phi_j^{\text{CM}}$ and $\mathbf{P}_j^{\text{ref}}$
    L: Derive $\mathcal{U}_j^{\text{low,final}} = \bigcap \{\mathcal{U}_j, \; \mathcal{U}_j^{\text{low,slack}}\}$
    L: Solve $\mathbf{U}_j^{\text{low},*} = \arg\min_{[\mathbf{U}_j^T \;\; \boldsymbol{\theta}_j^T]^T \in \mathcal{U}_j^{\text{low,final}}} \mathbf{f}_j^T \mathbf{U}_j {+} (\mathbf{f}_j^{\text{slack}})^T \boldsymbol{\theta}_j$
end for
```

4-6 Numerical Experiment Results and Discussion

The aggregation of EV behavior in VB models results in a loss of information and as a result, a decrease in performance is observed with respect to the Centralized Model Predictive Control (CMPC) solution. By developing tight SoC and power constraints performance was already improved, yet still a mismatch is observed in the numerical experiments. With this mismatch is referred to the difference in actual behavior of the EVs and the behavior as modeled by the aggregated VB. Since this mismatch is a direct consequence of the aggregation, the main challenge is minimize the effect of this mismatch on the system performance. As a result of this mismatch, the numerical experiments showed that the lower-level controllers were not able to fully track the provided reference. To ensure that the global constraints are satisfied in the closed-loop behavior, the penalty vector $\mathbf{f}_j^{\text{slack}}$ was set to have a larger penalty on the first time step than for the rest of the horizon, such that the power reference for the first time step could always be tracked within the allowed maximum error $P_{k,j}^{\text{cor}}$.

MPC Open Loop

Two examples of open loop VB results for the high-level controller can be found in Figure 4-4 and Figure 4-5.

In Figure 4-4, two equally sized subsets are used, of which the results are shown in the two left figure columns, the overall combined system behavior is shown in the most right column of figures. From top to bottom is shown, the SoC, the power, the price signals and the number of EVs present. For each subset, the SoC envelope and power envelope are represented by the minimum and maximum allowable value. The single overall global constraint $\mathbf{P}^{\text{CM}}$, together with the contracted version used in the optimization, here $\mathbf{P}^{\text{cont}}$ for brevity, are shown in the most right column. With respect to the overall power interaction the level of conservativeness is very limited in this example, i.e. 12 kW less on 500 kW total.

It is clear that the minimum power value is contracted in the start of the prediction horizon for both subsets. Therefore, some discharging options have been disabled. In the SoC figures in the top row, the value changes for each time an EV arrives and departs, for example resulting in the steep descent in subset 1 around Program Time Unit (PTU) 65 as one can see in the number of EVs present in the subset. One can also observe that subset 2 clearly is not allowed to charge to its full potential in the first 15 PTUs, this is a result of the global constraint P^{CM}, as one can see in the most right column.

The second example, shown in Figure 4-5 presents the open loop results for the same input data, but then using six subsets and with the same number of EVs distributed among them. In fact, the first three subsets in the second example form the first subset of the first example and the same holds for the rest. It can be observed that the overall system behavior is almost the same. As in the first example, the global constraint does not allow all subsets to fully charge. To be more specific, the first couple of PTUs subset 6 is not allowed to charge and subsequently subsets 1, 5 and 2 are not allowed to fully charge respectively, between PTU 55 and 60.

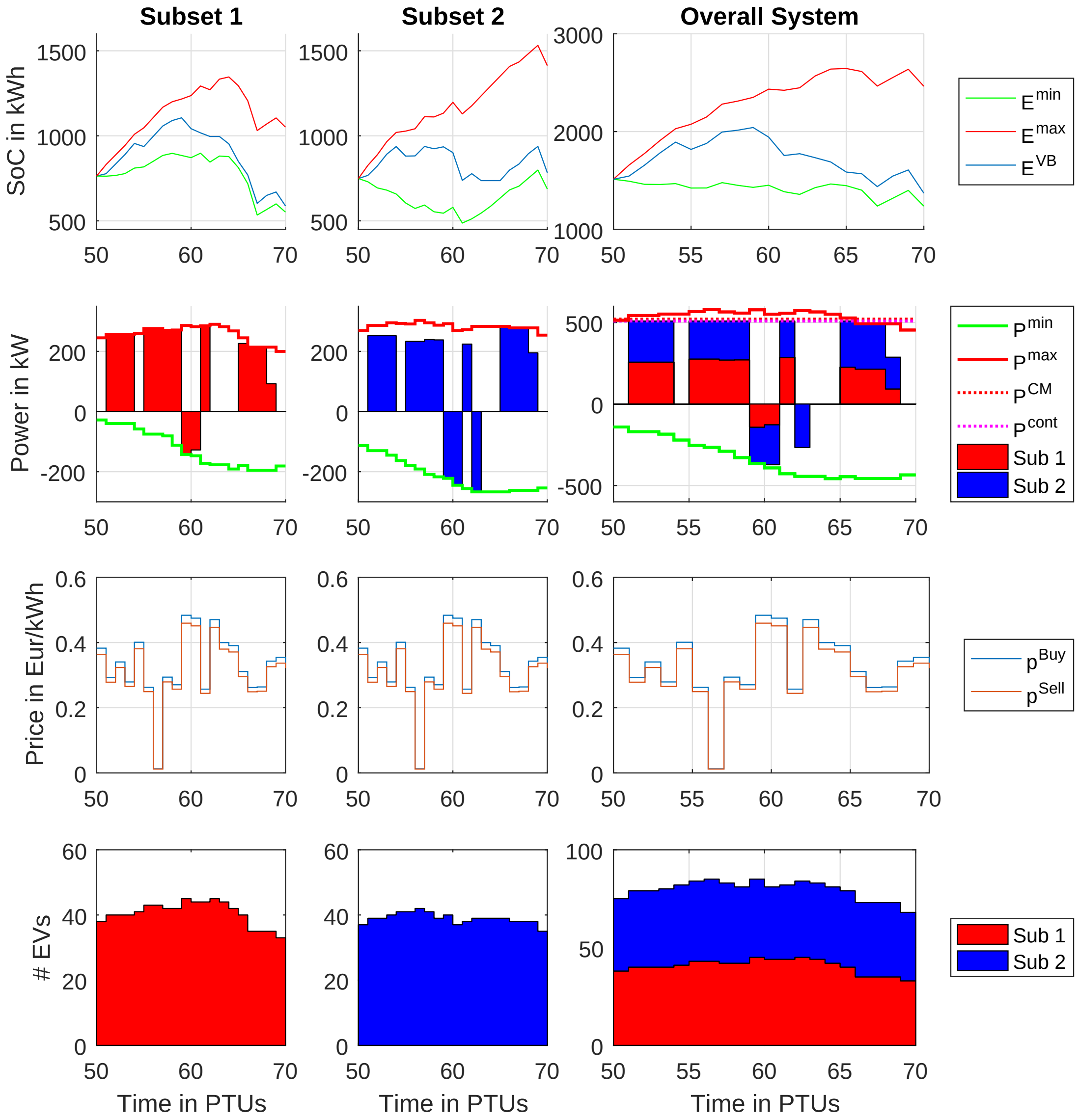


Figure 4-4: Open loop HDe-MPC high-level controller VB predictions for two subsets with 80 EVs in total, combined with overall system behavior

With Program Time Unit (PTU) abbreviated as time steps

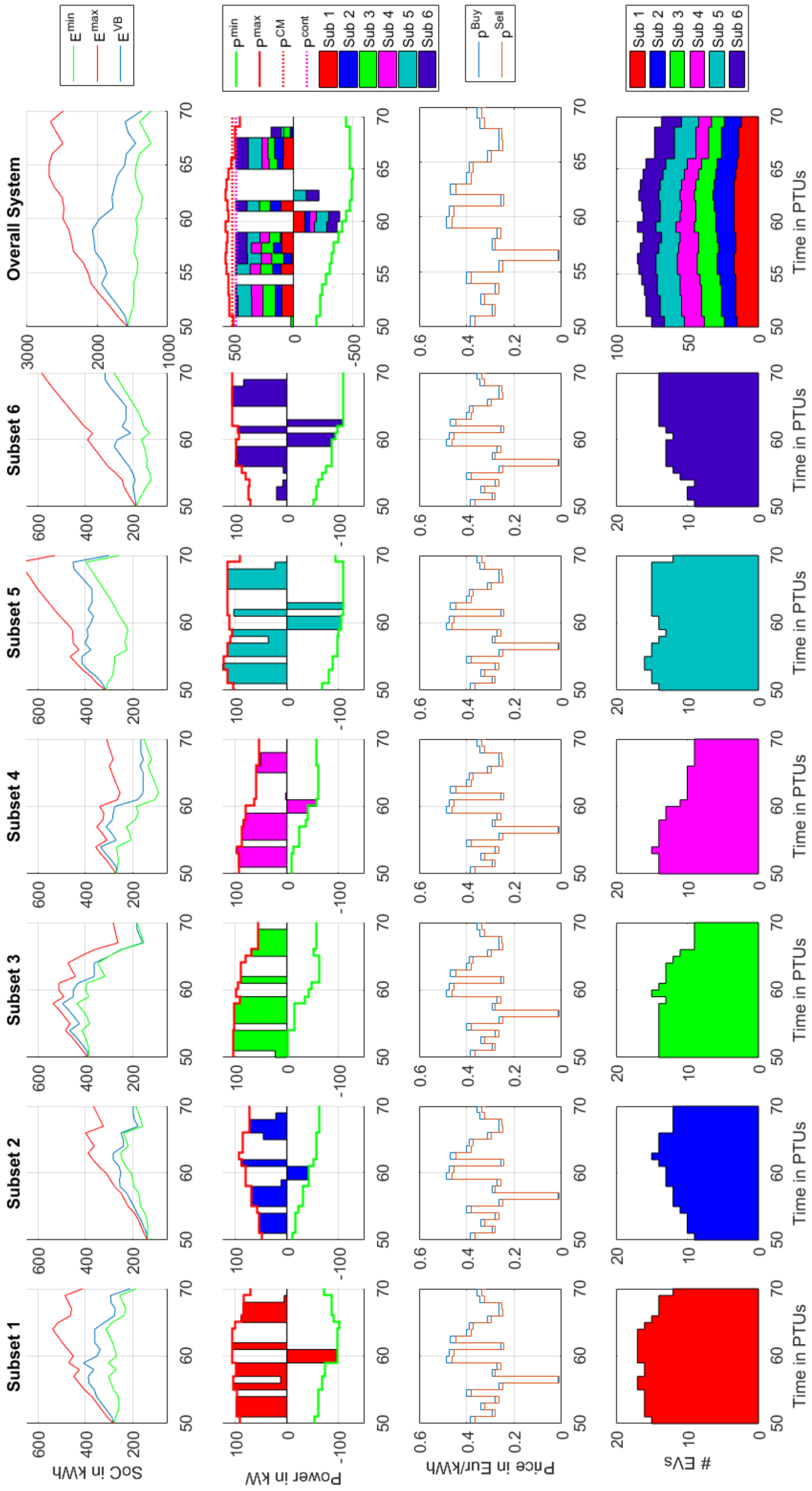


Figure 4-5: Open loop HDe-MPC high-level controller VB predictions for six subsets with 80 EVs in total, combined with overall system behavior

With Program Time Unit (PTU) abbreviated as time steps

4-7 Conclusions

In this chapter, the Virtual Battery (VB) model to represent the aggregated behavior of a subset is introduced and new tight bounds are developed. The developed HDe-MPC algorithm is derived and the control problems for the high-level and lower-level controllers are presented. The following conclusions can be drawn from the numerical experiments:

- The HDe-MPC algorithm is able to come up with feasible solutions consistently. Yet, due to the allowed error in the lower-level controller problems schedules are implemented which cause the higher-level controller to be infeasible due to the contracted resources. By decreasing the conservativeness, feasible solutions can then be found.
- As a result of the aggregated information and reference tracking setup, performance is suboptimal
- The computational cost of the HDe-MPC algorithm scales very well for a growing number of EVs per subset, since the size of the VB model remains the same. As such, most of the computational cost can be found in the lower-level controllers.
- The improved VB model is a lot more complex than the binary input model for individual EVs, yet provides clear insights in the energy flexibility options of an aggregated fleet of EVs.
- By further decreasing the mismatch between the subset behavior and the modeled behavior by a VB, performance is expected to be improved. Options to decrease the mismatch can be to extend the VB model such that more information is incorporated.

Chapter 5

Comparison of Centralized, Distributed and Hierarchical Architectures

In this chapter, the developed algorithms in Distributed Model Predictive Control with Resource Allocation (DMPC-RA) and Hierarchical Decentralized Model Predictive Control (HDe-MPC) architecture will be compared to a Centralized Model Predictive Control (CMPC) benchmark. Numerical experiments will be performed in which the algorithms will be used in closed-loop.

The rest of this chapter is organized as follows. In Section 5-1, the algorithms are briefly described and their general lay-out explained. Section 5-2 presents the performance criteria by which the algorithms which will be compared. The numerical experiment setup will be described in Section 5-3, discussing both the data used in the numerical experiments as the specific tuning parameters for the algorithms. Section 5-4 presents the results of the numerical experiments and compares the algorithms based on the performance criteria. To conclude, Section 5-5 summarizes the main insights drawn from the numerical experiments and comparison.

5-1 Algorithms

The closed loop behavior of the following three MPC algorithms will be compared:

- CMPC with global information. This algorithms is based on a single large controller that has information on all individual Electric Vehicles (EVs) and all constraints. This algorithm should therefore lead to the optimal MPC performance with respect to the optimization problem of the system as defined in (2-18). this provides a performance benchmark to assess the relative performance of the two developed methods. The CMPC solves one large Integer Linear Programming problem (ILP).

- DMPC-RA, with local information and feasible iterations. This algorithm uses an iterative scheme between multiple MPCs and a coordinator in which resources are allocated to the subset MPCs by the coordinator and the MPCs optimize their problems using the available resources. At the end of each iteration the MPCs indicate to what extent they would benefit from extra resources such that the coordinator can calculate an improved resource allocation. This iteration process continues until the resource allocation update converges or when a maximum number of iterations is reached. Each subset MPC solver an ILP.

- HDe-MPC, with local information. This algorithm uses a reference tracking setup in which a high-level controller calculates a power reference signal for the lower-level controllers, which track this reference using an absolute error minimization scheme. The high-level controller only has aggregated information available, for which the lower-level controllers provide an update at the start of each sampling time. Using the aggregated information, the high-level controller models the behavior of each subset using a Virtual Battery (VB) per subset and optimizes the VB behavior. The optimized VB power signal is send to the lower-level controllers, which need to track this reference and optimize the local operational costs. Both the high-level and lower-level controllers solve Mixed Integer Linear Programming problems (MILPs).

5-2 Performance Criteria

The control algorithms will be compared based on the following criteria:

Closed Loop Performance

To asses the performance of the architectures the closed loop performance will be used, which is the cost of the implemented power schedule. The price signals $p^{\mathrm{b}}, p^{\mathrm{s}}$ are measured in *Euro/ kWh* and as a result the performance is then calculated in *Euros*. The close loop performance of CMPC is calculated as,

$$J^{\mathrm{CMPC}} = \sum_{t=1}^{N^{\mathrm{PTU}}} f_{t,k=0}^{T} U_{t,k=0} \tag{5-1}$$

with N^{PTU} as the number of time steps (Program Time Units (PTUs)) in the closed-loop experiments. The closed loop performance of both DMPC-RA and HDe-MPC is calculated as,

$$J^{\mathrm{method}} = \sum_{t=1}^{N^{\mathrm{PTU}}} \sum_{j=1}^{N_j} f_{t,k=0,j}^{T} U_{t,k=0,j}. \tag{5-2}$$

Note that for HDe-MPC, the penalties corresponding to the reference tracking error or not incorporated.

To compare the architectures, the performance of the CMPC is used as a benchmark. Let J^{method} be the cost for a method of interest and J^{CMPC} be the cost for the centralized solution for the same input datasets, then the performance gap of the method with respect to the centralized solution is calculated as,

$$\alpha = \frac{J^{\text{method}} - J^{\text{CMPC}}}{J^{\text{CMPC}}}. \tag{5-3}$$

Parallel Computation Time

The parallel computation time is calculated as the summation of the individual computation times for each sample time t and uses the longest computation time from the subset MPCs,

$$T^{\text{par}} = \sum_{t=1}^{N^{\text{PTU}}} T^{\text{coor}}(t) + \sum_{t=1}^{N^{\text{PTU}}} \max(T_1^{\text{sub}}(t), T_2^{\text{sub}}(t), \ldots, T_{N_j}^{\text{sub}}(t)). \tag{5-4}$$

in which $T^{\text{coor}}(t)$ is the time spent by the coordinator or high-level controller at sample time t and $T_j^{\text{sub}}(t)$ is the time spent by the MPC of subset j at time t. Since the CMPC framework does not have local controllers, $T_j^{\text{sub}}(t)=0$ for all t and j.

5-3 Numerical Experiment Setup

All numerical experiments are performed on a HP Elitebook 8570w with 8GB of RAM and an Intel core-i7 processor with a clock frequency of 2.3 GHz. Gurobi is used as solver with the same settings for all experiments.

All algorithms use a prediction horizon of 5 hours, with the 15 minutes PTU duration, this means $N=20$.

5-3-1 Synthetic Environment Data

The EV parameter data are based on realistic values and drawn from random distributions, as done in [6, 7]. Yet, a distinction is made, such that three types of EVs arise. All EV parameters are individual draws from the distributions. The price data is also drawn from a random distribution and the network constraint $\mathbf{P}^{\text{CM}}$ is calculated based on the drawn EV charging powers, such that it can be assured the network constraint is active.

Electric Vehicle Parameters

The EV parameters are drawn from uniform distributions. Each EV can be one of three types, drawn from a uniform distribution, the three types are: High Power EV, Medium Power EV and Low Power EV. Then, based on the type of EV, the power value is drawn from an integer uniform distribution, with $P_{j,i}^{\text{c}}=P_{j,i}^{\text{d}}$, and the final State of Charge (SoC) is drawn from an integer uniform distribution as $E_{j,i}^{\text{ref}}$, on which the other SoC parameters are based. The charging and discharging efficiencies are drawn independently from each other. The distributions are shown in Table 5-1, with the minimum and maximum value of the distribution shown.

Table 5-1: EV Parameter Distributions

	$P^{\rm c}_{j,i}$=$P^{\rm d}_{j,i}$ (kW)	$E^{\rm ref}_{j,i}$ (kWh)	$\eta^{\rm c}_{j,i}$ (-)	$\eta^{\rm d}_{j,i}$ (-)
High Power EV	[8; 12]	[60; 65]	[0.85; 0.90]	[0.85; 0.90]
Medium Power EV	[5; 8]	[40; 45]	[0.85; 0.90]	[0.85; 0.90]
Low Power EV	[3; 5]	[20; 25]	[0.85; 0.90]	[0.85; 0.90]

Based on $E^{\rm ref}_{j,i}$, the rest of the SoC parameters is determined as,

$$\begin{aligned} E^{\rm max}_{j,i} &= 1.1 E^{\rm ref}_{j,i} \\ E^{\rm min}_{j,i} &= 0.1 E^{\rm ref}_{j,i} \\ E^{\rm arr}_{j,i} &= 0.2 E^{\rm ref}_{j,i} \end{aligned} \tag{5-5}$$

The minimum number of charging PTUs is calculated as,

$$N^{\rm PTU,\ charge} = \text{ceil}\left(\frac{E^{\rm ref}_{j,i} - E^{\rm arr}_{j,i}}{\tau P^{\rm c}_{j,i} \eta^{\rm c}_{j,i}}\right) \tag{5-6}$$

with ceil(.) to round up to the nearest integer. The arrival time $T^{\rm arr}_{j,i}$ is directly drawn from an integer uniform distribution. The departure time $T^{\rm dep}_{j,i}$ is based on the arrival time and draws from an integer distribution the number of extra connected time steps, to add flexibility. To have a fair comparison between the algorithms, all EVs need to be fully charged before the end of the closed loop experiment, as such the maximum value of $T^{\rm dep}_{j,i}$ is it cut off at $N^{\rm PTU}$.

$$\begin{aligned} T^{\rm arr}_{j,i} &= [1; (N^{\rm PTU} - N^{\rm PTU,\ charge})] \\ T^{\rm dep}_{j,i} &= \min(T^{\rm arr}_{j,i} + N^{\rm PTU,\ charge} + [2/\tau; 6/\tau], N^{\rm PTU}) \end{aligned} \tag{5-7}$$

Price Data

The buy price data is drawn from a standard normal distribution, given as $randn$, with an added mean value of $0.3 Eur/kWh$. The sell price is 0.95 times the buy price.

$$\begin{aligned} p^{\rm b}(t) &= 0.3 + .01 randn, && \forall t \\ p^{\rm s}(t) &= 0.95 p^{\rm b}(t), && \forall t \end{aligned} \tag{5-8}$$

Network Data

For the numerical experiments, one single overall constraint is used in which all subsets are jointly constrained, i.e.

$$\Lambda^{\rm CM}_k = [1\ 1\ \dots\ 1] \quad \forall t. \tag{5-9}$$

Depending on the type of experiment, the values for $P^{\rm CM}_k$ will be calculated.

5-4 Numerical Experiments

This section presents the results of two large comparison studies using numerical experiments. In the first study, the number of subsets is kept constant and the number of EVs are changed, to compare the behavior of the algorithms for a growing number of EVs and observe trends. In the second study, the number of EVs is kept constant and the number of subsets is changed, to study the scaling behavior of the algorithms for a growing number of subset MPCs.

5-4-1 Constant number of subsets, changing number of EVs

A constant number of 5 subsets is used, with the total number of EVs ranging from 50 to 250 EVs, with steps of 50 and denoted as N^{EV}. The EVs are equally divided over the subsets and for the DMPC-RA algorithm two versions are tested, with the maximum allowed number of iterations set to 5 and 10. the network constraint P_k^{CM} is provided as a sinusoid with added average and a random disturbance as,

$$\begin{aligned} P_k^{\bar{\text{C}}\text{M}} &= \frac{24}{N^{\text{PTU}}\tau} \sum_{i=1}^{N^{\text{EV}}} \left(\frac{E_{j,i}^{\text{ref}} - E_{j,i}^{\text{arr}}}{\tau (T_{j,i}^{\text{dep}} - T_{j,i}^{\text{arr}})} \right) \\ P_k^{\text{CM}}(t) &= 1.3 P_k^{\bar{\text{C}}\text{M}} + 0.25 P_k^{\bar{\text{C}}\text{M}} \sin\left(\frac{2\pi\tau t}{24}\right) + [-0.25 P_k^{\bar{\text{C}}\text{M}};\ 0.25 P_k^{\bar{\text{C}}\text{M}}], \quad \forall t \end{aligned} \tag{5-10}$$

A total of 10 random draws are performed for each number of EVs, in which no outliers are observed and all algorithms have completed the experiment. An example for this study is shown in Figure 5-1, which shows the CMPC results of the first random draw with 50 EVs. The top plot shows all individual EV SoCs colored by their corresponding subset, the effect of the three EV types can be observed. The second plot shows the aggregated power interaction with the network with the used network constraint. The third plot presents the price signals and the bottom plot shows the number of EVs connected over time, one can observe that some EVs are connected until N^{PTU}, after that time they have met their SoC reference value.

The overall results of the numerical experiments are presented in Figure 5-2. With in Figure 5-2a the parallel computation time. It can clearly be observed how the CMPC time grows quickly for an increasing number of EVs. The computation time of the DMPC-RA algorithm appears to scale in the same way as CMPC, yet due to smaller problem sizes it and limited number of iterations it is terminated earlier. The parallel computation times for HDe-MPC appear to scale linearly and provides the fastest solution for 250 EVs.

To give an indication of the spread in best possible performance, related to the 10 random draws, Figure 5-2b provides the boxplots representing the spread of the CMPC performance. As expected one can see that the costs almost scales linearly with the number of EVs present in the system.

The performance comparison between DMPC-RA and HDe-MPC with the CMPC benchmark is presented in Figure 5-2c with a logarithmic scale on the y-axis. It can be observed the performance gap between the DMPC-RA and the CMPC algorithm diminishes as the number of EVs increase. The DMPC-RA version with 10 iterations does show a limited improved performance with respect to the 5 iteration version, yet the difference is small. The performance

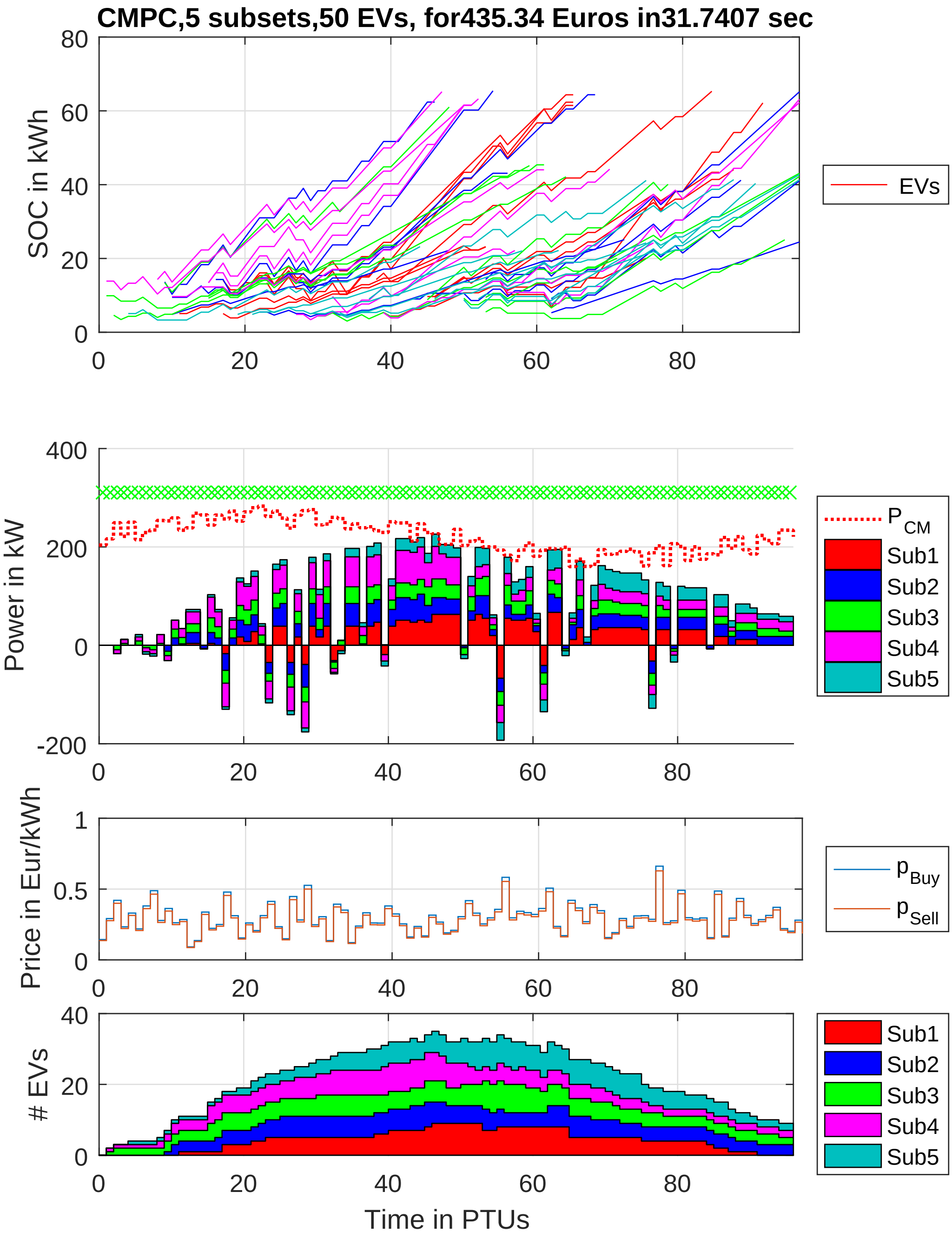


Figure 5-1: CMPC Closed loop numerical experiment result for first random draw with 50 EVs

gap between the HDe-MPC algorithm with the benchmark remains at approximately 10%, this can be explained by the mismatch between the aggregated behavior in the VB model and the actual EV behavior. Due to the aggregation, information is lost.

5-4-2 Constant number of EVs, changing number of subsets

The number of EVs is kept constant in these numerical experiments and they are divided equally over the subsets. For the DMPC-RA algorithm three versions are tested, with the maximum allowed number of iterations set to 5, 10 and 25. The network constraint P_k^{CM} is provided as constant. The available summed maximum charging power over the entire horizon of the experiment is calculated and multiplied by 0.9. By doing so, it can be ensured the global constraint will force power consumption to be shifted at the times when most EVs are connected to the aggregator. Furthermore it still allows the algorithms to find solutions in reasonable computation time. The network constraint P_k^{CM} is calculated as,

$$P_k^{\mathrm{CM}}(t) = 0.9 \max\left(\sum_{i=1}^{N^{\mathrm{EV}}} P_{j,i}^{\mathrm{c}}\right), \quad \forall t \tag{5-11}$$

A total of 5 random draws are performed, in which the data for the EVs remains the same and the EVs are only divided over a different number of subsets. No outliers are observed and all algorithms have completed the experiment. A detailed example of the benchmark CMPC algorithm is shown in Figure 5-3 for the first random draw, with 2 subsets. The top plot represents the aggregated subset power charging and discharging interaction with the distribution grid, the second plot represents the used price signals and the bottom plot shows the number of connected EVs per subset.

The overall results of the experiments are shown in Figure 5-4. The top plot, Figure 5-4a shows the average parallel computation time. The computation time of the CMPC benchmark remains constant as expected, since the only difference is that the EVs are divided over different subsets. Thus it can be clearly observed how the developed algorithms only outperform the benchmark for growing number of subsets, with respect to the computation time. It can also be observed how the computation time almost scales linearly with the number of allowed iterations. Indeed, during the middle of the experiment, when the network constraint is active, the DMPC-RA algorithm is consistently terminated as it has reached the maximum number of iterations, while still improving the performance over iterations. From this observation it can be concluded, that the developed method to determine $\boldsymbol{\psi}_j$ is a suitable method.

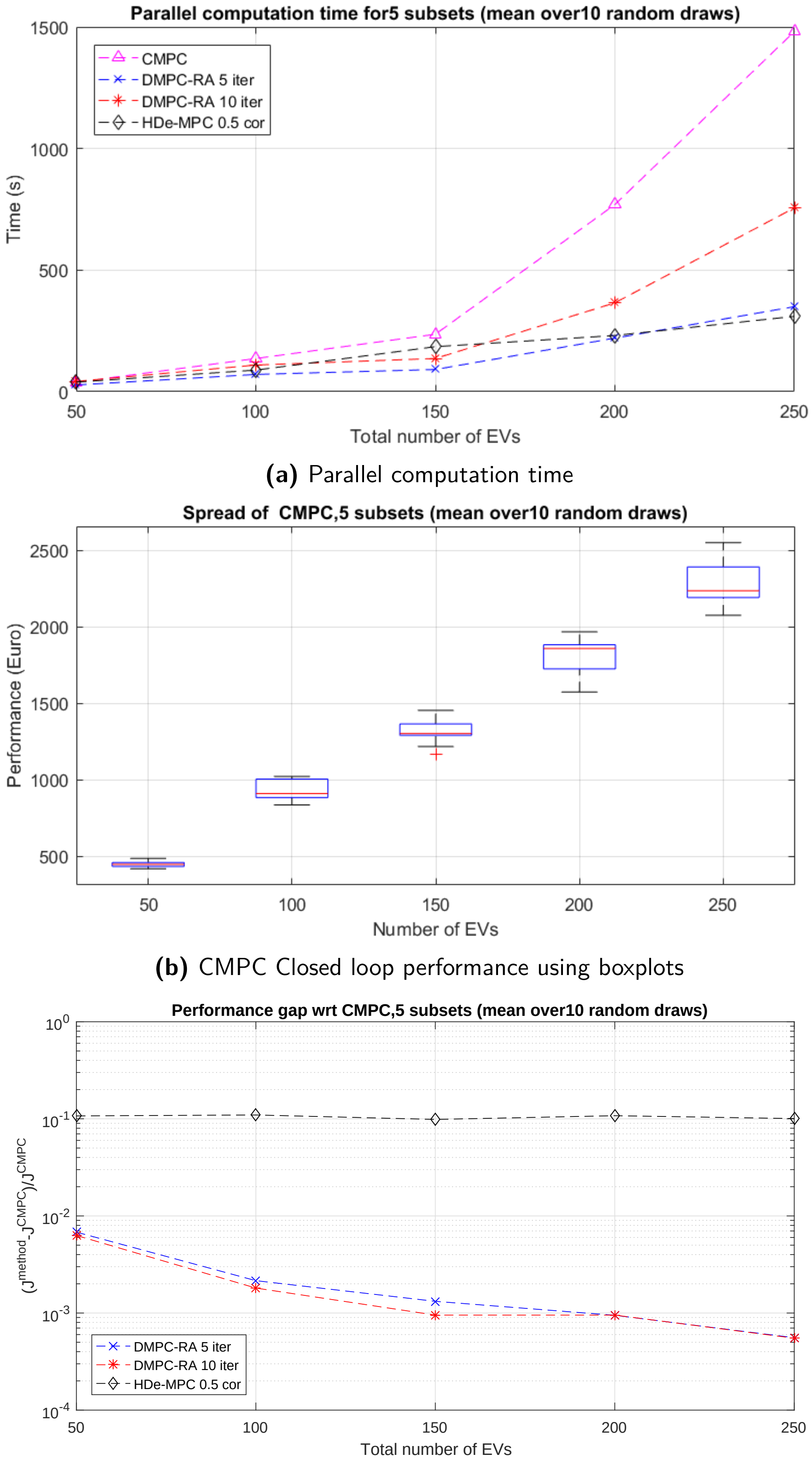
The second plot, Figure 5-4b shows that due to the growing number of subsets, performance is decreased for the developed algorithms, while the CMPC performance remains constant. This is an expected result for the DMPC-RA algorithm, as it has a limited number of iterations and when more subsets need to negotiate, it will take a larger number of iterations until the optimal performance is approached. Yet, the HDe-MPC algorithm also presents a decrease in performance and costs are increased for a larger number of subsets. This is an unexpected result, since the VBs should be able to represent the subset behavior better for smaller subsets and as such it was expected the performance would slightly improve for a growing number of subsets.

In Figure 5-4c, the performance gap with respect to the CMPC result is presented with a logarithmic y-axis. Here it can be observed that the red DMPC-RA line for 10 iterations is not drawn to the data point for 2 subsets. In fact, for this data point, the DMPC-RA had a lower average cost than the CMPC cost. Although somewhat controversial this can be easily explained by noting two facts: 1) the CMPC behavior itself is suboptimal due to the limited prediction horizon and 2) in closed loop the initial conditions for the MPC algorithms over the numerical experiment will not be the same for the CMPC and DMPC-RA algorithms. In open loop the CMPC algorithm will always outperform the DMPC-RA algorithm in this integer setting, causing the CMPC to have different initial conditions in the next time step than the DMPC-RA algorithm. For this data point, this has been beneficial for the DMPC-RA algorithm.

5-5 Conclusions

In this chapter the results of the performed numerical experiments are presented and the algorithms are compared. The following conclusions can be drawn from the numerical experiments:

- All algorithms have shown to be able to come up with feasible power schedules consistently in changing numerical experiment setups.
- The DMPC-RA algorithm shows improving performance for a growing number of iterations, approaching the CMPC benchmark performance. The parallel computation time of DMPC-RA sharply decreases for a growing number of subset MPCs, yet at the cost of performance. A balance might be found by allowing the DMPC-RA algorithm to use more iterations, such that parallel computation time is allowed to increase and performance can be improved.
- The HDe-MPC algorithm shows approximately linear scaling for a growing number of EVs divided over a constant number of subsets, yet the performance gap with respect to the CMPC remains at approximately 10%. Further improvements to decrease the mismatch between the VB model and the actual subset behavior are expected to improve the performance of the HDe-MPC algorithm.



(a) Parallel computation time

(b) CMPC Closed loop performance using boxplots

(c) Closed loop performance with respect to the CMPC performance

Figure 5-2: Closed loop numerical experiment results for 5 subsets, with a growing number of EVs

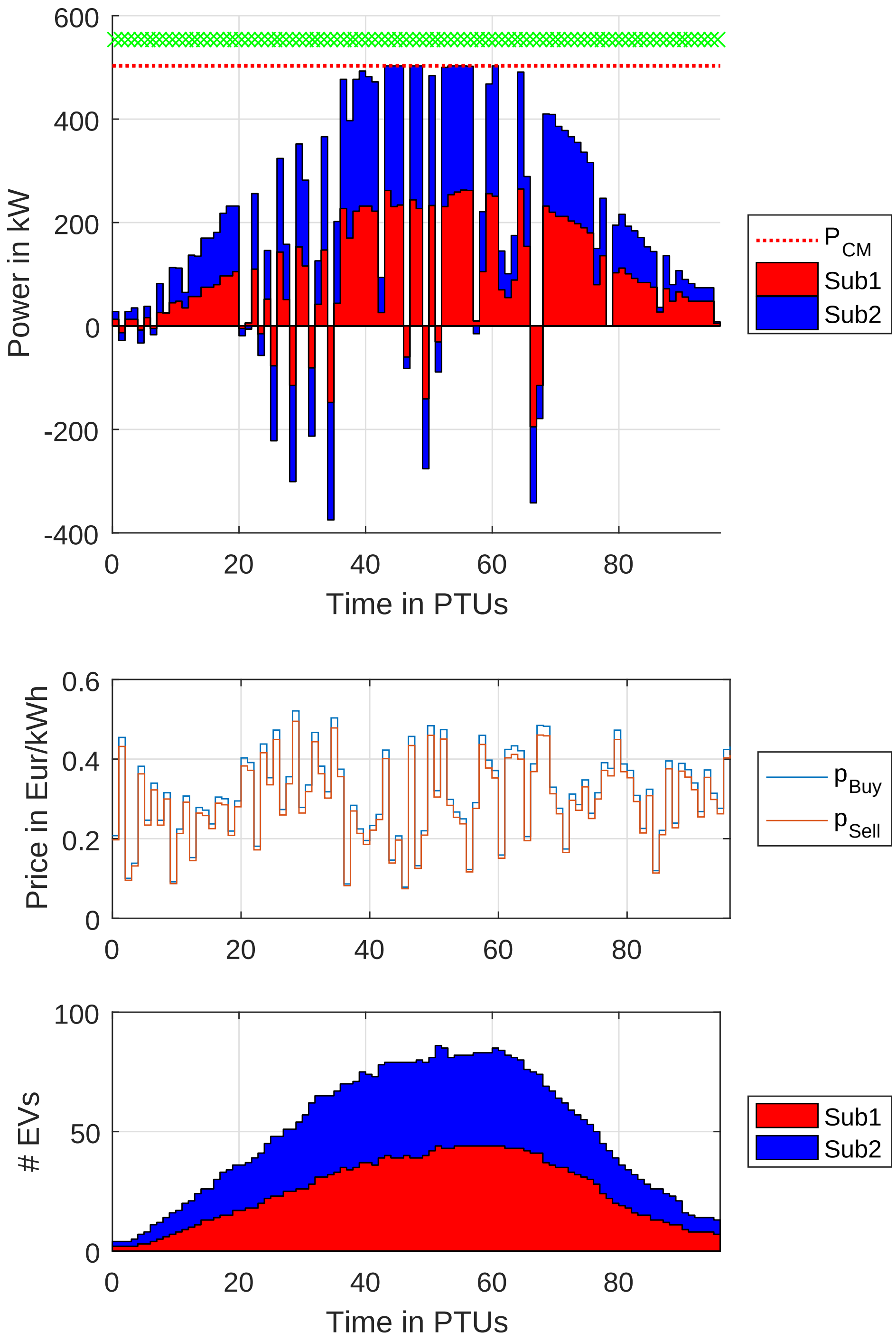
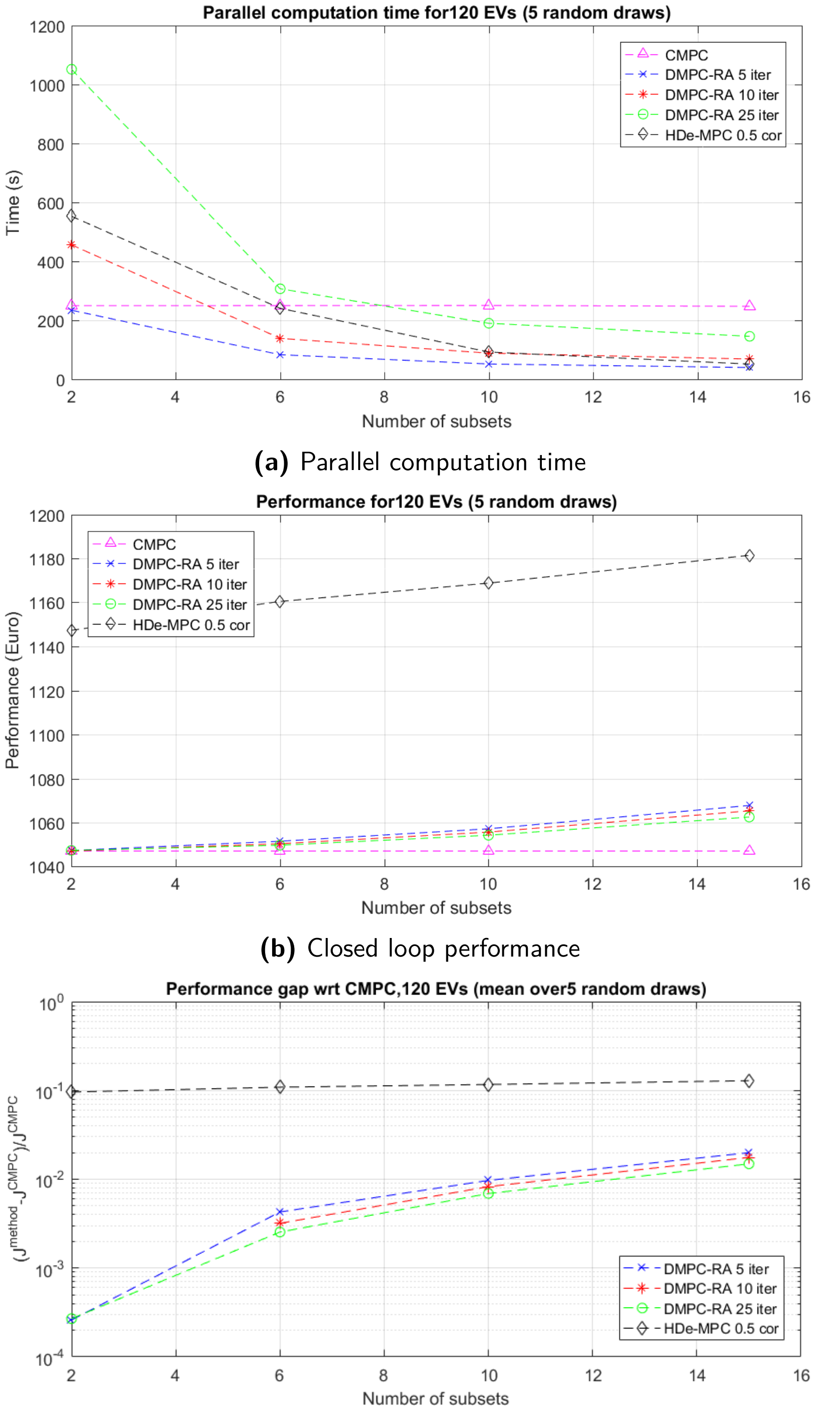


Figure 5-3: CMPC Closed loop numerical experiment result for first random draw with 120 EVs, equally divided over 2 subsets,



(a) Parallel computation time

(b) Closed loop performance

(c) Closed loop performance with respect to the CMPC performance

Figure 5-4: Closed loop numerical experiment results for 120 EVs, equally divided over a growing number of subsets

Chapter 6

Conclusions and Future Work

6-1 Conclusions

Energy flexibility is necessary in an electricity system to match the instantaneous supply and demand. Currently the main providers of energy flexibility can be found on the supply side of the system, yet due to the shift towards renewable energy sources this energy flexibility will phase out. To be able to maintain the balance, the demand needs to be controlled such that is able to provide energy flexibility to the system. Electric Vehicles (EVs) are well suited for this, due to their buffer ability and fast response time. In this thesis the role of EV aggregator is assumed, such that the aggregator is able to directly control the EVs within his portfolio. By incorporating spatial information about the distribution of the EVs, congestion management services can be offered and the options to valorize the energy flexibility of the fleet are increased.

The work in this thesis has aimed at the development of distributed control algorithms for a large fleet of EVs to control the power interaction with the network. Due to the binary charging and discharging behavior of the EVs, a large scale constrained Integer Linear Programming problem (ILP) has to be solved online. To this end, the framework of Model Predictive Control (MPC) has been applied and a Distributed Model Predictive Control with Resource Allocation (DMPC-RA) algorithm and a Hierarchical Decentralized Model Predictive Control (HDe-MPC) algorithm have been developed, to distributed the control challenge and potentially decrease the computational time.

Numerical experiments have shown that all tested algorithms are able to come up with feasible power schedules consistently in changing numerical experiment setups. The two developed algorithms are compared to a Centralized Model Predictive Control (CMPC) benchmark and the main conclusions can be summarized as:

- The DMPC-RA algorithm shows improving performance for a growing number of iterations, approaching the CMPC benchmark performance. The parallel computation time of DMPC-RA sharply decreases for a growing number of subset MPCs, yet at the cost of performance. A balance might be found by allowing the DMPC-RA algorithm to use more iterations, such that parallel computation time is allowed to increase and performance can be improved.

- The HDe-MPC algorithm shows approximately linear scaling for a growing number of EVs divided over a constant number of subsets, yet the performance gap with respect to the CMPC remains at approximately 10%. Further improvements to decrease the mismatch between the Virtual Battery (VB) model and the actual subset behavior are expected to improve the performance of the HDe-MPC algorithm.

6-2 Future Work

In this section, promising directions for future work will be identified.

Possible improvements options for the DMPC-RA architecture can be found in extending the coordinator problem, to enable multiple global constraint handling. Moreover, a more extensive study can be performed to optimize the ψ_j definition for linear cost functions in resource allocation problems with binary input models.

To improve the HDe-MPC architecture, one can try to decrease the mismatch between the aggregated high-level model and actual lower-level behavior. By extending the VB model, performance is expected to be further improved. Furthermore, different communicating strategies between the high-level and lower-level controllers can be investigated, for example a resource allocation scheme. This potentially allows the lower-level controllers to have more responsibility and flexibility options, while the global network constraints are still respected.

Energy flexibility on the demand side is not limited to EVs and aggregators are expected to form a portfolio of appliances with different properties. The challenges in combining these different types of energy flexibility models, with their corresponding constraints, can be an interesting direction to pursue. For the current setting, within the framework of subsets, a study can be performed to incorporate reference tracking and further split the setup such that energy flexibility options can be valorized even further to provide imbalance management services for Balance Responsible Parties (BRPs) as well.

One of the main assumptions in this thesis, is that all systems are completely deterministic and perfect forecasts are available for all variables and signals. Yet, in practice this assumption does not hold and the reliability of the forecasts is limited. Including stochasticity for the price signals, network constraints and EV data will enable interesting future research directions.

With respect to a long-term perspective, it is interesting to do more fundamental work towards finding distributed control methods for hybrid systems with integer variables, that can be guaranteed to converge to the optimal global performance in finite time. Many real life applications are best modeled by hybrid systems or binary input models, such as traffic and electricity networks.

Appendix A

Detailed Variable Definitions

This appendix provides the definitions of the compact vectors and matrices. For brevity, all derivations are given for a single sample time t. The horizon counter k will be used as subscript in the derivation, unless stated otherwise.

A-1 CMPC Definitions

This section will present the detailed definitions such that the Centralized Model Predictive Control (CMPC) algorithm can be compactly written as,

$$\begin{aligned} \underset{\mathbf{U}}{\text{minimize}} \quad & \mathbf{f}^T\mathbf{U} \\ \text{s.t.} \quad & \mathbf{U}{\in}\mathcal{U} \\ & \mathbf{\Lambda}^{\text{CM}}\mathbf{\Phi}^{\text{CM}}\mathbf{U}{\leq}\mathbf{P}^{\text{CM}}. \end{aligned} \tag{A-1}$$

Input Vector

All local Electric Vehicle (EV) input vectors for k are first joined per subset j, then per step k, and then over the entire horizon as,

$$\mathbf{U}{=}\begin{bmatrix} U_{k=0} \\ U_{k=1} \\ \vdots \\ U_{k=N-1} \end{bmatrix}, U_k{=}\begin{bmatrix} U_{k,j=1} \\ U_{k,j=2} \\ \vdots \\ U_{k,j=N_j} \end{bmatrix}, U_{k,j}{=}\begin{bmatrix} \vdots \\ u_{j,i}(k) \\ \vdots \end{bmatrix} \quad \forall i{\in}\mathcal{I}_j(k) \tag{A-2}$$

in which $U_{k,j}$ contains all $u_{j,i}(k)$ vectors for all EVs in $\mathcal{I}_j(k)$, as such, the size of the input vectors changes over time as EVs arrive and depart.

Cost Vector

The cost vectors of all individual EVs are combined, such that the cost function of the CMPC problem can be represented as,

$$\text{minimize} \quad \mathbf{f}^T\mathbf{U} \tag{A-3}$$

To this end, the individual EV cost vectors for k are first joined per subset, then per step k and then over the entire horizon as,

$$\mathbf{f}{=}\begin{bmatrix} f_{k=0} \\ f_{k=1} \\ \vdots \\ f_{k=N-1} \end{bmatrix}, f_k{=}\begin{bmatrix} f_{k,j=1} \\ f_{k,j=2} \\ \vdots \\ f_{k,j=N_j} \end{bmatrix}, f_{k,j}{=}\begin{bmatrix} \vdots \\ f_{j,i}(k) \\ \vdots \end{bmatrix} \quad \forall i{\in}\mathcal{I}_j(k) \tag{A-4}$$

Local Constraints

When all local constraints are joined and rewritten using substitution, they can be written as constraints on the input vector as,

$$\mathbf{U} \in \mathcal{U} \tag{A-5}$$

where $\mathcal{U}$ is formed by,

$$\mathcal{U} = \{\mathbf{U} \mid G_0\mathbf{U}{\leq}w_0{+}E_0\mathbf{x}(0) \text{ and } \mathbf{U}{\in}\{0,1\}^{2N^{\text{EV inst}}}\} \tag{A-6}$$

with the matrices G_0, E_0 and vector w_0 defined subsequently, $\mathbf{x}(0)$ as the vector containing the initial State of Charges (SoCs) of the EVs present within the prediction horizon and $N^{\text{EV inst}}$ as the total number of EVs instances present over the horizon, defined using the cardinality of the sets $\mathcal{I}_j(k)$ as,

$$N^{\text{EV inst}} = \sum_{k\in\mathcal{N}}\sum_{j=1}^{N_j}|\mathcal{I}_j(k)|. \tag{A-7}$$

The vector $\mathbf{x}$ contains all SoCs of all EVs present over the horizon and is defined in the same way as $\mathbf{U}$. The constraints on $\mathbf{x}$ will be rewritten as constraints on $\mathbf{U}$, using substitution of the dynamics of the system, which were defined as,

$$E_{j,i}(k{+}1) = E_{j,i}(k){+}B_{j,i}u_{j,i}(k) \quad \forall k{\in}\mathcal{N}, \forall j, \forall i{\in}\mathcal{I}_j(k) \tag{A-8}$$

which can be joined for all EVs and subsets for each step in the horizon as,

$$x_{k+1} = A_kx_k{+}B_kU_k \quad \forall k{\in}\mathcal{N}. \tag{A-9}$$

Note that due to the arrival and departure of EVs, these vectors and matrices may change in size and as such are dependent on time. Then let the remaining local constraints,

$$\begin{aligned} &\begin{bmatrix} 1 & 1 \end{bmatrix} u_{j,i}(k) \leq 1, && \forall k{\in}\mathcal{N}, \forall j, \forall i{\in}\mathcal{I}_j(k) \\ &E_{j,i}^{\text{MPC,min}}(k{+}1) \leq E_{j,i}(k{+}1) \leq E_{j,i}^{\text{MPC,max}}(k{+}1) && \forall k{\in}\mathcal{N}, \forall j, \forall i{\in}\mathcal{I}_j(k) \end{aligned} \tag{A-10}$$

be rewritten such that all EVs and subsets for each step in the horizon are joined as,

$$\begin{aligned} A^u_k U_k \leq b^u_k \qquad & \forall k \in \mathcal{N} \\ A^x_{k+1} x_{k+1} \leq b^x_{k+1} \quad & \forall k \in \mathcal{N}. \end{aligned} \tag{A-11}$$

In fact, for the construction of E_0 it is beneficial to write the state constraints as,

$$-S_{k+1} A^{x,matrix} x_{k+1} \leq b^x_{k+1} \quad \forall k \in \mathcal{N}, \tag{A-12}$$

with $S_{k+1} \in \{0,1\}^{2N^{\text{EV}}_k \times 2N^{\text{EV hor}}}$ indicating which EVs are connected at time step $k+1$ within the prediction horizon. The matrix S_{k+1} has an identity matrix of size 2 whenever an EV is present in the corresponding step. Here, $N^{\text{EV hor}}$ is the number of EVs present in the entire prediction horizon and N^{EV}_k is the number of EVs present in time step $k+1$, defined using the cardinality of the set $\mathcal{I}_j(k)$ as,

$$N^{\text{EV}}_k = \sum_{j=1}^{N_j} |\mathcal{I}_j(k)|. \tag{A-13}$$

And since all individual EVs have a minimum and maximum state constraint and state-space matrix of A=1, their A^x=$[-1 \quad 1]^T$ are identical and the matrix $A^{x,matrix}$ is defined as,

$$A^{x,matrix} = \begin{bmatrix} A^x & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & A^x & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & A^x \end{bmatrix} \in \{0,1\}^{2N^{\text{EV hor}} \times N^{\text{EV hor}}}, \tag{A-14}$$

with $\mathbf{0}$ as the matrix containing all zeros of appropriate dimensions. The matrix E_0 and the vector w_0 can be defined as,

$$E_0 = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \\ -S_1 A^{x,matrix} \\ -S_2 A^{x,matrix} \\ \vdots \\ -S_N A^{x,matrix} \end{bmatrix}, w_0 = \begin{bmatrix} b^u_{k=0} \\ b^u_{k=1} \\ \vdots \\ b^u_{k=N-1} \\ b^x_{k=1} \\ b^x_{k=2} \\ \vdots \\ b^x_{k=N} \end{bmatrix}. \tag{A-15}$$

the matrix G_0 is stacked accordingly.

Global Constraints

The global constraints can be written as:

$$\mathbf{\Lambda}^{\text{CM}} \mathbf{\Phi}^{\text{CM}} \mathbf{U} \leq \mathbf{P}^{\text{CM}} \tag{A-16}$$

The $\mathbf{\Lambda}^{\text{CM}}$ matrix provides the coupling between the subsets, the $\mathbf{\Phi}^{\text{CM}}$ matrix provides the power values of the EVs represented in $\mathbf{U}$, and the $\mathbf{P}^{\text{CM}}$ vector provides the power values by which the combination of subsets are constrained. The $\mathbf{\Lambda}^{\mathrm{CM}}$ matrix is defined by diagonally stacking the coupling matrices for the individual steps in the horizon,

$$\mathbf{\Lambda}^{\mathrm{CM}}=\begin{bmatrix} \Lambda^{\mathrm{CM}}_{k=0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \Lambda^{\mathrm{CM}}_{k=1} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \Lambda^{\mathrm{CM}}_{k=N-1} \end{bmatrix}, \tag{A-17}$$

with $\Lambda^{\mathrm{CM}}_k \in \{0,1\}^{N^{\mathrm{CM}}_k \times N_j}$ as the matrix which represents the combination of subsets which are jointly constrained and N^{CM}_k as the number of global constraints for k. Then $\mathbf{\Phi}^{\mathrm{CM}}$ and Φ^{CM}_k are defined as,

$$\mathbf{\Phi}^{\mathrm{CM}}=\begin{bmatrix} \Phi^{\mathrm{CM}}_{k=0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \Phi^{\mathrm{CM}}_{k=1} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \Phi^{\mathrm{CM}}_{k=N-1} \end{bmatrix}, \Phi^{\mathrm{CM}}_k=\begin{bmatrix} \Phi^{\mathrm{CM}}_{k,j=1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \Phi^{\mathrm{CM}}_{k,j=2} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \Phi^{\mathrm{CM}}_{k,j=N_j} \end{bmatrix} \tag{A-18}$$

with the row vector $\Phi^{\mathrm{CM}}_{k,j}$ defined as,

$$\Phi^{\mathrm{CM}}_{k,j}=\begin{bmatrix} \dots & P^{\mathrm{c}}_{j,i} & P^{\mathrm{d}}_{j,i} & \dots \end{bmatrix} \quad \forall i \in \mathcal{I}_j(k). \tag{A-19}$$

The $\mathbf{P}^{\mathrm{CM}}$ vector is defined as,

$$\mathbf{P}^{\mathrm{CM}}=\begin{bmatrix} P^{\mathrm{CM}}_{k=0} \\ P^{\mathrm{CM}}_{k=1} \\ \vdots \\ P^{\mathrm{CM}}_{k=N-1} \end{bmatrix} \tag{A-20}$$

in which the column vector $P^{\mathrm{CM}}_k \in \mathbb{R}^{N^{\mathrm{CM}}_k}$ holds the power values by which the subsets are constrained.

A-2 DMPC-RA Definitions

The rationale used for the centralized definitions holds as well for the subset DMPC-RA definitions. Yet, variables are not joined over all subsets, but by first joining the EV variables per subset and then directly over the horizon. This section will present the definitions as are used in Chapter 3 such that the subset Model Predictive Control (MPC) problem can be compactly written as,

$$\begin{aligned} \text{minimize} \quad & \mathbf{f}_j^T \mathbf{U}_j \\ \text{s.t.} \quad & \mathbf{U}_j \in \mathcal{U}_j \\ & \Phi^{\mathrm{CM}}_j \mathbf{U}_j \leq \gamma_j. \end{aligned} \tag{A-21}$$

Input Vector and Cost Vector

The subset MPC input vector $\mathbf{U}_j$ is defined per subset by directly joining $U_{k,j}$ over the horizon. The same holds for the subset MPC cost vector $\mathbf{f}_j$,

$$\mathbf{U}_j = \begin{bmatrix} U_{k=0,j} \\ U_{k=1,j} \\ \vdots \\ U_{k=N-1,j} \end{bmatrix}, \mathbf{f}_j = \begin{bmatrix} f_{k=0,j} \\ f_{k=1,j} \\ \vdots \\ f_{k=N-1,j} \end{bmatrix} \tag{A-22}$$

with $U_{k,j}$ and $f_{k,j}$ as previously defined.

Local Constraints

When all local constraints are joined and rewritten using substitution, they can be written as constraints on the subset input vector as,

$$\mathbf{U}_j \in \mathcal{U}_j, \tag{A-23}$$

where $\mathcal{U}_j$ is formed by,

$$\mathcal{U}_j = \{\mathbf{U}_j \mid G_0\mathbf{U}_j \leq w_0 + E_0\mathbf{x}_j(0) \text{ and } \mathbf{U}_j \in \{0,1\}^{2N_j^{\text{EV inst}}}\} \tag{A-24}$$

with $N_j^{\text{EV inst}}$ as the total number of EVs instances present over the horizon within subset j, defined using the cardinality of $\mathcal{I}_j(k)$ as,

$$N_j^{\text{EV inst}} = \sum_{k \in \mathcal{N}} |\mathcal{I}_j(k)| \tag{A-25}$$

and the matrices G_0, E_0 and vector w_0 are defined in the same way as for the centralized problem, but then only combined per subset j. For brevity the derivation is omitted.

Decoupled Global Constraints

To obtain the decoupled global constraint formulation,

$$\Phi_j^{\text{CM}}\mathbf{U}_j \leq \boldsymbol{\gamma}_j, \tag{A-26}$$

the matrix Φ_j^{CM} is defined by diagonally stacking the row vectors $\Phi_{k,j}^{\text{CM}}$ over the horizon, done per subset j,

$$\Phi_j^{\text{CM}} = \begin{bmatrix} \Phi_{k=0,j}^{\text{CM}} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \Phi_{k=1,j}^{\text{CM}} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \Phi_{k=N-1,j}^{\text{CM}} \end{bmatrix}, \tag{A-27}$$

with $\Phi_{k,j}^{\text{CM}}$ and $\boldsymbol{\gamma}_j$ as previously defined.

A-3 HDe-MPC Virtual Battery Definitions

The rationale used for writing the local EV constraints in set notation for the centralized definitions holds as well for the local Virtual Battery (VB) definitions for the HDe-MPC algorithm.. This section will present the VB set definition as used in Chapter 4, such that the local constraints for VB j can be written as,

$$U_j^{\mathrm{VB}} \in \mathcal{U}_j^{\mathrm{VB}} \tag{A-28}$$

If all VB constraints are joined over the horizon and rewritten using substitution, they can be written as constraints on U_j^{VB} for which $\mathcal{U}_j^{\mathrm{VB}}$ is formed by,

$$\mathcal{U}_j^{\mathrm{VB}} = \{U_j^{\mathrm{VB}} \mid G_0 U_j^{\mathrm{VB}} \leq w_0 + E_0 E_j^{\mathrm{VB}}(0)\} \tag{A-29}$$

with the matrices G_0, E_0 and vector w_0 defined in the same way as for the centralized problem, but then only joined over the prediction horizon. The scalar $E_j^{\mathrm{VB}}(0)$ represents the initial SoC for the VB. The constraints on $E_j^{\mathrm{VB}}(k+1)$ will be rewritten as constraints on U_j^{VB}, using substitution of the dynamics of the system, which were defined as,

$$E_j^{\mathrm{VB}}(k+1) = E_j^{\mathrm{VB}}(k) + B_{k,j}^{\mathrm{VB}} U_{k,j}^{\mathrm{VB}} + w_{k,j}^{\mathrm{VB}}, \quad \forall k \in \mathcal{N}, \tag{A-30}$$

which can be rewritten as,

$$x_{k+1,j}^{\mathrm{VB}} = A_{k,j} x_{k,j}^{\mathrm{VB}} + U_{k,j}^{\mathrm{VB}} U_{k,j}^{\mathrm{VB}} + w_{k,j}^{\mathrm{VB}}, \quad \forall k \in \mathcal{N}. \tag{A-31}$$

Note that due to the arrival and departure of EVs, these vectors and matrices are dependent on time. Then let the local constraints, be rewritten such that they are joined as,

$$\begin{aligned} A_{k,j}^{u} U_{k,j}^{\mathrm{VB}} &\leq b_{k,j}^{u} & \forall k \in \mathcal{N} \\ A_{k+1,j}^{x} x_{k+1,j}^{\mathrm{VB}} &\leq b_{k+1,j}^{x} & \forall k \in \mathcal{N}. \end{aligned} \tag{A-32}$$

For brevity the further derivation is omitted.

A-4 HDe-MPC High-Level Controller Definitions

The rationale used for the centralized definitions holds as well for the high-level controller definitions for the HDe-MPC algorithm, yet using the VB model instead of the binary input models. This section will present the definitions as are used in Chapter 4, such that the high-level controller problem can be compactly written as,

$$\begin{aligned} \underset{\mathbf{U}^{\mathrm{high}}}{\text{minimize}} \quad & (\mathbf{f}^{\mathrm{high}})^T \mathbf{U}^{\mathrm{high}} \\ \text{s.t.} \quad & \mathbf{U}^{\mathrm{high}} \in \mathcal{U}^{\mathrm{high}} \\ & \mathbf{\Lambda}^{\mathrm{CM}} \mathbf{\Phi}^{\mathrm{CM,high}} \mathbf{U}^{\mathrm{high}} \leq \mathbf{P}^{\mathrm{CM,cont}} \end{aligned} \tag{A-33}$$

Cost Vector

The cost vector $\mathbf{f}^{\text{high}}$ is defined by joining the individual VB cost vectors $f_{k,j}^{\text{high}}$ as,

$$\mathbf{f}^{\text{high}}=\begin{bmatrix} f_{k=0}^{\text{high}} \\ f_{k=1}^{\text{high}} \\ \vdots \\ f_{k=N-1}^{\text{high}} \end{bmatrix}, f_k^{\text{high}}=\begin{bmatrix} f_{k,j=1}^{\text{high}} \\ f_{k,j=2}^{\text{high}} \\ \vdots \\ f_{k,j=N_j}^{\text{high}} \end{bmatrix}. \tag{A-34}$$

Virtual Battery Constraints

The lower-level controllers provide the high-level controller with the sets $\mathcal{U}_j^{\text{VB}}$ for all j, such that $U_j^{\text{VB}}\in\mathcal{U}_j^{\text{VB}}$. If joined for all VBs, this is represented as,

$$\mathbf{U}^{\text{high}}\in\mathcal{U}^{\text{high}} \tag{A-35}$$

Global Constraints

To represent the global constraints as,

$$\mathbf{\Lambda}^{\text{CM}}\mathbf{\Phi}^{\text{CM,high}}\mathbf{U}^{\text{high}}\leq\mathbf{P}^{\text{CM,cont}} \tag{A-36}$$

the matrix $\mathbf{\Phi}^{\text{CM,high}}$ is defined as,

$$\mathbf{\Phi}^{\text{CM,high}}=\begin{bmatrix} \Phi_k^{\text{CM,high}} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \Phi_k^{\text{CM,high}} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \Phi_k^{\text{CM,high}} \end{bmatrix}, \tag{A-37}$$

with the constant $\Phi_k^{\text{CM,high}}$ as,

$$\Phi_k^{\text{CM,high}}=\begin{bmatrix} \Phi_{k,j}^{\text{CM,high}} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \Phi_{k,j}^{\text{CM,high}} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \Phi_{k,j}^{\text{CM,high}} \end{bmatrix}\in\{0,1\}^{N_j\times 3N_j}, \tag{A-38}$$

with $\Phi_{k,j}^{\text{CM,high}}=[1 \quad 0 \quad 0]$ since only $P_j^{\text{VB}}(k)$ is coupled.

Bibliography

[1] International Energy Agency (IEA), "Global EV Outlook 2016," 2016.

[2] M. D. Galus, M. G. Vayá, T. Krause, and G. Andersson, "The role of electric vehicles in smart grids," *Wiley Interdisciplinary Reviews: Energy and Environment*, vol. 2, no. 4, pp. 384–400, 2013.

[3] Universal Smart Energy Framework (USEF), "USEF: The Framework Explained," 2015.

[4] G. Papaefthymiou, K. Grave, and K. Dragoon, "Flexibility options in electricity systems," *Ecofys, European Copper Institute*, no. March, p. 51, 2014.

[5] R. Vujanic, P. M. Esfahani, P. Goulart, and M. Morari, "Electric Vehicles Aggregator Optimization : a Fast and Solver-Free Solution Method," in *53rd IEEE Conference on Decision and Control*, pp. 5027–5032, 2014.

[6] R. Vujanic, P. M. Esfahani, P. Goulart, S. Mariéthoz, and M. Morari, "A decomposition method for large scale MILPs, with performance guarantees and a power system application," *Automatica*, vol. 67, pp. 144–156, 2016.

[7] R. Luo, R. Bourdais, T. J. van den Boom, and B. De Schutter, "Multi-agent model predictive control based on resource allocation coordination for a class of hybrid systems with limited information sharing," *Engineering Applications of Artificial Intelligence*, vol. 58, no. September 2016, pp. 123–133, 2017.

[8] R. Luo, *Multi-Agent Control of Urban Transportation Networks and of Hybrid Systems with Limited Information Sharing.* PhD thesis, Delft University of Technology, 2016.

[9] F. Ruelens, S. Vandael, W. Leterme, B. J. Claessens, M. P. F. Hommelberg, T. Holvoet, and R. Belmans, "Demand side management of electric vehicles with uncertainty on arrival and departure times," *IEEE PES Innovative Smart Grid Technologies Conference Europe*, pp. 1–8, 2012.

[10] S. Vandael, B. Claessens, M. P. F. Hommelberg, T. Holvoet, and G. Deconinck, “A scalable three-step approach for demand side management of plug-in hybrid vehicles,” *IEEE Transactions on Smart Grid*, vol. 4, no. 2, pp. 720–728, 2013.

[11] K. De Craemer, S. Vandael, B. Claessens, and G. Deconinck, “An event-driven dual coordination mechanism for demand side management of PHEVs,” *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 751–760, 2014.

[12] D. P. Bertsekas, *Nonlinear programming.* Athena scientific Belmont, 1999.

[13] A. Bemporad and M. Morari, “Control of systems integrating logic, dynamics, and constraints,” *Automatica*, vol. 35, no. 3, pp. 407–427, 1999.

[14] A. Parisio, E. Rikos, and L. Glielmo, “A model predictive control approach to microgrid operation optimization,” *IEEE Transactions on Control Systems Technology*, vol. 22, no. 5, pp. 1813–1827, 2014.

Glossary

List of Acronyms

BRP	Balance Responsible Party
CMPC	Centralized Model Predictive Control
DMPC-RA	Distributed Model Predictive Control with Resource Allocation
DSO	Distribution System Operator
EV	Electric Vehicle
HDe-MPC	Hierarchical Decentralized Model Predictive Control
ILP	Integer Linear Programming problem
MILP	Mixed Integer Linear Programming problem
MLD	Mixed Logical Dynamical system
MPC	Model Predictive Control
SoC	State of Charge
PTU	Program Time Unit
TSO	Transmission System Operator
V2G	Vehicle to Grid
VB	Virtual Battery