

**UNIVERSITÄT
BAYREUTH**

**ERWEITERUNGEN EINES ANSATZES ZUR VERTEILTEN
PRÄDIKTIVEN REGELUNG VON SMART-GRIDS**

MASTERARBEIT

VON

MARKUS KLAR
DEZEMBER 2018

LEHRSTUHL FÜR ANGEWANDTE MATHEMATIK

UNIVERSITÄT BAYREUTH

Erstprüfer: Prof. Dr. Lars Grüne
Zweitprüfer: Prof. Dr. Kurt Chudej

Vorgelegt am 10.12.2018

Bayreuth

Abstract

Ein Smart Grid besteht aus mehreren Haushalten, die gleichzeitig Stromerzeuger und -verbraucher sind. Für den Netzbetreiber entsteht die Herausforderung, Spannungsschwankungen im Stromnetz, welche durch die über den Tag variierenden Verbrauchs- und Erzeugungsmengen entstehen, auszugleichen. In dieser Masterarbeit wird ein Ansatz zur verteilten modellprädiktiven Regelung von Smart Grids durch die Steuerung von Haushaltsbatterien von Dr. Braun [1] aufgegriffen und erweitert. Wir untersuchen den *Distributed MPC* im Bezug auf Verbrauchsprognosen und werden zeigen, dass aggregierte Prognosen für die Optimierung ausreichend sind. Außerdem erweitern wir das Modell um den Einsatz von Elektroautos und erzeugen Simulationen verschiedener Szenarien. Dabei implizieren die Ergebnisse, dass die optimale Steuerung des Systems viel dazu beitragen kann, die Überlastung unseres Stromnetzes durch Elektromobilität zu verhindern.

Schlüsselwörter: Modellprädiktive Regelung (MPC), verteilte Optimierung, Smart Grid, Verbrauchsprognosen, Elektromobilität, Haushaltsbatterien

Inhaltsverzeichnis

Abstract	i
1 Einführung	1
1.1 Die Energiewende - Chancen und Herausforderungen	1
1.2 Ziel dieser Masterarbeit	3
2 Mathematische Grundlagen	5
2.1 Mathematische Kontrolltheorie	5
2.2 Nichtlineare Modellprädiktive Regelung (NMPC)	10
3 Modellierung und optimale Steuerung eines Smart Grids	17
3.1 Modellbezogene Grundlagen	17
3.2 Centralized Model Predictive Control (CMPC)	22
3.3 Distributed Model Predictive Control (DMPC)	22
4 Verbrauchsschätzungen	29
4.1 Problematik und Auswege	29
4.2 Einsatz gemittelter Verbrauchsschätzungen	30
4.3 Numerische Resultate	33
5 Das Elektroauto im Smart Grid	41
5.1 Erweiterung des Modells	41
5.2 DMPC mit Elektroautos	44
6 Numerische Untersuchungen des DMPC mit Elektroautos	47
6.1 Grundmodell	47
6.2 Die Turnpike Eigenschaft	48
6.3 Numerische Simulation des DMPC mit Elektroautos	50
7 Fazit und Ausblick	59
7.1 MPC in Smart Grids als Chance für die Energiewende	59
7.2 Ausblick auf künftige Arbeiten	60
Literaturverzeichnis	61
Anhang	64
A Implementierung des DMPC in Matlab	64

Einführung

1.1 Die Energiewende - Chancen und Herausforderungen

Unsere Welt befindet sich im Wandel und das ist überall zu spüren: In der Wirtschaft, der Forschung, der Politik, aber vor allem in unserem alltäglichen Leben. Das Thema Digitalisierung ist allgegenwärtig und neue Technologien überschwemmen den Markt. Computer und Smartphones sind nicht mehr aus unserer Welt wegzudenken. Die Digitalisierung und wachsende Bevölkerungszahlen erfordern immer mehr Strom, jedoch soll dieser nicht mehr aus fossilen Brennstoffen erzeugt werden. Neben dem Atomausstieg wurde im Jahr 2011 auch eine umfassende Energiewende vom deutschen Bundestag beschlossen, die eine Steigerung des Anteils der Stromerzeugung aus erneuerbaren Energien bis 2030 auf bereits mindestens 50 Prozent vorsieht [2]. Daraufhin wurden mithilfe staatlicher Förderung in den letzten Jahren viele Photovoltaikanlagen auf öffentlichen Gebäuden und Privathäusern installiert. Nach einer aktuellen Studie des Fraunhofer-Instituts für Solare Energiesysteme (ISE) [3] wurden 2017 mit ca. 40 TWh etwa 7,2 % des Netto-Stromverbrauchs in Deutschland durch Photovoltaik abgedeckt - und es ist noch kein Ende dieses Trends in Sicht. Doch diese Entwicklung bringt auch neue Probleme mit sich. Ein viel diskutiertes Thema ist der schwierige Übergang von zentraler zur dezentralen Stromerzeugung (siehe z.B. [4]). Unser aktuelles Stromnetzwerk ist diesen neuen Anforderungen nicht gewachsen und ein sehr teurer und aufwendiger Netzausbau scheint unumgänglich. Daher ist es jetzt dringend notwendig, sich mit den aufkommenden Problemen zu beschäftigen.

Doch nicht nur die Herausforderungen, die durch die Entstehung dezentraler Stromerzeuger auftreten, müssen angegangen werden. Ein weiteres Thema, das eng mit der Entwicklung erneuerbarer Energien und neuer Netzgestaltungskonzepte verbunden ist, ist die Elektromobilität. Und das nicht nur aufgrund des Diesel-Skandals, der weltweites Aufsehen erregte und die Gespräche über alternative Kraftstoffe anheizte. Aus wachsender Angst vor unvorhersehbaren Auswirkungen der Klimaerwärmung wurden

EU-Normen verschärft, die den Schadstoffausstoß von Fahrzeugen reduzieren sollen¹. Die Automobilindustrie ist somit gezwungen, emissionsarme Alternativen zu Verbrennungsmotoren anzubieten. Dabei liegt der Fokus aktuell auf der Entwicklung rein Batterie-getriebener Fahrzeuge (kurz: BEV, engl. für *Battery Electric Vehicle*). Die deutsche Bundesregierung unterstützt den Ausbau von Elektromobilität unter anderem durch Kaufprämien oder den Ausbau der Ladeinfrastruktur [6]. Dass auch die bayrische Landesregierung von der Zukunftsfähigkeit des Elektroautos überzeugt ist, spiegelt sich in der Eröffnung des Zentrums für Batterieforschung *BayBatt* an der Universität Bayreuth im September diesen Jahres wider, in welchem unter anderem an effektiveren Batterien für BEVs geforscht wird [7].

Doch es ist zu erwarten, dass dieser Umschwung zur Überlastung unseres Stromnetzes führen wird. Die Studie *“Der E-Mobilitäts Blackout”* der Unternehmensberatung John Wyman aus dem Januar diesen Jahres [8] zeigt beispielsweise, dass bereits in fünf bis zehn Jahren mit Stromausfällen aufgrund der angestiegenen Elektromobilität zu rechnen ist. Stromnetze in ganz Europa sind nicht stark genug, um die erwartete Anzahl an Batterie-betriebenen Fahrzeugen zu unterstützen. Dies liegt unter anderem an dem im Vergleich zum durchschnittlichen Stromverbrauch eines Haushalts sehr hohem Verbrauch der Elektroautos. Ein Ausweg wäre also der Ausbau unserer Stromnetze. Dieser ist allerdings nicht nur teuer und dauert sehr lange, sondern wird, wie z.B. im Fall der Mega-Trasse *“Südlink”*, von den Bürgern selbst aus diversen Gründen erschwert oder sogar verhindert [9].

Verantwortlich für die Überlastung des Stromnetzes ist jedoch nicht allein der hohe Stromverbrauch der Elektroautos, sondern ebenso das Ladeverhalten der Besitzer. Diese schließen ihren Wagen in der Regel nach dem Heimkommen direkt an die Steckdose an, um die Batterie so schnell wie möglich wieder voll zu laden. Da ein Großteil der deutschen Bevölkerung sehr ähnliche Arbeitszeiten hat, kommen die meisten Arbeitnehmer auch zu ähnlichen Uhrzeiten nach Hause. Dieses Verhalten verursacht zusammen mit dem hohen Stromverbrauch der BEVs einen Spannungs-Peak im Stromnetz, da die Gesamtstromnachfrage innerhalb weniger Stunden um ein Vielfaches ansteigt. Ein weiterer Einflussfaktor auf Schwankungen in der Netzauslastung ist die gestiegene Anzahl an hauseigenen Photovoltaik Anlagen, die im Zuge des Erneuerbare-Energien-Gesetz (EEG) stark gefördert wurden. Jedoch produzieren diese dann am meisten Strom, wenn er nicht gebraucht wird. Der Stromverbrauch ist unter der Woche vormittags und abends am größten, dies sind aber Zeiten, in denen die Solaranlage i.d.R. am wenigsten Strom erzeugt. Das bedeutet, dass der mittags und nachmittags erzeugte Strom zu einem großen Teil zurück ins Netz eingespeist wird und die bereits geringe Nachfrage sogar noch weiter senkt. Es braucht also Speichermedien sowie eine intelligente Steuerung, um Über- und Unterspannungen im Netz zu vermeiden.

¹So wurde beispielsweise 2016 eine europäische Strategie für emissionsarme Mobilität beschlossen ([5]).

1.2 Ziel dieser Masterarbeit

Zu Beginn dieser Arbeit werden wir Grundlagen zur mathematischen Kontrolltheorie und für die Optimierung mit modellprädiktiver Regelung erarbeiten. Dabei werden wir einige der wichtigsten Aussagen über Stabilität und Performance herausarbeiten. Anschließend werden wir genauer auf die Optimierungsalgorithmen eingehen, die in der zugrundeliegenden Dissertation von Dr. Braun *Hierarchical distributed optimization and predictive control of a smart grid* [1] vorgestellt werden. Danach ist die Arbeit in zwei Teile aufgeteilt:

In Kapitel 4 werden wir analysieren, wie wichtig genaue Schätzungen des Stromverbrauchs für die Optimierung sind. Außerdem werden wir die Frage klären, ob aggregierte Informationen über den Verbrauch ausreichend sind. Des Weiteren werden wir untersuchen, wie gut der Algorithmus mit fehlerhaften Verbrauchsprognosen umgehen kann.

In Kapitel 5 werden wir das in Kapitel 3 eingeführte Modell und den Optimierungsalgorithmus um den Einsatz von Elektroautos erweitern. Um die Performance der Erweiterung zu untersuchen, werden wir im letzten Abschnitt der Arbeit ausführliche Simulationen durchführen. Dafür wurde ein Matlab Programm entwickelt, das das Modell für gegebene Daten mit dem vorgestellten Optimierungsalgorithmus simuliert. Anhand der Ergebnisse analysieren wir die Auswirkungen, die unterschiedliche Anteile an Elektroautos auf den verteilten Optimierungsalgorithmus haben.

Wir schließen mit einem Fazit und einem Ausblick auf künftig mögliche Arbeiten im Bezug auf das betrachtete Modell. Im Anhang finden sich nützliche Informationen zur verwendeten Software sowie eine Beschreibung der Implementierung der Simulation in Matlab.

Mathematische Grundlagen

Das Ziel dieser Arbeit ist es, ein bestehendes Modell zur verteilten, modellprädiktiven Regelung eines Smart Grids zu erweitern. Daher müssen wir zunächst die Grundlagen sowie die Funktionsweise des zugrundeliegenden Modells verstehen. Wir beginnen mit der Definition grundlegender Kontrollsysteme und deren Eigenschaften, anschließend betrachten wir zeitdiskrete Optimalsteuerungsprobleme und erklären den Einsatz der modellprädiktiven Regelung bei Stabilisierungsproblemen und ökonomische Problemen.

2.1 Mathematische Kontrolltheorie

2.1.1 Kontrollsysteme

In der Regel liegt zu einer gegebenen Zeit $t \in \mathbb{R}$ und Dimensionen $n, m \in \mathbb{N}$ ein System vor, dessen *Zustand* $x(t) \in \mathbb{X} \subseteq \mathbb{R}^n$ mittels geeigneter *Kontrolle* $u(t) \in \mathbb{U} \subseteq \mathbb{R}^m$ zu einem bestimmten Zielzustand gesteuert werden soll. Oft können diese Systeme durch eine zeitinvarianten Differentialgleichung der Form

$$\dot{x}(t) = f(x(t), u(t)) \tag{2.1}$$

beschrieben werden. Dabei steht die auch häufig *Systemdynamik* genannte Funktion $f : \mathbb{X} \times \mathbb{U} \rightarrow \mathbb{X}$ für die Zustandsänderung bei gegebener Kontrolle. Dabei unterscheiden wir je nach der Beschaffenheit von \mathbb{X} , \mathbb{U} und f zwischen linearen und nichtlinearen Kontrollsystemen. Nach dem Satz von Carathèodory existiert unter bestimmten Voraussetzungen¹ zu jedem Startzustand $x_0 \in \mathbb{X}$ und jeder Kontrollfunktion $u \in \mathcal{U} := \{u : \mathbb{R} \rightarrow \mathbb{U}\}$ eine Lösung der Differentialgleichung. Diese nennen wir $\phi(\cdot, x_0, u) : \mathbb{R} \rightarrow \mathbb{X}$. Das Ziel ist in der Regel die Steuerung des Systems zu einem Gleichgewichtspaar $(x^*, u^*) \in \mathbb{X} \times \mathbb{U}$, d.h. $f(x^*, u^*) = x^*$. Wir streben dabei eine Stabilisierung mittels eines *Feedbacks* $\mu : \mathbb{X} \rightarrow \mathbb{U}$ an. Da wir das System so auf

$$\dot{x}(t) = f(x(t), \mu(x(t))) =: F(x(t)) \tag{2.2}$$

¹Diese bestehen im Grunde aus der Stetigkeit von f und Messbarkeit der Kontrollen, für Genaueres siehe z.B. [10].

reduzieren können, wobei $F : \mathbb{X} \rightarrow \mathbb{X}$ eine Funktion ist, die durch das Feedback μ induziert wird. Die Lösung dieser Differentialgleichung nennen wir $\phi(\cdot, x_0) : \mathbb{R} \rightarrow \mathbb{X}$

Um die Stabilität eines Gleichgewichts zu definieren, benötigen wir zunächst einige Vergleichsfunktionen. Diese werden auch im Folgenden des Öfteren von Bedeutung sein.

Definition 2.1.1.

- $\mathcal{K} := \{\alpha : \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+ \mid \alpha \text{ stetig, streng monoton wachsend, } \alpha(0) = 0\}$
- $\mathcal{K}_\infty := \{\alpha \in \mathcal{K} \mid \alpha \text{ ist unbeschränkt}\}$
- $\mathcal{L} := \{\gamma : \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+ \mid \gamma \text{ stetig, streng monoton fallend, } \gamma(t) \xrightarrow{t \rightarrow \infty} 0\}$
- $\mathcal{KL} := \{\beta : \mathbb{R}_0^+ \times \mathbb{R}_0^+ \rightarrow \mathbb{R}_0^+ \mid \beta \text{ stetig, } \beta(\cdot, t^*) \in \mathcal{K}, \beta(r^*, \cdot) \in \mathcal{L} \forall t^*, r^* > 0\}$

Wenn wir zeigen können, dass der Abstand der Zustände der Trajektorie $\phi(\cdot, x_0)$, die durch den Startzustand x_0 gegeben ist, zum Gleichgewicht von oben durch eine streng monoton gegen 0 fallende Funktion beschränkt ist, so nennen wir das Gleichgewicht asymptotisch stabil.

Definition 2.1.2. Sei x^* ein Gleichgewicht des Systems (2.2). Das Gleichgewicht x^* heißt (lokal) asymptotisch stabil, wenn eine offene Umgebung \mathcal{O} von x^* und eine Funktion $\beta \in \mathcal{KL}$ existieren, sodass:

$$\forall x_0 \in \mathcal{O} \forall t \in \mathbb{R}_0^+ : |\phi(t, x_0) - x^*| \leq \beta(|x_0 - x^*|, t)$$

Diese Eigenschaft heißt global, falls $\mathcal{O} = \mathbb{X}$ und exponentiell, falls $\beta(r, t) = ce^{-\sigma t}r$ für $c, \sigma > 0$.

Ein essentielles Mittel zur Lösung von nichtlinearen kontrolltheoretischen Problemen sind die sogenannten Lyapunov-Funktionen. Eine solche Funktion beschreibt eine Art "Potential" eines Zustands und ist selbst durch Vergleichsfunktionen aus \mathcal{K} beschränkt. Sie stellt sicher, dass das Potential eines Zustands durch das Anwenden der Systemdynamik abnimmt. Eine entscheidende Erkenntnis ist, dass die Existenz einer solchen Funktion für die asymptotische Stabilität des Systems ausreichend ist.

Definition 2.1.3. Sei \mathcal{O} eine offene Umgebung um das Gleichgewicht x^* . Eine stetige Funktion $V : \mathcal{O} \rightarrow \mathbb{R}$, die auf $\mathcal{O} \setminus \{x^*\}$ stetig differenzierbar ist, heißt (lokale) Lyapunov-Funktion für das durch (2.2) gegebene Kontrollsystem, wenn folgende Bedingungen gelten:

Es existieren $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ und eine stetige Funktion $W : \mathcal{O} \rightarrow \mathbb{R}$, sodass für alle $x \in \mathcal{O} \setminus \{x^*\}$ gilt:

$$W(x) > 0 \tag{2.3}$$

$$\alpha_1(|x - x^*|) \leq V(x) \leq \alpha_2(|x - x^*|) \tag{2.4}$$

$$DV(x)F(x) \leq -W(x) \tag{2.5}$$

V heißt globale Lyapunov-Funktion, falls $\mathcal{O} = \mathbb{X}$

Satz 2.1.1. Sei x^* ein Gleichgewicht des Systems (2.2). Existiert eine lokale/globale Lyapunov Funktion, so ist das Gleichgewicht x^* lokal/global asymptotisch stabil.

Beweis. Zum Beispiel in Kapitel 5.7 von [11] □

Unser Kontrollsystem entfernt sich jedoch ein wenig von diesem grundlegenden Modellierungsansatz. Zum einen betrachten wir kein kontinuierliches System, sondern nur diskrete Zeitpunkte, zu denen der Zustand des Systems gemessen wird und Kontrollen gewählt werden können. Dies liegt z.B. daran, dass die Laderaten der Haushaltsbatterien, die nicht zu jedem Zeitpunkt geändert werden können.

Zum anderen wird es bei unserer Optimierung nicht auf ein klassisches Stabilisierungsproblem hinauslaufen, da wir das Gleichgewicht, zu dem wir das System steuern möchten, im Vorhinein nicht kennen. Stattdessen ist unser Ziel, eine gewählte Steuerung anhand der von ihr erzeugten Stromnachfrage zu bewerten und unser System zu dem besten Zustand hin zu steuern. Dies führt uns zu der folgenden Klasse von Kontrollsystemen.

2.1.2 Zeitdiskrete Kontrollsysteme und optimale Steuerung

Bevor wir uns den Optimalsteuerungsproblemen widmen, wollen wir zunächst die Klasse der *zeitdiskreten Kontrollsysteme* definieren. Diese unterscheiden sich zunächst von klassischen Kontrollsystemen dadurch, dass Zustände und Steuerungen nur zu diskreten Zeitpunkten bestimmt werden können. Das bedeutet, dass das System für jeden Zeitpunkt $k \in \mathbb{N}$ einen Zustand $x(k) \in \mathbb{X}$ sowie eine Kontrolle $u(k) \in \mathbb{U}$ besitzt. Des Weiteren wollen wir uns nun auch auf Systeme beziehen, deren Systemdynamik sich über die Zeit verändern kann. Das bedeutet, dass wir statt zu einem Zielgleichgewicht (x^*, u^*) nun zu einer Trajektorie von Referenzzuständen $x^{\text{ref}} : \mathbb{N} \rightarrow \mathbb{X}$ hin steuern. Aus der Systemdynamik wird die Funktion $f : \mathbb{N} \times \mathbb{X} \times \mathbb{U} \rightarrow \mathbb{X}$, die den Zustand des Systems zum nächsten Zeitpunkt

$$x(k+1) = f(k, x(k), u(k)) \quad (2.6)$$

bestimmt. Da die Berechnung des Zustands nun von der Zeit abhängig ist, muss auch die Feedback Regelung $\mu : \mathbb{N} \times \mathbb{X} \rightarrow \mathbb{U}$ zeitabhängig gegeben sein. Nach Einsetzen des Feedbacks erhalten wir mit $F : \mathbb{N} \times \mathbb{X} \rightarrow \mathbb{X}$ also

$$x(k+1) = f(x(k), \mu(k, x(k))) = F(k, x(k)). \quad (2.7)$$

Die Trajektorie von Zuständen zum Startzustand $x_0 \in \mathbb{X}$ ist jetzt zusätzlich vom Startzeitpunkt $n_0 \in \mathbb{N}$ abhängig. Daher nennen wir diese $\phi(\cdot, n_0, x_0)$, wobei diese Trajektorie nur Zustände zu diskreten Zeitpunkten enthält. Die Begriffe der asymptotischen Stabilität und der Lyapunov-Funktionen lassen sich mit ein paar kleinen Änderungen übertragen. Folgende Definitionen und Sätze sowie deren Beweise finden sich in Kapitel 2 von [12].

Definition 2.1.4. Sei $x^{\text{ref}} : \mathbb{N} \rightarrow \mathbb{X}$ eine Trajektorie für das System (2.7), das heißt $x^{\text{ref}}(k+1) = F(k, x^{\text{ref}}(k)) \forall k \in \mathbb{N}$.

1. x^{ref} heißt (lokal) uniform asymptotisch stabil, wenn ein $\varepsilon > 0$ und eine Funktion $\beta \in \mathcal{KL}$ existieren, sodass

$$|\phi(k, n_0, x_0) - x^{\text{ref}}(k)| \leq \beta(|x_0 - x^{\text{ref}}(n_0)|, k - n_0) \quad (2.8)$$

für alle $x_0 \in \mathcal{B}_\varepsilon(x^{\text{ref}}(n_0))$ und alle $n_0, k \in \mathbb{N}$ mit $k \geq n_0$ gilt.

x^{ref} heißt uniform asymptotisch stabil auf einer vorwärts invarianten Familie von Mengen $Y(k)$ ² mit $x^{\text{ref}}(k) \in Y(k)$, wenn es ein $\beta \in \mathcal{KL}$ gibt, sodass (2.8) für alle $n_0, k \in \mathbb{N}$ mit $k \geq n_0$ und alle $x_0 \in Y(n_0)$ gilt.

x^{ref} heißt global uniform asymptotisch stabil, wenn $Y(k) = \mathbb{X}$ für alle $k \in \mathbb{N}$ gilt.

2. Sei $S(k) \subseteq \mathbb{X}$ und $\mathcal{S} := \{(k, x) | k \in \mathbb{N}, x \in S(k)\}$. Eine stetige Funktion $V : \mathcal{S} \rightarrow \mathbb{R}_0^+$, die auf $\mathcal{O} \setminus \{0\}$ stetig differenzierbar ist, heißt uniform zeitverändernde Lyapunov-Funktion für das durch (2.7) gegebene Kontrollsystem, wenn folgende Bedingungen gelten:

- (a) Es existieren $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$ sodass

$$\alpha_1(|x - x^{\text{ref}}(k)|) \leq V(k, x) \leq \alpha_2(|x - x^{\text{ref}}(k)|) \quad (2.9)$$

für alle $k \in \mathbb{N}$ und alle $x \in S(k)$ gilt.

- (b) Es existiert ein $\alpha_V \in \mathcal{K}$, sodass

$$|V(k+1, F(x))| \leq |V(k, x)| - \alpha_V(|x - x^{\text{ref}}(k)|) \quad (2.10)$$

für alle $k \in \mathbb{N}$ und alle $x \in S(k)$ mit $F(k, x) \in S(k+1)$ gilt.

Satz 2.1.2. Sei x^{ref} eine Trajektorie für das Kontrollsystem (2.7) und V eine uniforme zeitverändernde Lyapunov Funktion auf $S(k)$ nach Definition 2.1.4. Wenn für jedes $S(k)$ eine Kugel $\mathcal{B}_V(x^{\text{ref}}(k))$ mit $F(k, x) \in S(k+1)$ für alle $x \in \mathcal{B}_V(x^{\text{ref}}(k))$ existiert, so ist die Trajektorie x^{ref} lokal asymptotisch stabil. Wenn die Familie der Mengen $S(k)$ vorwärts invariant ist, dann ist x^{ref} asymptotisch stabil. Wenn $S(k) = \mathbb{X}$ für alle $k \in \mathbb{N}$ gilt, so ist x^{ref} global asymptotisch stabil.

Da in der Realität z.B. durch Messfehler meistens keine punktgenaue Kontrolle möglich ist, wird in [12] neben dem klassischen Stabilitätsbegriff die sogenannte *P-praktisch asymptotische Stabilität* eingeführt.

Definition 2.1.5. Sei x^{ref} eine Trajektorie des Systems (2.7), $Y(k) \subseteq \mathbb{X}$ eine vorwärts invariante Familie von Mengen und $P(k) \subset Y(k)$ Teilmengen von $Y(k)$. Die Trajektorie x^{ref} heißt *P-praktisch uniform asymptotisch stabil* auf $Y(k)$, wenn eine Funktion $\beta \in \mathcal{KL}$ existiert, sodass (2.8) für alle $x_0 \in Y(n_0)$, $n_0, k \in \mathbb{N}$ mit $k \geq n_0$ und $\phi(k, n_0, x_0) \notin P(k)$.

²Eine Familie von Mengen $Y(k)$ heißt vorwärts invariant, falls $F(k, x) \in Y(k+1)$ für alle $x \in Y(k)$ und alle $k \in \mathbb{N}$ gilt.

Satz 2.1.3. Sei x^{ref} eine Trajektorie des Systems (2.7), $Y(k) \subseteq \mathbb{X}$ eine vorwärts invariante Familie von Mengen und $P(k) \subset Y(k)$ Teilmengen von $Y(k)$. Existiert eine uniforme zeitvariierende Lyapunov Funktion auf $S(k) = Y(k) \setminus P(k)$ nach Definition 2.1.4, so ist x^{ref} P -praktisch asymptotisch stabil auf $Y(k)$.

Bisher haben wir die Theorie zu reinen Stabilisierungsproblemen dargestellt. Oft ist es jedoch der Fall, dass das Gleichgewicht unbekannt ist oder kein eindeutiges Gleichgewicht existiert. Stattdessen ist es häufig möglich, eine Zustandstrajektorie nach bestimmten Kriterien zu bewerten. Als Ergebnis erhält man eine Kostenfunktion, welche durch die gewünschte Kontrollfolge minimiert wird. In einem diskreten Kontrollsystem wie (2.6) bedeutet das, dass zu jedem Zeitpunkt *Stufenkosten* $g : \mathbb{N} \times \mathbb{X} \times \mathbb{U} \rightarrow \mathbb{R}_0^+$ entstehen, welche über den betrachteten Zeitraum aufsummiert werden. Da wir zu jedem Zeitpunkt Steuerungen wählen, interessieren wir uns für die Kosten einer *Kontrollfolge* $\mathbf{u}(\cdot) \in \mathbb{U}^\infty := \{u : \mathbb{N} \rightarrow \mathbb{U}\}$. Diese ergeben sich demnach abhängig von einem Startzustand $x_0 \in \mathbb{X}$ als

$$J : \mathbb{X} \times \mathbb{U}^\infty \rightarrow \mathbb{R}_0^+, \quad J(x_0, \mathbf{u}(\cdot)) := \sum_{k=0}^{\infty} g(k, x_{\mathbf{u}}(k), u(k)).$$

Dabei erzeugt $\mathbf{u}(\cdot)$ eine *Zustandstrajektorie* $x_{\mathbf{u}}(\cdot)$, die sich mithilfe des Startzustandes über die Systemdynamik berechnen lässt:

$$x_{\mathbf{u}}(0) := x_0, \quad x_{\mathbf{u}}(k+1) := f(x(k), u(k)), \quad k \in \mathbb{N}.$$

Wir suchen also eine Steuerung, die die gegebenen Kosten minimiert. Dazu lösen wir ein sogenanntes *Optimalsteuerungsproblem*, welches durch das Optimierungsproblem

$$\min_{\mathbf{u}(\cdot) \in \mathbb{U}^\infty} J(x_0, \mathbf{u}(\cdot))$$

gegeben ist. Dabei ist auch hier eine *Feedback-Regelung* $\mu : \mathbb{N} \times \mathbb{X} \rightarrow \mathbb{U}$ unser Ziel, d.h. dass wir die Kontrolle abhängig vom Zustand wählen können.

In realen Problemen, die bis zu einem unbekanntem Zeitpunkt optimal gesteuert werden sollen ist es oft nicht möglich, bereits alle Informationen über das System zu besitzen geschweige denn eine Feedback-Regelung anzugeben, die für alle Zeitpunkte die optimalen Steuerungen liefert. Häufig ist es aber umsetzbar, für einen gewissen Zeitraum in die Zukunft genug Angaben machen zu können, um das System zumindest für diesen Zeitraum optimal zu steuern. Genauso verhält es sich bei unserem vorliegenden Problem der Minimierung von Spannungsschwankungen innerhalb eines Smart Grids. Wir können nur für einen begrenzten Zeitraum präzise Vorhersagen zu Stromerzeugung und -verbrauch machen und daher nur für diesen Zeitraum eine optimale Steuerung berechnen. Da sich die Auswirkungen einer gewählten Kontrolle in einem zeitlich unbegrenzten System jedoch erst zu einem viel späteren Zeitpunkt bemerkbar machen können, benötigen wir eine clevere Vorgehensweise, die dieses Problem umgeht. Genau an dieser Stelle kommt uns das Prinzip der modellprädiktive Regelung zu Gute, auf das wir im Folgenden genauer eingehen.

2.2 Nichtlineare Modellprädiktive Regelung (NMPC)

2.2.1 Klassisches NMPC

Nichtlineare modellprädiktive Regelung (engl. *Nonlinear Model Predictive Control*, kurz: NMPC) ist ein Verfahren zur Lösung von gegebenenfalls nichtlinearen Optimalsteuerungsproblemen wie den oben vorgestellten Stabilisierungsproblemen. Die Idee des MPC Verfahrens ist es, das vorliegende (gegebenenfalls zeitlich unbeschränkte) diskrete Problem in mehrere Teilprobleme mit festem Horizont $N \in \mathbb{N}$ zu unterteilen. Diese werden dann nacheinander gelöst, wobei die erhaltenen Lösungen iterativ eine Lösung für das Gesamtproblem bilden. Dieses Vorgehen ist insbesondere dann gut geeignet, wenn relevante Informationen nur für eine bestimmte Zeit in die Zukunft vorhergesagt werden können, allerdings ein System mit unbekanntem bzw. unendlichem Horizont vorliegt.

Algorithmus 1 zeigt das Vorgehen für ein Optimalsteuerungsproblem, wie es in Abschnitt 2.1 definiert wurde. Zum Zeitpunkt $k \in \mathbb{N}$ berechnet man ausgehend vom Startzustand die optimale Steuerung für die nächsten N Zeitpunkte. Dadurch erhält man eine Folge von Kontrollen $\mathbf{u}(k; N) := (u(k), \dots, u(k + N - 1)) \in \mathbb{U}^N$, die zum Zeitpunkt k beginnt und N Schritte in die Zukunft reicht, sowie die durch $\mathbf{u}(k; N)$ erzeugte Trajektorie $x_{\mathbf{u}} \in \mathbb{X}^N$ der Zustände

$$x_{\mathbf{u}}(k) := x(k), \quad x_{\mathbf{u}}(m + 1) := f(m, x_{\mathbf{u}}(m), u(m)), \quad \forall m \in \{k, \dots, k + N - 1\}. \quad (2.11)$$

Um optimale Kontrollen zu wählen, die das System zu einer Zieltrajektorie steuern, benötigen wir Stufenkosten $g : \mathbb{N} \times \mathbb{X} \times \mathbb{U} \rightarrow \mathbb{R}^+$, die die Abweichung von der Referenztrajektorie x^{ref} bestrafen. Oft strebt man neben einem Zielzustand auch eine bestimmte, beispielsweise möglichst energieeffiziente, Regelung an. Daher ermöglichen wir zusätzlich die Bestrafung der Abweichung von einem bestimmten $\mathbf{u}^{\text{ref}} : \mathbb{N} \rightarrow \mathbb{U}$. Eine klassische Wahl der Stufenkosten ist

$$g(m, x_{\mathbf{u}}(m), u(m)) := |x_{\mathbf{u}}(m) - x^{\text{ref}}(m)|^2 + \lambda |u(m) - u^{\text{ref}}(m)|^2, \quad (2.12)$$

wobei $\lambda \geq 0$ ein beliebiger Skalierungsparameter ist. Eine Kontrollfolge $\mathbf{u} \in \mathbb{U}^N$ wird anhand der durchschnittlichen Kosten bewertet, die sie in den nächsten N Zeitpunkten durch die von ihr und dem Startzustand $x(k) \in \mathbb{X}$ induzierten Zustandstrajektorie erzeugt. Daher definieren wir die Kostenfunktion $J_N : \mathbb{N} \times \mathbb{X} \times \mathbb{U}^N \rightarrow \mathbb{R}^+$ als

$$J_N(k, x(k), \mathbf{u}(\cdot)) := \frac{1}{N} \sum_{m=k}^{k+N-1} g(m, x_{\mathbf{u}}(m), u(m)). \quad (2.13)$$

Wir suchen also eine Kontrollfolge, die für die nächsten N Zeitpunkte die kleinsten Kosten erzeugt. Das resultierende Optimalsteuerungsproblem lautet also

$$\min \quad J_N(k, x(k), \mathbf{u}(\cdot)), \quad (2.14)$$

wobei über alle zulässigen Kontrollfolgen $\mathbf{u}(\cdot) \in \mathbb{U}^N$ optimiert wird. Als Ergebnis erhalten wir eine für das Subsystem optimale Kontrollfolge $\mathbf{u}^*(k; N)$, welche auch

als *Open-Loop* Lösung bezeichnet wird. Das erste Element dieser Folge entspricht unserem Feedback $\mu^*(k, x(k)) = u^*(k)$, welches wir auf unser System anwenden. Dadurch erhalten wir den nächsten Zustand $x(k+1)$. Diesen Zustand nehmen wir als neuen Startzustand und wiederholen alle vorherigen Schritte, d.h. wir berechnen die optimale Kontrollfolge für die nächsten N Zeitpunkte $\mathbf{u}^*(k+1; N)$ und damit wieder $\mu^*(k+1, x(k+1)) = u^*(k+1)$. Durch dieses iterative Vorgehen gelangen wir schließlich zu einer vollständigen Feedback-Regelung μ^* des Systems, der sogenannten *Closed-Loop* Lösung.

Durch diese spezielle Vorgehensweise ist MPC besonders für Probleme geeignet, die auf Vorhersagen basieren, deren Genauigkeit mit längerem Horizont abnimmt. Das ‘‘Verschieben’’ des Horizonts nach jedem Optimierungsschritt ist der Grund dafür, dass MPC auch oft als *Receding Horizon Control* bezeichnet wird.

Beispiel 2.2.1. In Abbildung 2.1 ist die Trajektorie eines MPC Verfahrens bis zum Endzeitpunkt 10 zu sehen. Bei einem Startwert von $x(0) = 1$ und der Systemdynamik $f(k, x, u) = x + u$ wird der Abstand des Zustands und der Kontrolle von 0 quadratisch bestraft, d.h.

$$g(k, x, u) = x^2 + u^2 \quad \forall k \in \mathbb{N}.$$

Um zu sehen, welche Auswirkungen der MPC Horizont $N = 3$ auf die Optimierung hat, setzen wir die Nebenbedingung $x(6) \geq 1$ ein. An den gestrichelten Linien der Open-Loop Lösungen kann man sehen, dass der Algorithmus erst zum Zeitpunkt 3 von der Hürde erfährt und diese in die Optimierung mit einbeziehen kann. Die Closed-Loop Lösung weicht daher zu diesem Zeitpunkt von den vorherigen Open-Loop Lösungen ab und steuert den Zustand wieder nach oben.

Algorithmus 1 Klassisches NMPC mit zeitvariierender Referenz x^{ref}

Input: Lege den MPC-Horizont $N \in \mathbb{N}$, die Anzahl an Optimierungsschritten $\mathcal{N} \in \mathbb{N} \cup \{\infty\}$ und den Startzustand $x(0) \in \mathbb{X}$ fest. Setze $k = 0$.

1. Zum Zeitpunkt k bestimme den Zustand des Systems $x(k)$.
2. Setze $x_0 = x(k)$.
3. Löse das OCP

$$\min J_N(k, x_0, \mathbf{u}(\cdot)) = \sum_{m=k}^{k+N-1} g(m, x_{\mathbf{u}}(m), u(m)),$$

wobei über alle zulässigen Kontrollfolgen $u(\cdot) \in \mathbb{U}^N$ optimiert wird und $x_{\mathbf{u}}(\cdot)$ die durch (2.11) erzeugt wird und erhalte die optimale Kontrollfolge $\mathbf{u}^*(k; N)$.

4. Setze das Feedback als $\mu^*(k, x(k)) = \mathbf{u}^*(k)$ und wende die Systemdynamik an, um in den nächsten Zustand zu steuern.
 5. Setze $k = k + 1$. Falls $k \leq \mathcal{N}$, dann gehe zu 1. Sonst beende den Algorithmus und gib μ^* zurück.
-

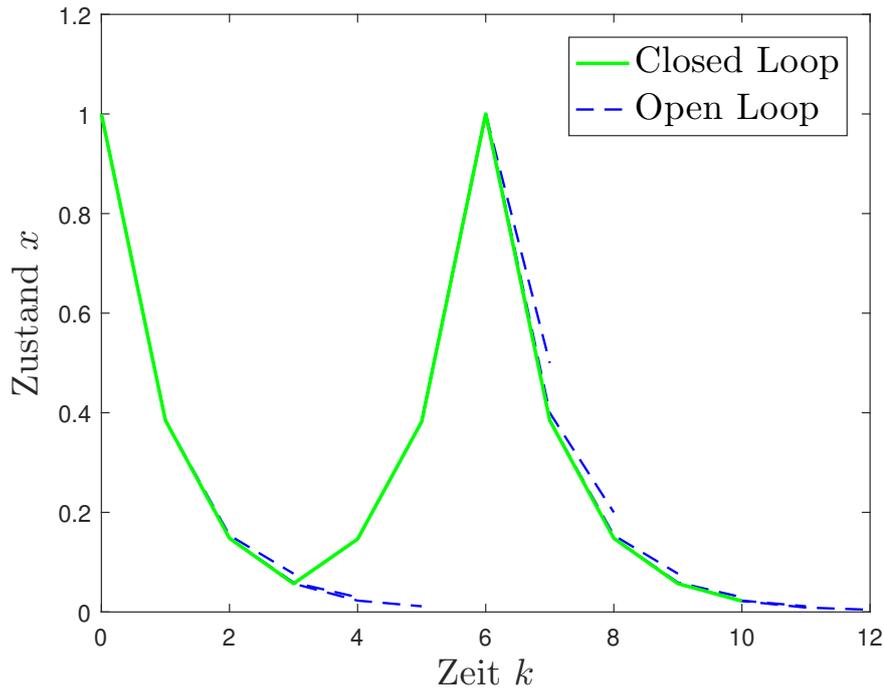


Abbildung 2.1: Open- und Closed-Loop Zustandstrajektorien des Beispiels 2.2.1

Eines der klassischen Prinzipien der optimalen Steuerung ist die *dynamische Programmierung*, welche in den 50er Jahren von Richard Bellman entwickelt wurde. Die Grundidee ist es, ein kompliziertes Problem in kleinere Subprobleme aufzuteilen, die leichter zu lösen sind. Die Lösungen der Subprobleme ergeben dann gemeinsam eine Lösung für das Gesamtsystem. Genauer beschrieben wird das Prinzip beispielsweise in [13]. Da MPC stark an dieses Prinzip erinnert, wundert es nicht, dass die Theorie der dynamischen Programmierung auch im MPC Kontext anwendbar ist. Die folgenden Definitionen und Sätze behandeln diesen Ansatz und finden sich mit Beweisen in Abschnitt 3.4 von [12].

Definition 2.2.1. Für ein Optimalsteuerungsproblem (2.14), dem Startzustand $x_0 \in \mathbb{X}$, einem Zeitpunkt $k \in \mathbb{N}$ und MPC Horizont $N \in \mathbb{N}$ heißt

$$V(k, x_0) := \inf_{\mathbf{u}(\cdot) \in \mathbb{U}^{k,N}(x_0)} J_N(k, x_0, \mathbf{u}(\cdot))$$

die *optimale Wertefunktion*. Eine Kontrollfolge $\mathbf{u}^*(k; N) \in \mathbb{U}^{k,N}(x_0)$ heißt *optimale Kontrollfolge* für x_0 , wenn

$$V(k, x_0) = J_N(k, x_0, \mathbf{u}^*(\cdot))$$

gilt. Die zugehörige Zustandstrajektorie $x_{\mathbf{u}^*}$ heißt *optimale Zustandstrajektorie*.

Satz 2.2.1 (Dynamische Programmierung). *Betrachte das Optimalsteuerungsproblem (2.14) mit Startzustand $x_0 \in \mathbb{X}$. Dann gilt für alle $k, N \in \mathbb{N}$ und $K \in \{1, \dots, N\}$:*

$$V_N(k, x_0) = \inf_{\mathbf{u}(\cdot) \in \mathbb{U}^{k, N-K}(x_0)} \left\{ \sum_{m=k}^{k+K-1} g(m, x_{\mathbf{u}}(m), u(m)) + V_{N-K}(k+K, x_{\mathbf{u}}(K)) \right\}$$

Wenn eine optimale Kontrollfolge \mathbf{u}^* für x_0 existiert, so gilt außerdem

$$V_N(k, x_0) = \sum_{m=k}^{k+K-1} g(m, x_{\mathbf{u}^*}(m), u^*(m)) + V_{N-K}(k+K, x_{\mathbf{u}^*}(K)). \quad (2.15)$$

Das Prinzip der dynamischen Programmierung lässt sich nun auf den NMPC Algorithmus anwenden. Als Ergebnis erhalten wir die Aussage, dass die optimale Wertefunktion durch die Kosten des vom NMPC generierten Feedbacks rekursiv erzeugt wird. Das bedeutet auch, dass die optimale Kontrollfolge aus dem Feedback gewonnen werden kann.

Satz 2.2.2. *Betrachte das Optimalsteuerungsproblem (2.14) mit Startzustand $x_0 \in \mathbb{X}$ und $k, N \in \mathbb{N}$. Existiert eine optimale Kontrollfolge \mathbf{u}^* , so erfüllt das Feedback aus dem NMPC Algorithmus 1 $\mu^*(k, x_0) = u^*(0)$ die Gleichungen*

$$\mu^*(k, x_0) \in \arg \min_{u \in \mathbb{U}^{k,1}(x_0)} \{g(k, x_0, u) + V_{N-1}(k+1, f(k, x_0, u))\} \quad (2.16)$$

sowie

$$V_N(k, x_0) = g(k, x_0, \mu^*(k, x_0)) + V_{N-1}(k+1, f(k, x_0, \mu^*(k, x_0))) \quad (2.17)$$

2.2.2 Ökonomisches NMPC

Wenn wie in Abschnitt 2.1.2 unser Ziel nicht das Steuern des Zustands zu einem bestimmten Ziel ist, sondern andere Eigenschaften einer Kontrollfolge, wie zum Beispiel Energieeffizienz oder Kosten, optimiert werden sollen, so spricht man im allgemeinen von ökonomischen NMPC. Unser vorliegendes Problem fällt in diesen Bereich, da wir Schwankung innerhalb eines Stromnetzes minimieren wollen und kein Ziel-Gleichgewicht bekannt ist. Wir wissen nur, dass wir ein Gleichgewichtspaar (x^*, u^*) anstreben, das minimale Stufenkosten erzeugt, d.h.

$$g(x^*, u^*) \leq g(x, u) \quad \forall x \in \mathbb{X}, u \in \mathbb{U}.$$

Da nicht mehr ohne Weiteres feststellbar ist, dass NMPC unter diesen Bedingungen noch sinnvolle Ergebnisse liefern, führen wir den Begriff der Dissipativität analog zu Kapitel 8 von [12] ein.

Definition 2.2.2. Ein Optimalsteuerungsproblem ist an einem Gleichgewicht (x^*, u^*) strikt dissipativ, falls es Funktionen $\lambda : \mathbb{X} \rightarrow \mathbb{R}$ und $\alpha \in \mathcal{K}_\infty$ gibt, sodass folgende Bedingungen erfüllt sind:

1. λ ist von unten beschränkt
2. $\lambda(x^*) = 0$
3. Für alle $x \in \mathbb{X}$ und $u \in \mathbb{U}$ gilt:

$$g(x, u) - g(x^*, u^*) + \lambda(x) - \lambda(f(x, u)) \geq \alpha(|x - x^*|)$$

Das Optimalsteuerungsproblem heißt dissipativ an (x^*, u^*) falls $\alpha \equiv 0$.

Diese Eigenschaft kann durch ein physikalisches Phänomen veranschaulicht werden: Betrachtet man $\lambda(x)$ als die Energie des Systems in Zustand x , so stellt die strikte Dissipativität sicher, dass durch jede Anwendung der Systemdynamik Energie in der Höhe von mindestens $\alpha(|x - x^*|)$ an die Umgebung verloren geht. Da im Gleichgewicht keine Energie mehr vorhanden ist, strebt unser System diesen Zustand an.

In der Literatur wird zwischen Problemen mit und ohne stabilisierende Endbedingungen unterschieden. Da wir in unserem Modell keine Endbedingungen stellen werden, betrachten wir diesen Fall genauer. Wenn wir den Stabilitätsbegriff etwas einschränken, können wir unter bestimmten weiteren Annahmen zeigen, dass aus der strikten Dissipativität die folgende Form der Stabilität folgt, wie sie in Kapitel 6 von [12] zu finden ist.

Definition 2.2.3. Betrachte den NMPC Algorithmus 1, dessen Feedback Regelung μ^* und die Lösung $x^{\mu^*}(k)$.

1. Das Gleichgewicht heißt *semiglobal asymptotisch stabil im Bezug auf den MPC Horizont N* falls ein $\beta \in \mathcal{KL}$ existiert sodass folgende Aussage gilt: Für jedes $\Delta > 0$ existiert ein $N_\Delta \in \mathbb{N}$ sodass für alle $N \geq N_\Delta$ und alle $x \in \mathbb{X}$ mit $|x - x^*| \leq \Delta$ gilt:

$$|x^{\mu^*}(k) - x^*| \leq \beta(|x - x^*|, k) \quad \forall k \in \mathbb{N}$$

2. Das Gleichgewicht heißt *semiglobal praktisch asymptotisch stabil im Bezug auf den MPC Horizont N* falls ein $\beta \in \mathcal{KL}$ existiert sodass folgende Aussage gilt: Für jedes $\delta > 0$ und $\Delta > \delta$ existiert ein $N_{\delta, \Delta} \in \mathbb{N}$ sodass für alle $N \geq N_{\delta, \Delta}$ und alle $x \in \mathbb{X}$ mit $|x - x^*| \leq \Delta$ gilt:

$$|x^{\mu^*}(k) - x^*| \leq \max\{\beta(|x - x^*|, k), \delta\} \quad \forall k \in \mathbb{N}$$

Satz 2.2.3. Betrachte den NMPC Algorithmus 1 für ein strikt dissipatives Optimalsteuerungsproblem. Erfüllen λ und V_N am Gleichgewicht x^* bestimmte Stetigkeitsannahmen (siehe [12], Kapitel 8), so ist x^* semiglobal praktisch asymptotisch stabil in Bezug auf den MPC Horizont N .

Beweis. Siehe [12], Abschnitt 8.6. □

Diese Grundlagen sollen uns genügen, um das Modell eines Smart Grids aufzustellen. Im Folgenden werden wir uns nicht genauer mit dem Nachweis der Stabilität oder Lösbarkeit auseinander setzen, sondern den Fokus auf die Modellierung und numerische Simulationen legen. In Abschnitt 6.2 werden wir noch auf die Turnpike Eigenschaft eingehen - eine weitere Eigenschaft von Optimalsteuerungsproblemen, die mit der Stabilität von optimalen Gleichgewichten zusammenhängt.

Kapitel 3

Modellierung und optimale Steuerung eines Smart Grids

Im folgenden Kapitel betrachten wir, wie ein Smart Grid als Optimalsteuerungsproblem modelliert werden kann, das mit dem NMPC Verfahren lösbar ist. Dabei werden wir die Modellierung der zugrundeliegenden Dissertation [1] übernehmen. Diese wird im Folgenden erläutert. Insbesondere wird der zur Lösung unseres Problems eingesetzte verteilte Optimierungsalgorithmus analysiert. Am Ende dieses Kapitels werden wir die Erweiterung des bestehenden Modells um Elektroautos einführen und dabei entstehende Herausforderungen behandeln.

3.1 Modellbezogene Grundlagen

3.1.1 Systemparameter, Zustände und Steuerungen

Ein Smart Grid ist in unserem Sinne ein Netzwerk aus mehreren Haushalten, die durch die Nutzung elektrischer Geräte als Stromverbraucher und durch die Erzeugung von Strom durch Photovoltaikanlagen als Stromerzeuger agieren. Jeder Haushalt kann eine Haushaltsbatterie sowie ein Elektroauto besitzen. Für die Stromnachfrage entscheidend ist die Wahl, wann und wie stark die Haushaltsbatterie oder die Batterie des Elektroautos¹ ge- oder entladen werden. Wir betrachten insgesamt $\mathcal{I} \in \mathbb{N}$ viele Haushalte, die in einem Netzwerk verbunden sind. Unser Ziel ist es, dessen Verhalten in einem zeitdiskreten System bis zu einem Endzeitpunkt $\mathcal{N} \in \mathbb{N} \cup \{\infty\}$ zu optimieren. Dafür unterteilen wir die Zeit in Intervalle der Länge $T \in \mathbb{R}_{>0}$. Jeder Haushalt bildet mit einer Photovoltaikanlage und einer Batterie ein sogenanntes *Residential Energy System* (kurz: RES) und ist damit gleichzeitig Stromverbraucher und -erzeuger. Für jedes RES $i \in \mathbb{N}_{\mathcal{I}} := \{1, \dots, \mathcal{I}\}$ und für jeden Zeitpunkt $k \in \mathbb{N}_{\mathcal{N}} := \{0, \dots, \mathcal{N}\}$ führen wir mehrere Größen ein, die in Abbildung 3.1 dargestellt sind²:

¹Damit ist von der Batterie die Rede, die den Strom zum Fahren liefert. Diese ist nicht zu verwechseln mit der Batterie, die in einem Auto mit Verbrennermotor zum Anlassen verwendet wird. Des Weiteren sprechen wir nur von wiederaufladbaren Batterien, wir unterscheiden also nicht wie im Sprachgebrauch üblich zwischen den Begriffen Batterie und Akku.

²Bzw. $\mathbb{N}_{\mathcal{N}} = \mathbb{N}$, falls $\mathcal{N} = \infty$.

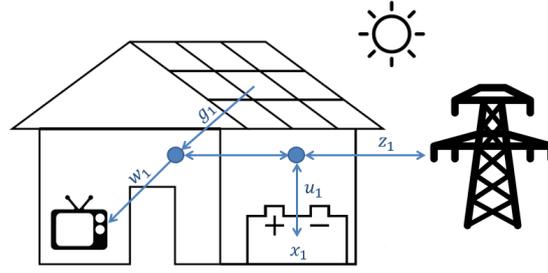


Abbildung 3.1: Die Variablen des RES 1 im Smart Grid

Die aktuelle Ladung der Hausbatterie in Kilowattstunden (kWh) $x_i(k) \in \mathbb{R}$ entspricht dem Zustand des RES. Zu jedem Zeitpunkt muss entschieden werden, ob und wie stark die Batterie ge- oder entladen werden soll. Diese Entscheidung spiegelt sich in der Kontrolle $u_i(k) \in \mathbb{R}$ in Kilowatt (kW) wider. Dabei steht ein negatives u für das Entladen und ein positives für das Aufladen der Batterie. Wir legen fest, dass die Kontrolle innerhalb eines Intervalls der Länge T nicht geändert werden kann. So wird die Batterie eines RESs beispielsweise bei $T = 0.5$ und $u_i(k) = 2$ im Zeitpunkt k um genau 1 kWh aufgeladen. Eine weitere Größe $z_i(k) \in \mathbb{R}$ gibt die tatsächliche Stromnachfrage des RES i zum Zeitpunkt k in kW an. Diese hängt direkt von der Stromerzeugung $g_i(k) \in \mathbb{R}_{\geq 0}$ und dem -verbrauch $w_i(k) \in \mathbb{R}_{\geq 0}$ des RES (beides in kW) ab und wird für die Kommunikation zwischen den RES im Folgenden noch eine wichtige Rolle spielen.

Durch Anpassungen an der Systemdynamik und den Nebenbedingungen können diese Größen analog zum Vorgehen in [1] hinzugefügt werden. Wir betrachten also die Systemdynamik

$$x_i(k+1) = f_i(x_i(k), u_i(k)) = x_i(k) + Tu_i(k) \quad (3.1)$$

$$z_i(k) = w_i(k) - g_i(k) + u_i(k). \quad (3.2)$$

Bemerkung 3.1.1. Die tatsächliche Stromnachfrage z_i wird im Folgenden auch *Kommunikationsvariable* genannt. Da sich $u_i(k)$ mit den gegebenen Informationen zu Erzeugung und Verbrauch direkt aus $z_i(k)$ berechnen lässt (und umgekehrt), werden wir z_i wie die Steuerung für das RES i behandeln.

Oft werden wir nur an Durchschnittswerten über alle RES interessiert sein. Für eine übersichtlichere Schreibweise sind diese mit einem $\hat{\cdot}$ markiert. Die Durchschnittsnachfrage zum Zeitpunkt k ist beispielsweise gegeben durch

$$\hat{z}(k) := \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} z_i(k).$$

Da die Kapazitäten und maximale Laderaten der Batterien beschränkt sind, führen wir Nebenbedingungen ein, die stets erfüllt sein müssen. Für RES $i \in \mathbb{N}_{\mathcal{I}}$ bezeichnet ein $C_i \in \mathbb{R}_{\geq 0}$ die Kapazität der Hausbatterie. Wir nehmen an, dass sich diese während des Betrachtungszeitraums nicht verändert. Die Laderate ist durch ein $\bar{u}_i \in \mathbb{R}_{\geq 0}$ nach oben

bzw. durch ein $\underline{u}_i \in \mathbb{R}_{\leq 0}$ nach unten beschränkt. Es muss also für alle $k \in \mathbb{N}_{\mathcal{N}}$ gelten

$$0 \leq x_i(k) \leq C_i, \quad \underline{u}_i \leq u_i(k) \leq \bar{u}_i.$$

Damit lassen sich die Mengen der zulässigen Zustände und Kontrollen für RES i aufstellen als

$$\mathbb{X}_i := \{x_i(k) \in \mathbb{R} \mid 0 \leq x_i(k) \leq C_i\}, \quad \mathbb{U}_i := \{u_i(k) \in \mathbb{R} \mid \underline{u}_i \leq u_i(k) \leq \bar{u}_i\}. \quad (3.3)$$

Nun setzen wir das Modell in den MPC-Kontext. Unser Ziel im k -ten MPC-Schritt ist es, für einen festgelegten Horizont $N \in \mathbb{N}$ eine Kontrollfolge $\mathbf{u}_i(k; N) \in \mathbb{U}_i^N$ für jedes RES $i \in \mathbb{N}_{\mathcal{I}}$ zu bestimmen. Ausgehend von einem Startzustand $x_{i,0} \in \mathbb{X}_i$ lässt sich die entstehende Zustandstrajektorie $\mathbf{x}_i^{\mathbf{u}_i}(k; N)_{x_{i,0}}$ durch iteratives Anwenden von (3.1) erzeugen³. Die Kontrollfolge $\mathbf{u}_i(k; N)$ ist nur dann zulässig, wenn sie sicherstellt, dass alle Zustände der Trajektorie in \mathbb{X}_i liegen. Die Menge der zulässigen Kontrollfolgen ist also gegeben durch

$$\mathbb{U}_i^{k,N}(x_{i,0}) := \left\{ \mathbf{u}_i(k; N) \in \mathbb{R}^N \left| \begin{array}{l} x_i(k) = x_{i,0} \\ x_i(j+1) = x_i(j) + T u_i(j) \\ u_i(j) \in \mathbb{U}_i, \quad x_i(j) \in \mathbb{X}_i \\ \forall j = k, \dots, k+N-1 \end{array} \right. \right\}. \quad (3.4)$$

Da wir im Folgenden wie in Bemerkung 3.1.1 angesprochen hauptsächlich mit der tatsächlich entstehenden Stromnachfrage z_i arbeiten werden, betrachten wir insbesondere die Menge der zulässigen Stromnachfragefolgen

$$\mathbb{D}_i^{k,N}(x_{i,0}) := \left\{ \mathbf{z}_i(k; N) \in \mathbb{R}^N \left| \begin{array}{l} x_i(k) = x_{i,0} \\ x_i(j+1) = x_i(j) + T u_i(j) \\ z_i(j) = w_i(j) - g_i(j) + u_i(j) \\ u_i(j) \in \mathbb{U}_i, \quad x_i(j) \in \mathbb{X}_i \\ \forall j = k, \dots, k+N-1 \end{array} \right. \right\}. \quad (3.5)$$

Für die Optimierung des Gesamtsystems definieren wir außerdem die Mengen

$$\begin{aligned} \mathbb{U}^{k,N}(x_0) &:= \mathbb{U}_1^{k,N}(x_{1,0}) \times \dots \times \mathbb{U}_{\mathcal{I}}^{k,N}(x_{\mathcal{I},0}) \text{ und} \\ \mathbb{D}^{k,N}(x_0) &= \mathbb{D}_1^{k,N}(x_{1,0}) \times \dots \times \mathbb{D}_{\mathcal{I}}^{k,N}(x_{\mathcal{I},0}). \end{aligned} \quad (3.6)$$

Unser Ziel ist nun die Optimierung dieses Systems bezüglich bestimmter Kriterien. Bevor wir diese genauer bestimmen, vervollständigen wir unsere Nomenklatur. Wie zu Beginn dieses Kapitels erwähnt, wollen wir das Smart Grid bis zu einem Endzeitpunkt \mathcal{N} optimal steuern⁴. Dafür führen wir als *Closed-Loop*-Lösung das Feedback $\mu_i^* : \mathbb{X}_i \rightarrow \mathbb{U}_i$ als Funktion vom Zustands- in den Kontrollraum ein, die uns zu einem gegebenen Zustand für ein RES die optimale Steuerung zurückliefert. Die entstehende

³Analog zum Vorgehen in (2.11).

⁴Oder beliebig lange, falls $\mathcal{N} = \infty$.

Trajektorie bezeichnen wir als $\mathbf{x}_i^{\mu_i^*}(0; \mathcal{N})_{x_{i,0}}$, wobei diese durch iteratives Anwenden der Vorschrift

$$x_i(0) := x_{i,0}, \quad x_i^{\mu_i^*}(k+1) := x_i(k) + T\mu_i^*(x_i(k)) \quad \forall k \in \mathbb{N}_N \quad (3.7)$$

entsteht. Ein Feedback μ_i heißt zulässig, falls die Zustandstrajektorie stets im Raum der zulässigen Zustände \mathbb{X}_i liegt. Da die tatsächlich entstehende Stromnachfrage die entscheidende Größe ist, führen wir analog zu oben auch $z_i^{u_i}(k; N)_{x_{i,0}}$ bzw. $z_i^{\mu_i}(0; \mathcal{N})_{x_{i,0}}$ als Open- bzw. Closed-Loop Lösungen ein.

Bemerkung 3.1.2. Um die Darstellung etwas übersichtlicher zu halten, werden im Folgenden die Abhängigkeit von Zeitpunkt, Horizont und Startzustand nicht explizit angegeben, solange sie nicht für das Verständnis notwendig sind.

Bemerkung 3.1.3. In Kapitel 3.4.1 von [1] werden mehrere Möglichkeiten aufgezeigt, wie man Effizienzprobleme von Batterien, wie zum Beispiel den Stromverlust beim Auf- oder Entladen, in der Modellierung berücksichtigen kann. Diese sind mitunter hauptverantwortlich für die Nichtlinearität des Systems. Da wir jedoch an der reinen Erweiterung des Modells um Elektroautos interessiert sind, betrachten wir das System ohne derartige Verluste. Um jedoch in der Erweiterung die Möglichkeit von derartigen Ergänzungen nicht zu verlieren, betrachten wir ein möglicherweise nichtlineares System, auch wenn die bisherige Modellierung linear ist.

Der nächste Schritt in der Modellierung ist die Wahl geeigneter Stufenkosten für das MPC-Verfahren. Denn um die Bedeutung von ‘‘Optimalität’’ in diesem Zusammenhang charakterisieren zu können, werden geeignete Bewertungskriterien benötigt, die unser Ziel der Schwankungsminimierung im Gesamtnetz beschreiben.

3.1.2 Performance-Werte

Um die Qualität von Lösungen zu beurteilen, führen wir verschiedene Performance-Werte ein, die bereits in Kapitel 4.2. von [1] bzw. in [14] zur Anwendung kamen. Das Ziel unserer Optimierung ist die Minimierung von Schwankungen in der durchschnittlichen Stromnachfrage. Dabei können verschiedene ‘‘Schwankungseigenschaften’’ betrachtet werden. Im Folgenden beurteilen wir Closed-Loop Lösungen $\mathbf{z}^{\mu^*}(0; \mathcal{N}) \in \mathbb{D}^{0,N}$ sowie Open-Loop Lösungen $\mathbf{z}^*(k; N) \in \mathbb{D}^{k,N}$ zum Zeitpunkt $k \in \mathbb{N}_N$.

Peak-to-Peak (PTP)

Der erste Ansatz ist die Minimierung der Amplitude, also der Differenz des maximalen und minimalen Ausschlags. Dieses Kriterium wird durch den *Peak-to-Peak*-Wert gemessen. Die Idee hinter diesem Performance-Wert ist, dass Netzbetreiber gerade mit großen Unterschieden in der Stromnachfrage Schwierigkeiten haben.

$$PTP(\mathbf{z}^{\mu^*}) := \max_{k=0,\dots,N-1} \hat{z}^{\mu^*}(k) - \min_{k=0,\dots,N-1} \hat{z}^{\mu^*}(k) \quad (3.8)$$

$$PTP_k(\mathbf{z}^*) := \max_{j=k,\dots,N+k-1} \hat{z}^*(j) - \min_{j=k,\dots,k+N-1} \hat{z}^*(j) \quad (3.9)$$

Würden wir jedoch nach diesem Kriterium optimieren, könnten Lösungen mit stark oszillierender Gesamtnachfrage entstehen. Ebenso wären Resultate mit sehr hoher Grundnachfrage möglich. Um das zu verhindern, führen wir einen Performance-Wert ein, für den sowohl die Minimierung der Amplitude als auch das stabile Aufrechterhalten einer sinnvollen Stromnachfrage eine Rolle spielen.

Mean-Quadratic-Deviation (MQD)

Das nächsten Kriterium bestraft die Abweichung von der durchschnittlichen Nachfrage, die ohne Einsatz von Batterien herrschen würde. Diese ist für die Closed- bzw. Open-Loop Anwendung jeweils gegeben durch

$$\hat{\zeta} := \frac{1}{\mathcal{I}\mathcal{N}} \sum_{i=1}^{\mathcal{I}} \sum_{j=0}^{\mathcal{N}-1} w_i(j) - g_i(j) \quad (3.10)$$

$$\hat{\zeta}(k) := \frac{1}{\mathcal{I}\mathcal{N}} \sum_{i=1}^{\mathcal{I}} \sum_{j=k}^{k+\mathcal{N}-1} w_i(j) - g_i(j). \quad (3.11)$$

Um besonders große Amplituden zu vermeiden, gewichten wir die Abweichung quadratisch. Die Mean-Quadratic-Deviation ergibt sich dementsprechend als

$$MQD(\mathbf{z}^{\mu^*}) := \frac{1}{\mathcal{N}} \left\| \hat{\mathbf{z}}^{\mu}(0; \mathcal{N}) - \hat{\zeta} \right\|^2 \quad (3.12)$$

$$MQD_k(\mathbf{z}^*) := \frac{1}{\mathcal{N}} \left\| \hat{\mathbf{z}}^*(k; \mathcal{N}) - \hat{\zeta}(k) \right\|^2, \quad (3.13)$$

wobei $\|\cdot\|$ für die euklidische Norm steht. Es wird also eine möglichst konstante Grundnachfrage angestrebt, die abhängig von Stromverbrauch und -erzeugung höher oder niedriger ausfallen kann. Außerdem berücksichtigt die Durchschnittsnachfrage bei geeigneter Wahl des MPC Horizonts \mathcal{N} auch die natürlichen Schwankungen innerhalb eines Tages.

Average-Smoothing-Factors (ASF)

Ein weiteres Bewertungskriterium greift die Oszillation von Lösungen an. Der ASF bestraft die Abweichung von einer Nachfrage zu der vom vorherigen Zeitpunkt quadratisch:

$$ASF(\mathbf{z}^{\mu^*}) := \frac{1}{\mathcal{N}-1} \sum_{k=0}^{\mathcal{N}-2} (\hat{z}^{\mu^*}(k+1) - \hat{z}^{\mu^*}(k))^2 \quad (3.14)$$

$$ASF_k(\mathbf{z}^*) := \frac{1}{\mathcal{N}-1} \sum_{j=k}^{\mathcal{N}-2} (\hat{z}^*(j+1) - \hat{z}^*(j))^2. \quad (3.15)$$

Allerdings bestraft dieses Bewertungskriterium keine sehr hohen oder niedrigen Nachfragen.

Bemerkung 3.1.4. Da der MQD Performance-Wert die meisten Optimierungsziele erfüllt und in der Implementierung gut umzusetzen ist, legen wir die Minimierung dieses

Wertes für den Rest der Arbeit als Ziel fest. Es lassen sich jedoch auch die anderen beiden Performance-Werte oder eine Kombination aus mehreren für die Optimierung einsetzen, wie es in Abschnitt 4.2 [1] beschrieben wird.

3.2 Centralized Model Predictive Control (CMPC)

Der erste Ansatz zur modellprädiktiven Regelung des Smart Grids ist ein global gesteuertes System. Das bedeutet, dass jedes RES alle Informationen mit einer *Central Entity*⁵ (kurz: CE) teilt, welche die optimalen Steuerungen für das komplette System berechnet. Das restliche Vorgehen entspricht dem ökonomischen NMPC aus Abschnitt 2.2 mit dem Ziel, die Stromnachfrage zu $\hat{\zeta}(k)$ zu steuern.

Zu einem Zeitpunkt $k \in \mathbb{N}_N$ senden alle beteiligten RES ihren aktuellen Batteriestand an die CE. Diese bestimmt dann die optimale Nachfrage $\mathbf{z}^* \in \mathbb{D}^{k,N}(x_0)$ und damit die optimalen Steuerungen für die nächsten $N \in \mathbb{N}$ Zeitpunkte. Für die Stufenkosten des CMPC richten wir uns nach dem MQD Performance-Wert, d.h. der Abstand zur durchschnittlichen Stromnachfrage wird quadratisch bestraft. Ein Vektor von Stromnachfragen $\mathbf{z} \in \mathbb{R}^{\mathcal{I}}$ erzeugt somit in der k -ten Stufe die Kosten

$$l_k(\mathbf{z}) := \left(\frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} z_i - \hat{\zeta}(k) \right)^2 = \left(\hat{z} - \hat{\zeta}(k) \right)^2.$$

Die Gesamtkostenfunktion für das CMPC in Stufe k ergibt sich dann als

$$J_N(x(k), \mathbf{u}(k; N)) = \sum_{j=0}^{N-1} \left(\hat{z}(j) - \hat{\zeta}(k) \right)^2,$$

wobei $\mathbf{z} \in \mathbb{D}^N$ die Trajektorie der Stromnachfragen darstellt, welche beim Startzustand $x(k) \in \mathbb{X}$ und der gewählten Regelung $\mathbf{u}(k; N) \in \mathbb{U}^N$ entsteht. Das heißt, dass $z(j)$ der Stromnachfrage aller RES zum Zeitpunkt $k+j$ entspricht. Die Funktionsweise des CMPC ist in Algorithmus 2 dargestellt.

3.3 Distributed Model Predictive Control (DMPC)

Trotz guter Ergebnisse im Bezug auf den erreichbaren Performance Wert hat der CMPC einige große Nachteile. Einerseits kann das in jeder Iteration zu lösende Optimierungsproblem (3.16) sehr groß werden: Schon bei 100 Haushalten, einem Diskretisierungszeitraum von 30 Minuten und einem MPC-Horizont von einem Tag erhält man insgesamt über 9000 Kontroll- und Zustandsvariablen. Da es sich beim vorliegenden Problem jedoch um ein Online-Problem handelt und es demnach während der Laufzeit gelöst werden muss, sollte die Berechnung einer optimalen Steuerung in möglichst kurzer Zeit erfolgen. Andererseits muss jeder Haushalt seinen derzeitigen Batteriestand sowie die erwarteten Stromverbrauchs- und Erzeugungswerte jederzeit an die

⁵Aus dem Englischen: *Zentrale Einheit*

CE schicken. Dies führt neben dem aufwändigen Informationsaustausch auch zu diversen datenschutzrechtlichen Problemen, weshalb sich unter Umständen nur wenige Teilnehmer an solch einem System finden lassen.

Algorithmus 2 Centralized Model Predictive Control

Sei $\mathcal{N} \in \mathbb{N} \cup \{\infty\}$ die Anzahl an Optimierungsschritten und $N \in \mathbb{N}$ der MPC Horizont. Setze $k = 0$.

RES $i, i \in \mathbb{N}_{\mathcal{I}}$

1. Bestimme den aktuellen Zustand $x_i(k) \in \mathbb{X}_i$, setze $x_{i,0} = x_i(k)$ und liefere Vorhersagen zu Stromverbrauch bzw. -erzeugung $\mathbf{w}_i(k; N)$, $\mathbf{g}_i(k; N)$.
2. Schicke $x_{i,0}$, $\mathbf{w}_i(k; N)$ und $\mathbf{g}_i(k; N)$ an die CE.

CE

3. Berechne die durchschnittliche Stromnachfrage

$$\hat{\zeta} = \frac{1}{N}(\hat{\mathbf{w}}(k; N) - \hat{\mathbf{g}}(k; N))\mathbf{1}^T$$

und bestimme die Stufenkosten l^k und damit die Kostenfunktion J_N .

4. Löse das Optimierungsproblem

$$\mathbf{u}^*(k; N) \in \arg \min_{\mathbf{u}(k; N) \in \mathbb{U}^{k; N}(x_0)} J_N(x_0, \mathbf{u}(k; N)) \quad (3.16)$$

und erhalte optimale Open-Loop Kontrollen $\mathbf{u}_i^*(k; N)$ für jedes $i \in \mathbb{N}_{\mathcal{I}}$.

5. Setze das Feedback $\mu_i^*(x(k)) := \mathbf{u}_i^*(k)$ und schicke $\mu_i^*(x(k))$ an das jeweilige RES.

Setze $k = k + 1$ und rücke damit den Horizont nach vorne. Falls $k \leq \mathcal{N}$ gehe zu Schritt 1. Sonst gib die Feedback-Regelung μ^* zurück.

Um diese beiden großen Probleme anzugehen, wurden in [1] mehrere andere Algorithmen vorgestellt. Darunter ein komplett dezentral arbeitender Algorithmus, in dem alle RES unabhängig von einer zentralen Einheit ihre Steuerungen berechnen. Da dieser Algorithmus allerdings nicht so gute Performance Werte erzielt, wurde eine ‘‘Kompromissform’’ dieser beiden eingeführt: Der *Distributed*⁶ MPC (kurz: DMPC). Mithilfe einer geschickt implementierten Kommunikationsstruktur zwischen den einzelnen RES und einer CE sowie einer dezentralen (dadurch auch parallelisierbaren) Optimierung gelingt es, die obigen Schwierigkeiten zu bewältigen.

Bemerkung 3.3.1. Neben dem eben genannten hierarchischem DMPC führt Dr. Braun in seiner Dissertation [1] zwei weitere Algorithmen-Typen ein: In Kapitel 6 und 7 der Doktorarbeit werden Vorgehen entwickelt, die auf der *dual gradient* Methode bzw. auf dualer Dekomposition basieren. Wir beschränken uns in dieser Arbeit jedoch auf den

⁶Aus dem Englischen: *verteilt*

in Kapitel 5 von [1] beschriebenen DMPC.

Da sich das Vorgehen des DMPCs bis auf die Berechnung der Open-Loop Lösung nicht vom CMPC unterscheidet, betrachten wir das Optimierungsproblem (3.16) aus Algorithmus 2 zu einem festen Zeitpunkt k . Das zu lösende Problem lässt sich sehr vereinfacht darstellen als

$$\begin{aligned} & \min G(\mathbf{z}) \\ & \text{s.t. } \mathbf{z} \in \mathbb{D}^{\mathcal{I} \times N} \end{aligned}$$

für eine Funktion $G : \mathbb{D}^{\mathcal{I} \times N} \rightarrow \mathbb{R}$.

Damit jedes RES selbstständig arbeiten kann, führen wir lokale Kostenfunktionen ein, die die Steuerungen anderer RES als gegeben annehmen. Für jedes RES $i \in \mathbb{N}_{\mathcal{I}}$ definieren wir daher den Vektor aller Stromnachfragen ohne die eigene als

$$\mathbf{p}_i := (\mathbf{z}_1^T \dots \mathbf{z}_{i-1}^T \mathbf{z}_{i+1}^T \dots \mathbf{z}_{\mathcal{I}}^T) \in \mathbb{R}^{(\mathcal{I}-1) \times N}. \quad (3.17)$$

Damit lassen sich die lokalen Kostenfunktionen bei gegebenem \mathbf{p}_i allein abhängig von der Stromnachfrage des RES i als

$$g_i(\cdot; \mathbf{p}_i) : \mathbb{R}^N \rightarrow \mathbb{R}, \quad g_i(\mathbf{z}_i; \mathbf{p}_i) := G(\mathbf{z})$$

aufstellen.

In Kapitel 5 von [1] werden mehrere Varianten des verteilten MPCs vorgestellt. Die Algorithmen zeichnen sich dadurch aus, dass sie eine spezielle Kommunikationsstruktur zwischen den RES und einer CE nutzen, um in einer zusätzlichen Schleife innerhalb eines Zeitschritts der eigentlichen MPC Routine iterativ eine optimale Lösung zu finden. Dabei wechseln sich 2 Phasen ab: Die RES berechnen mithilfe der obigen Kostenfunktionen eine optimale Steuerung und senden diese an die CE. Die CE muss nun die einzelnen Steuerungen optimal verbinden. Dafür berechnet sie die Konvexkombination aus der bisher besten Steuerung des kompletten Systems \mathbf{z} und den neuen Steuerungen der Einzelhaushalte \mathbf{z}^* , die den besten MQD-Wert besitzt. Dies geschieht durch die Berechnung einer optimalen *Schrittweite*

$$\arg \min_{\theta \in [0,1]} G(\theta \mathbf{z}^* + (1 - \theta) \mathbf{z}), \quad (3.18)$$

welche anschließend an die RES gesendet wird. Durch dieses Vorgehen wird die Lösung aufgrund der Konvexität von G und \mathbb{D} in jedem Durchlauf verbessert. In Kapitel 5.2.2 von [1] ist genauer beschrieben, wie sich die Lösung des DMPC so allmählich der des CMPC annähert. Die Kommunikationsstruktur des DMPC ist in Abbildung 3.2 dargestellt.

Da wir auch nichtlineare Systeme wie in Bemerkung 3.1.3 beschrieben zulassen wollen, fokussieren wir uns auf den in Kapitel 5.4 der Dissertation [1] vorgestellten Algorithmus 3. Sind die Anzahl der Haushalte \mathcal{I} sowie der MPC-Horizont N festgelegt, benötigt der Algorithmus noch eine Maximalanzahl der inneren Schleifendurchläufe

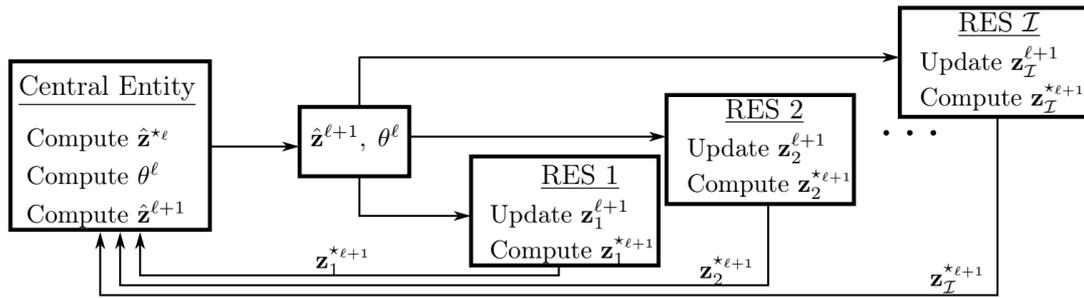


Abbildung 3.2: Kommunikationsstruktur des DMPC, Kapitel 5.3 [1]

$l_{\max} \in \mathbb{N} \cup \{\infty\}$ sowie einige Parameter $\varepsilon_1, \varepsilon_2, \delta > 0$, die die Genauigkeit der Optimierung bestimmen und den Abbruch der inneren Schleife regeln, wie später erklärt wird.

In der Hauptschleife werden nun die oben genannten zwei Phasen abwechselnd durchgeführt. Zuerst erhält die CE die von allen RES $i \in \mathbb{N}_{\mathcal{I}}$ berechneten Kontrollen $z_i^{*\ell}$ der l -ten Iteration sowie die Binärvariablen $\nu_i^l \in \{0, 1\}$, die als Indikator für die Zulässigkeit der bisher geplanten Kontrollen z_i^l dienen. Dieser Indikator ist gleich 1, falls die zugehörige Steuerung für RES i zulässig ist und 0 falls sie unzulässig ist.

Ausgehend von diesen Daten wird nun mittels (3.18) die optimale Schrittweite θ^l berechnet, die angibt, wie stark die RES ihre bisherige Steuerung an die neu berechnete Steuerung anpassen sollen. Mittels dieser Schrittweite wird dann die neue Steuerung z_i^{l+1} aus der bisherigen und den von den einzelnen RES berechneten Kontrollen bestimmt und anschließend mit der globalen Kostenfunktion bewertet. Abhängig von den gewählten Genauigkeitsparametern wird die Schleife abgebrochen oder die Schrittweite und die neue Kontrolle an die RES gesendet.

Danach sind die RES an der Reihe. Jeder Einzelne führt eine Aktualisierung seiner Steuerungs- und Kommunikationsvariablen durch, prüft, ob die von der CE übertragene Kontrolle zulässig ist und setzt den Indikator ν_i^{l+1} auf 0 bzw. 1. Anschließend wird eine neue Steuerung $z_i^{*\ell+1}$ berechnet, welche die lokale Kostenfunktion g_i unter Einbezug der geplanten Steuerungen aller RES minimiert. Abschließend werden die neuen Kontrollen an die CE geschickt und der Iterationszähler l wird inkrementiert. Danach beginnt die erste Phase von neuem.

Der Algorithmus bricht unter verschiedenen Umständen ab: Sollte die Performance-Verbesserung, also der Unterschied der globalen Kostenfunktion zweier aufeinanderfolgender Schritte, kleiner sein als ein $\varepsilon_1 > 0$ und ist die aktuell geplante Kontrolle zulässig, so endet die innerer Schleife mit einer ausreichend guten Lösung. Sollte die geplante Steuerung eines RES unzulässig sein, die von den RES berechnete Kontrolle jedoch schon einen Performance-Wert erzeugen, der sich um nicht mehr als $\varepsilon_2 > 0$ von dem des vorherigen Schritts unterscheidet, so wird ebenfalls abgebrochen. Ein weiterer Terminierungsgrund ist, wenn die von der CE an die RES gesendete Schrittweite kleiner als ein $\delta > 0$ ist.

Hierbei fällt ein entscheidender Unterschied zum CMPC auf: Die berechneten Lösungen können im Falle von nichtlinearen Nebenbedingungen unzulässig werden. Dies liegt daran, dass die Konvexität des Zustandsraums nicht mehr zwangsläufig gewährleistet sein muss, die Steuerungen jedoch mittels Konvexkombinationen erzeugt werden (vgl. Kap. 5.4 [1]).

Algorithmus 3 Hierarchische verteilte Optimierung von Smart Grids mit ggf. nichtlinearer Dynamik (DMPC)

Input: Lege die Anzahl der RES $\mathcal{I} \in \mathbb{N}$, den MPC-Horizont $N \in \mathbb{N}$, die maximale Iterationszahl $l_{\max} \in \mathbb{N} \cup \{\infty\}$ und die gewünschten Genauigkeiten $\varepsilon_1, \varepsilon_2, \delta > 0$ fest. Setze die zulässigen Mengen $\mathbb{D}_i^{k,N}(x_i(k))$ abhängig von den Startzuständen $x_i(k) \in \mathbb{X}_i$ und den zeitabhängigen Parametern $\mathbf{g}_i(k; N)$, $\mathbf{w}_i(k; N)$. Setze $l = 0$.

Hauptschleife:

Phase 1 (CE): Erhalte \mathbf{z}_i^{*l} und $\nu_i^l, i \in \mathbb{N}_{\mathcal{I}}$.

1. Berechne die Schrittweite $\theta^l = \arg \min_{\theta \in [0,1]} G(\theta \mathbf{z}^{*l} + (1 - \theta) \mathbf{z}^l)$.
2. Berechne $\mathbf{z}^{l+1} = \theta^l \mathbf{z}^{*l} + (1 - \theta^l) \mathbf{z}^l$ und bestimme die Performance-Werte $G^{*l} = G(\mathbf{z}^{*l})$ und $G^{l+1} = G(\mathbf{z}^{l+1})$.
3. Falls $|G^{l+1} - G^l| < \varepsilon_1$,
 - Wenn $\nu_i^l = 1$ für alle $i \in \mathbb{N}_{\mathcal{I}}$, gib θ^l zurück und beende den Algorithmus.
 - Wenn es ein $i \in \mathbb{N}_{\mathcal{I}}$ mit $\nu_i^l = 0$ gibt und $|G^{*l} - G^l| < \varepsilon_2$, gib θ^l zurück und beende den Algorithmus.
 - Sonst übertrage θ^l und \mathbf{z}^{l+1} an alle RES.
4. Falls $l = l_{\max}$ oder $\theta^l < \delta$,
 - Wenn $\nu_i^l = 1$ für alle $i \in \mathbb{N}_{\mathcal{I}}$, gib θ^l zurück und beende den Algorithmus.
 - Wenn es ein $i \in \mathbb{N}_{\mathcal{I}}$ mit $\nu_i^l = 0$ gibt, gib $\theta^l = 1$ zurück und beende den Algorithmus.
5. Sonst übertrage θ^l und \mathbf{z}^{l+1} an alle RES.

Phase 2 (RES $i, i \in \mathbb{N}_{\mathcal{I}}$): Erhalte θ^l und \mathbf{z}^{l+1} .

6. Aktualisiere die Kontroll- und Kommunikationsvariablen, also

$$\mathbf{u}_i^{l+1} = \theta^l \mathbf{u}_i^{*l} + (1 - \theta^l) \mathbf{u}_i^l \quad \text{und} \quad \mathbf{z}_i^{l+1} = \theta^l \mathbf{z}_i^{*l} + (1 - \theta^l) \mathbf{z}_i^l.$$

7. Berechne \mathbf{x}_i^{l+1} basierend auf \mathbf{u}_i^{l+1} , d.h.:

$$x_i^{l+1}(j+1) = f_i(x_i^{l+1}(j), u_i^{l+1}(j)), \quad \forall j \in \{k, \dots, k + N - 1\}$$

8. Falls $\mathbf{x}_i^{l+1} \in \mathbb{X}_i^N$ (also \mathbf{x}_i^{l+1} zulässig ist), so setze $\nu_i^{l+1} = 1$. Sonst setze $\nu_i^{l+1} = 0$.
9. Setze \mathbf{p}_i^{l+1} wie in Gleichung (3.17).
10. Löse das Optimierungsproblem $\mathbf{z}_i^{*l+1} = \arg \min_{\mathbf{z}_i \in \mathbb{D}_i} g_i(\mathbf{z}_i; \mathbf{p}_i^{l+1})$.
11. Übermittle \mathbf{z}_i^{*l+1} und ν_i^{l+1} an die CE.

Setze $l = l + 1$, gehe zu Phase 1.

Verbrauchsschätzungen

Bevor wir uns mit den Ergebnissen der Modellerweiterung um Elektroautos auseinandersetzen, wird im folgenden Kapitel erläutert, wieso genaue Schätzungen des Stromverbrauchs für die Optimierung essentiell sind und wie damit umgegangen werden kann, dass Prognosen für Einzelhaushalte in der Realität nicht präzise genug sind. Wir werden zeigen, dass eine Vorhersage für den Durchschnitts-Verbrauch in unserem Fall völlig ausreichend ist und selbst bei ungenauen Prognosen zu guten Ergebnissen führt.

4.1 Problematik und Auswege

Eine der wichtigsten Voraussetzungen modellprädiktiver Regelung ist, dass zu jedem Zeitpunkt bis zum Ende des MPC Horizonts sinnvolle und möglichst genaue Vorhersagen über den Umweltzustand gemacht werden können. Die beiden Größen, die in unserem Modell vorhergesagt werden müssen, sind der Verbrauch und die Erzeugung von Strom. Dabei beschränken wir uns in der Regel auf den Zeitraum eines Tages. Die Menge an erzeugtem Strom hängt neben der (feststehenden) Größe und Beschaffenheit der Anlage hauptsächlich von der Temperatur sowie Dauer und Kraft der Sonneneinstrahlung ab. Eine genaue Wettervorhersage ist also ausreichend, um den, durch jede einzelne Photovoltaik-Anlage erzeugten, Strom prognostizieren zu können. Wie z.B. in [15] berichtet, gibt es inzwischen viele Methoden um Prognosen zu entwickeln.

Die Verbrauchsschätzung ist allerdings komplizierter. Da jeder Haushalt Entscheidungen trifft, die den Stromverbrauch z.B. durch den Einsatz elektrischer Großgeräte (wie Waschmaschinen oder Herde) massiv beeinflussen können, ist die genaue Schätzung eines Individualverbrauchs sehr schwierig. Eine Methode damit umzugehen, besteht darin, den so genannten *kontrollierbaren Verbrauch* (engl. *controllable loads*) zu nutzen. Dieses Prinzip wurde bereits in einigen Studien betrachtet und auch in [1] vorgestellt. Kontrollierbarer Verbrauch ist jeglicher Stromverbrauch, der nicht zu einem festen Zeitpunkt stattfinden muss, sondern planbar ist. In [16] wurde festgestellt, dass bis zu 60 % des Verbrauchs eines Haushalts kontrollierbar sind. Dabei handelt es sich meist um den Strom für elektrische Großgeräte wie Kühlschränke oder Klimaanlage. Wie diese Möglichkeit zur Verringerung der Stromschwankungen im DMPC-Algorithmus genutzt werden kann, wird in [1] beschrieben und in dieser Arbeit nicht

weiter behandelt. Der hier behandelte Ansatz wird ein anderer sein.

Mit dem Problem, dass Verbrauchsschätzungen einzelner Haushalte sehr ungenau sind, beschäftigten sich z.B. Samuel Humeau et al. im Jahr 2013 in [17]. Um die Güte unterschiedlicher Algorithmen zu bewerten, berechneten sie unter anderem den *Normalized Root Mean Square Error* (NRMSE) der Prognosen. Für eine Folge von Verbräuchen $w = (w(1), \dots, w(N))$ und deren Vorhersage $\tilde{w} = (\tilde{w}(1), \dots, \tilde{w}(N))$ für ein $N \in \mathbb{N}$ ist dieser gegeben als

$$NRMSE(\tilde{w}) = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^N (w(i) - \tilde{w}(i))^2}}{\sqrt{\frac{1}{N} \sum_{i=1}^N w(i)^2}}.$$

Die Autoren verglichen Prognosemethoden für Einzelhaushalte und Distrikte bei einem Vorhersagezeitraum von 24 Stunden. Der durchschnittliche *NRMSE* bei allen getesteten Algorithmen lag bei der Prognose von Individualverbräuchen bei über 0.6, wobei eine einfache lineare Regression das beste Ergebnis lieferte. Im Vergleich dazu war bei einer Vorhersage für einen Distrikt aus 782 Haushalten mittels einer *Support Vector Machine Regression* ein Wert von 0.081 erreichbar. Demnach ist das Schätzen des gemittelten Verbrauchs wesentlich genauer möglich als das Prognostizieren des Verbrauchs eines einzelnen Haushalts.

Dieses Wissen wollen wir nutzen, um die Realisierbarkeit des DMPC-Algorithmus zu verbessern. Unsere Herangehensweise wird sein, dass wir komplett auf die Schätzung des Individualverbrauchs verzichten und stattdessen gemittelte Verbrauchsschätzungen verwenden. Im folgenden Abschnitt werden wir zeigen, dass diese Prognosewerte für die Minimierung von Stromschwankungen im Smart Grid mittels DMPC völlig ausreichend sind.

4.2 Einsatz gemittelter Verbrauchsschätzungen

Die Zukunftsinformationen über den Verbrauch spielen in der Smart Grid Regelung eine entscheidende Rolle. Die in 3.1 definierte Kostenfunktion bestraft im Zeitpunkt $k \in \mathbb{N}$ die Abweichung von der erwarteten Durchschnittsnachfrage $\hat{\zeta}(k)$ über die nächsten N Zeitschritte. Diese setzt sich aus dem erwarteten Stromverbrauch w_i und der erwarteten Stromerzeugung g_i der einzelnen Haushalte $i \in \mathbb{N}_{\mathcal{I}}$ für den MPC Horizont $N \in \mathbb{N}$ wie folgt zusammen:

$$\hat{\zeta}(k) = \frac{1}{\mathcal{I}N} \sum_{i=1}^{\mathcal{I}} \sum_{j=k}^{k+N-1} w_i(j) - g_i(j). \quad (4.1)$$

Um zu zeigen, dass mit gemittelten Verbrauchsschätzungen Ergebnisse genauso gute Performance-Werten erreichbar sind wie mit perfekten Verbrauchsinformationen, müssen wir die Verbrauchswerte durch gemittelte Werte ersetzen. Das bedeutet, dass

$$w^\varnothing(j) := \frac{1}{\mathcal{I}} \sum_{m=1}^{\mathcal{I}} w_m(j) \quad (4.2)$$

nun den Platz der $w_i(j)$ einnimmt. Wir können zeigen, dass diese Anpassung nichts an der erwarteten Durchschnittsnachfrage ändert. Dies liegt daran, dass $\hat{\zeta}(k)$ ein Wert ist, der bereits über die Haushalte gemittelt wurde. Im Folgenden werden alle Variablen und Funktionen, die für die Optimierung mit gemittelten Werten gesondert betrachtet werden, mit einem \cdot^\varnothing markiert.

Lemma 4.2.1. *Das Ersetzen von w_i durch w^\varnothing für alle $i \in \mathbb{N}_{\mathcal{I}}$ hat keine Auswirkung auf den Wert von $\hat{\zeta}(k)$.*

Beweis.

$$\begin{aligned} \hat{\zeta}^\varnothing(k) &:= \frac{1}{\mathcal{I}N} \sum_{i=1}^{\mathcal{I}} \sum_{j=k}^{k+N-1} w^\varnothing(j) - g_i(j) \\ &= \frac{1}{\mathcal{I}N} \sum_{i=1}^{\mathcal{I}} \sum_{j=k}^{k+N-1} \frac{1}{\mathcal{I}} \left(\sum_{m=1}^{\mathcal{I}} w_m(j) \right) - g_i(j) \\ &= \frac{1}{\mathcal{I}N} \sum_{j=k}^{k+N-1} \left(\sum_{i=1}^{\mathcal{I}} \frac{1}{\mathcal{I}} \sum_{m=1}^{\mathcal{I}} w_m(j) \right) - \left(\sum_{i=1}^{\mathcal{I}} g_i(j) \right) \\ &= \frac{1}{\mathcal{I}N} \sum_{j=k}^{k+N-1} \left(\sum_{m=1}^{\mathcal{I}} w_m(j) \right) - \left(\sum_{i=1}^{\mathcal{I}} g_i(j) \right) \\ &= \frac{1}{\mathcal{I}N} \sum_{j=k}^{k+N-1} \sum_{i=1}^{\mathcal{I}} w_i(j) - g_i(j) = \hat{\zeta}(k). \end{aligned}$$

□

Da der Performance-Wert einer zulässigen Regelung allein vom Abstand zur erwarteten Durchschnittsnachfrage abhängt, ergibt sich im neuen Szenario auch keine andere Bewertung.

Lemma 4.2.2. *Der Wert der Kostenfunktion einer zulässigen Nachfragefolge $\mathbf{z} \in \mathbb{D}$ wird durch die aggregierten Verbrauchsschätzungen nicht verändert:*

$$G^\varnothing(\mathbf{z}) := \left\| \hat{\mathbf{z}} - \hat{\zeta}^\varnothing \right\|^2 = \left\| \hat{\mathbf{z}} - \hat{\zeta} \right\|^2 = G(\mathbf{z}). \quad (4.3)$$

In Abschnitt 3.3 wurde angesprochen, dass die Performance-Werte der Lösungen des DMPCs in der inneren Optimierungsschleife gegen den Performance-Wert der Lösung des CMPCs konvergieren. Die einzige im Beweis relevante Größe, die durch das Ersetzen der Individualprognosen durch aggregierte Vorhersagen direkt beeinflusst wird, ist die Menge der zulässigen Regelungen $\mathbb{D}^{k,N} = \mathbb{D}_1^{k,N} \times \dots \times \mathbb{D}_{\mathcal{I}}^{k,N}$. Damit die Konvergenz auch unter den neuen Bedingungen gegeben ist, muss die neue Menge $\mathbb{D}^{k,N,\varnothing} = \mathbb{D}_1^{k,N,\varnothing} \times \dots \times \mathbb{D}_{\mathcal{I}}^{k,N,\varnothing}$ ebenfalls konvex, kompakt und nicht leer sein. Dass diese Eigenschaften erfüllt sind, kann man an der Definition von $\mathbb{D}_i^{k,N,\varnothing}$ erkennen.

Lemma 4.2.3. Für das MPC-Verfahren mit Horizont $N \in \mathbb{N}$ sei in einem Zeitschritt $k \in \mathbb{N}$ für einen Haushalt $i \in \mathbb{N}_{\mathcal{I}}$ der Startzustand $x_{i,0} \in \mathbb{X}_i$ gegeben. Dann ergibt sich mit aggregierten Verbrauchsprognosen die Menge der zulässigen (open-loop) Regelungen als

$$\mathbb{D}_i^{k,N,\varnothing}(x_{i,0}) = \left\{ z_i^{\varnothing}(k; N) \in \mathbb{R}^N \left| \begin{array}{l} x_i(k) = x_{i,0} \\ x_i(j+1) = x_i(j) + Tu_i(j) \\ z_i^{\varnothing}(j) = w^{\varnothing}(j) - g_i(j) + u_i(j) \\ u_i(j) \in \mathbb{U}_i, \quad x_i(j) \in \mathbb{X}_i \\ \forall j = k, \dots, k+N-1 \end{array} \right. \right\}. \quad (4.4)$$

Damit erfüllt \mathbb{D}^{\varnothing} alle für die Konvergenz des DMPCs erforderlichen Eigenschaften.

Nun bleibt nur noch zu zeigen, dass die Lösung des CMPCs mit aggregierten Vorhersagen den gleichen Zielfunktionswert erreicht, wie die optimale Kontrolle mit Individualprognosen.

Satz 4.2.1. Seien $z^{\star} \in \mathbb{D}$ und $z^{\star\varnothing} \in \mathbb{D}^{\varnothing}$ die Ergebnisse des CMPC-Algorithmus mit perfekter Verbrauchsschätzung bzw. mit aggregierten Prognosen. Dann gilt:

$$G(z^{\star}) = G^{\varnothing}(z^{\star\varnothing}). \quad (4.5)$$

Beweis. Die Menge an zulässigen Regelungen \mathbb{U} wird durch die geänderte Verbrauchsschätzung nicht beeinflusst. Das liegt daran, dass nur die Kommunikationsvariable z von w abhängt, nicht aber die beschränkten Werte der Stromladung x und der Laderate u . Die Mengen der betrachteten Nachfragen \mathbb{D} bzw. \mathbb{D}^{\varnothing} unterscheiden sich allein in der Berechnung von z aus u , w und g .

In unserer Kostenfunktion kommt nur die über alle Haushalte gemittelte Nachfrage \hat{z} vor. Daher lässt sich Gleichung (4.5) nun wie folgt zeigen:

$$\begin{aligned} G^{\varnothing}(z^{\star\varnothing}) &\stackrel{\text{Lem. 4.2.2}}{=} G(z^{\star\varnothing}) = \min_{z^{\varnothing} \in \mathbb{D}^{\varnothing}} G(z^{\varnothing}) = \min_{z^{\varnothing} \in \mathbb{D}^{\varnothing}} \left\| \hat{z}^{\varnothing} - \hat{\zeta} \right\|^2 \\ &= \min_{z^{\varnothing} \in \mathbb{D}^{\varnothing}} \left\| \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} z_i^{\varnothing} - \hat{\zeta} \right\|^2 \\ &= \min_{u \in \mathbb{U}} \left\| \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} (w_i^{\varnothing} - g_i + u_i) - \hat{\zeta} \right\|^2 \\ &= \min_{u \in \mathbb{U}} \left\| \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} \left(\frac{1}{\mathcal{I}} \sum_{m=1}^{\mathcal{I}} w_m - g_i + u_i \right) - \hat{\zeta} \right\|^2 \\ &= \min_{u \in \mathbb{U}} \left\| \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} (w_i - g_i + u_i) - \hat{\zeta} \right\|^2 \\ &= \min_{z \in \mathbb{D}} \left\| \frac{1}{\mathcal{I}} \sum_{i=1}^{\mathcal{I}} z_i - \hat{\zeta} \right\|^2 = G(z^{\star}). \end{aligned}$$

□

Wir haben also gezeigt, dass der DMPC-Algorithmus mit aggregierten Verbrauchsschätzungen genauso gute Ergebnisse liefert wie mit perfekten Individualprognosen. Im folgenden Abschnitt werden wir dieses Ergebnis durch eine numerische Simulation validieren. Außerdem untersuchen wir, wie der DMPC mit fehlerhaften Verbrauchsprognosen umgeht.

4.3 Numerische Resultate

In diesem Abschnitt werden Simulationsergebnisse vorgestellt und analysiert. Dafür wird wie in Kapitel 6 ein Matlab Programm eingesetzt, welches im Anhang genauer vorgestellt wird. Als Verbrauchs- und Erzeugungsmengen verwenden wir die vom australischen Stromanbieter Ausgrid zur Verfügung gestellte Daten [19], die bereits in [1] als Grundlage der Simulation dienen.

4.3.1 Perfekte vs. aggregierte Verbrauchsprognosen

Dass das Aggregieren der Prognosen keine Auswirkung auf die Qualität der Lösung hat, wird durch folgende Simulation deutlich: Unser Smart Grid bestehe aus 50 Haushalten, jeweils mit 4.0 kWh großen Batterien ausgestattet, welche eine maximale Lade- und Entladerate von 2.5 bzw. -2.5 kW besitzen. Die Startladung aller Batterien beträgt 2.0 kWh. Die Optimierung mit dem DMPC-Algorithmus wird für eine Woche in halbstündigen Intervallen und dem MPC-Horizont von 24 Stunden durchgeführt. Dabei werden für die Verbrauchsvorhersagen in einem Fall perfekte Zukunftsinformationen und im anderen Fall aus den Realdaten aggregierte Werte verwendet.

In Tabelle 4.1 sind die in Kapitel 3.1.2 definierten Performance-Werte der beiden Varianten zu sehen. Wie aufgrund der theoretischen Resultate erwartet, unterscheiden sich die Ergebnisse nicht voneinander. In Abbildung 4.1 sind die durchschnittliche Stromnachfrage, Batterieladung und Laderaten abgebildet. Auf der linken Seite der Abbildung sehen wir 4 verschiedene Stromnachfragen. Die durchschnittliche, ungesteuerte Stromnachfrage LG^1 ist in rot dargestellt. Die grüne Linie entspricht dem laufenden Durchschnitt, der für jeden Zeitpunkt $k \in \mathbb{N}_N$ der ‘optimalen’ Nachfrage $\hat{\zeta}(k)$ entspricht, welche vom DMPC angestrebt wird. Die durchschnittliche Nachfrage, die sich aus der vom DMPC erzeugten Kontrollfolge ergibt, ist blau und die durchschnittliche Nachfrage der Lösung des DMPC nach dem Aggregieren der Verbrauchsinformationen türkis dargestellt. Auf der rechten Seite der Abbildung sind die durchschnittliche Batterieladungen sowie Laderaten der Lösungen des DMPC mit perfekten und aggregierten Prognosen zu sehen. Daraus wird ebenso wie aus der Tabelle deutlich, dass beide Varianten im Durchschnitt identische Kontrollfolgen liefern. Tatsächlich berechnet der DMPC in unserer Simulation in beiden Fällen Regelungen, die sich selbst für einzelne Haushalte erst ab der vierten Nachkommastelle unterscheiden. Diese Eigenschaft ist deshalb erstaunlich, da die Verbrauchsprognose für einen einzelnen Haushalt sehr ungenau ausfallen kann und ohne die verteilte Optimierung gegebenenfalls ein

¹Aus dem Englischen: *load - generation*, also der Verbrauch der Haushalte minus ihrer Stromerzeugung durch private Photovoltaikanlagen.

großer Fehler zu erwarten wäre.

Bemerkung 4.3.1. Um zu sehen, wie eine Lösung des DMPC aussieht, werden wir im Laufe der Arbeit häufig Abbildungen wie 4.1 nutzen.

Variante	PTP	MQD	ASF	Anzahl Iterationen
Ungesteuert	1.27980	0.05005	0.08578	-
Perfekt	0.08194	0.00052	0.00294	6.66
Aggregiert	0.08194	0.00052	0.00294	6.66

Tabelle 4.1: Performance-Werte und durchschnittlich notwendige innere Schleifeniterationen des DMPCs (mit Warmstart) für eine Genauigkeit von 10^{-5} im Vergleich

Ein Nachteil des DMPCs gegenüber des CMPCs ist die Notwendigkeit mehrerer innerer Optimierungsrunden pro Zeitschritt, um eine gewisse Güte der Lösung garantieren zu können. Diese Iterationen machen den größten Teil der Laufzeit des Algorithmus aus. In Kapitel 5.5 von [1] wird gezeigt, dass mit Hilfe des Warmstarts schon fünf Iterationen ausreichend sein können, um eine Lösung zu erhalten, die nahe an der des CMPCs ist. Selbst in unserem Setting benötigt DMPC für die innere Schleife im Durchschnitt nur 6.66 Iterationen für eine Genauigkeit von 10^{-5} . Diese Anzahl wird durch den Einsatz aggregierter Verbrauchsprognosen nicht erhöht. Dies spricht dafür, dass die vorgestellte Methode eine sehr gute Möglichkeit ist, um mit schlechten Individualprognosen umzugehen.

4.3.2 Gestörte, aggregierte Verbrauchsprognosen

Wie der *NRMSE* von 0.081 aus [17] zeigt, kann auch die Schätzung des aggregierten Verbrauchs fehlerbehaftet sein. Daher ist es für uns von Interesse, wie der DMPC-Algorithmus mit diesen Fehlern umgeht. Um das zu analysieren, werden wir zwei verschiedene Ansätze zur Prognosestörung betrachten. Der erste Ansatz besteht darin, alle Vorhersagedaten vor der Simulation mit Zufallswerten, welche abhängig von der betrachteten Störungsstärke unterschiedlich groß werden können, zu verändern. Im zweiten Fall werden wir diese gestörten Prognosen vor der Optimierung mittels eines Splines glätten.

In Abbildung 4.2 ist die durchschnittliche Stromnachfrage zu sehen, die entsteht, wenn die aggregierten Verbrauchsinformationen bis zu 5, 10, 15, 20 bzw. 30 Prozent von ihrem tatsächlichen Wert (nach oben oder nach unten) abweichen können. Die zugehörigen Performance-Werte sowie der durchschnittliche *NRMSE* dieser Prognosen sind in Tabelle 4.2 eingetragen. Die Prognosegenauigkeit der Vorhersage mit bis zu 15 % Abweichung von den realen Verbrauchsdaten ist demnach vergleichbar mit der tatsächlich erreichbaren Genauigkeit aus [17].

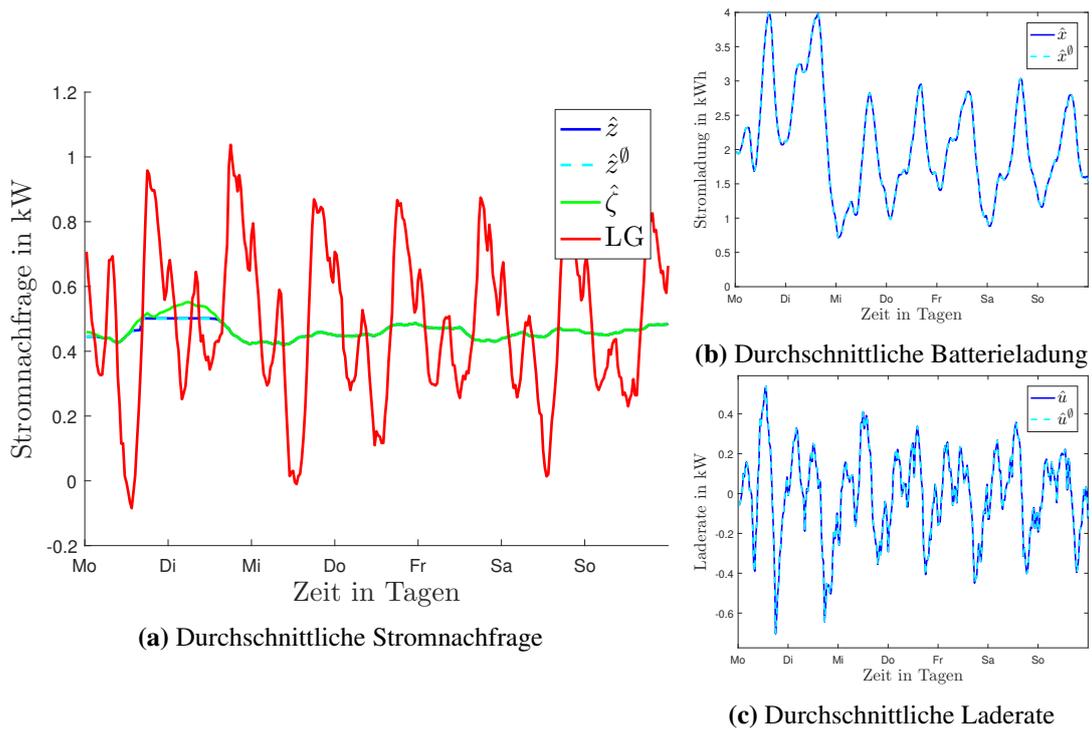


Abbildung 4.1: Simulation von 50 Haushalten über eine Woche mit perfekten Verbrauchsvorhersagen (blau) und aggregierten Prognosen (türkis)

Fehler	PTP	MQD	ASF	NRMS	Anzahl Iterationen
5 %	0.14778	0.00077	0.02201	0.028	6.83
10 %	0.21876	0.00150	0.04374	0.056	6.84
15 %	0.29555	0.00270	0.06550	0.084	6.87
20 %	0.37312	0.00437	0.08727	0.112	7.03
30 %	0.60742	0.00938	0.13745	0.175	7.52

Tabelle 4.2: Performance-Werte, Prognosegenauigkeiten und durchschnittlich notwendige innere Schleifeniterationen des DMPCs (mit Warmstart) für eine Genauigkeit von 10^{-5} für gestörte Prognosen

Da durch die Störung der Verbrauchsprognosen Fehler in der Optimierung entstehen, steigen die *PTP*-Werte mit zunehmendem Fehler. Auch die Werte von *MQD* werden größer, bleiben jedoch in einem im Vergleich zum ungesteuerten System akzeptablen Rahmen. Nur die *ASF*-Werte verschlechtern sich bei zunehmend ungenauer werdender Prognose so stark, dass bereits ab 20 % ein höherer Wert entsteht als im ungesteuerten System. Dies liegt daran, dass die durchschnittliche Änderung nicht in der Optimierung bewertet wird und diese zudem durch die fehlerhaften Verbrauchsprognosen stark beeinflusst wird. Da der Verbrauch zu jedem Zeitpunkt mit einem anderen Zufallswert verändert wird, oszilliert die gestörte Vorhersage sehr stark im Vergleich zur perfekten Prognose. Im schlimmsten Fall wird beispielsweise die Verbrauchsprognose

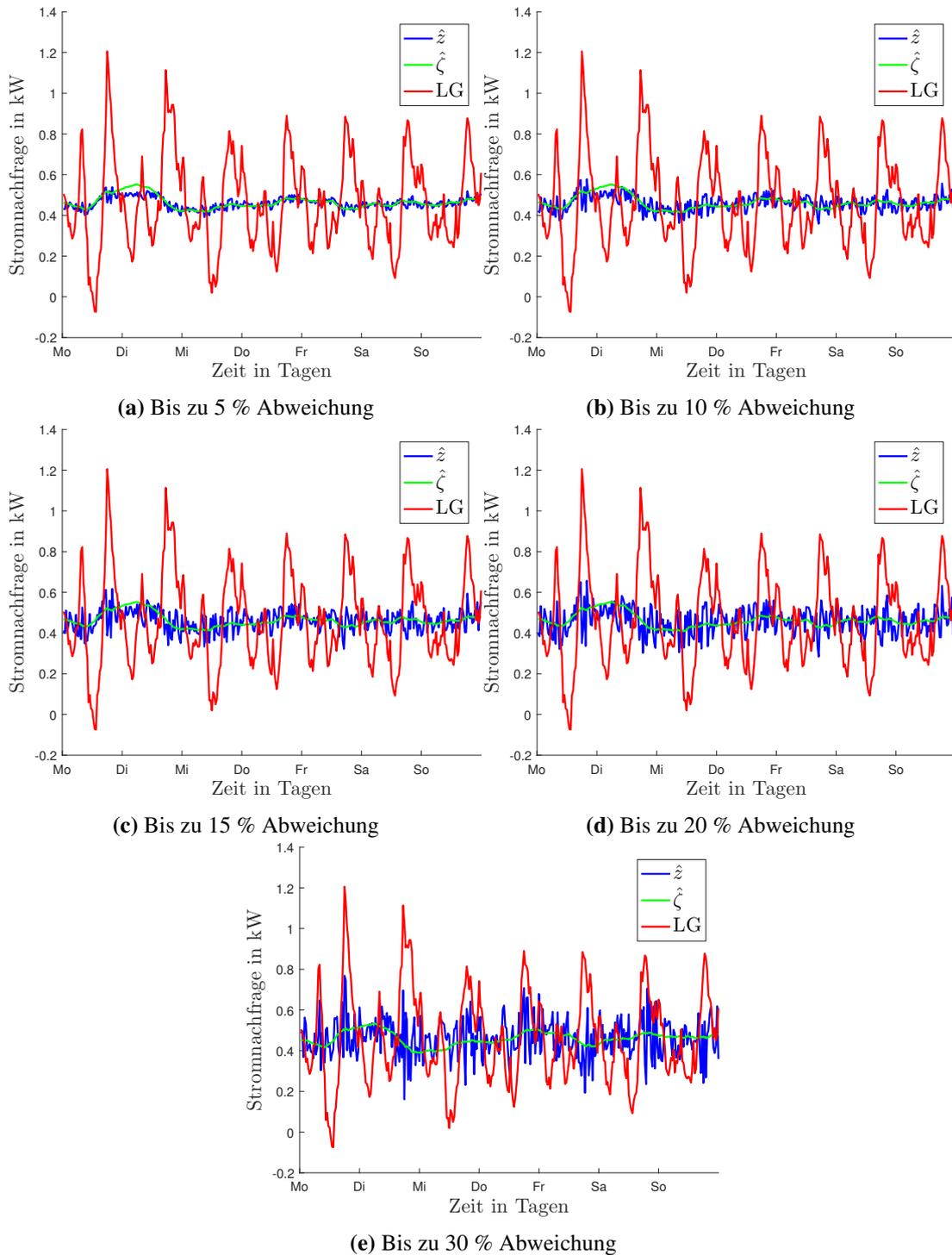
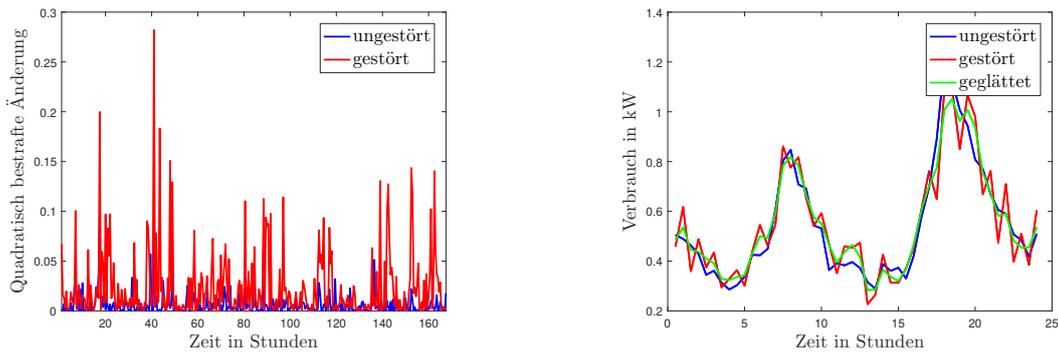


Abbildung 4.2: Durchschnittliche Stromnachfrage von 50 Haushalten über eine Woche mit gestörten, aggregierten Verbrauchsprognosen

zu einem Zeitpunkt k um 30 % vergrößert, aber im nachfolgenden Zeitpunkt um 30 % verringert. Der DMPC-Algorithmus liefert jedoch trotz dieser Schwierigkeiten im realistischen Szenario von bis zu 15-prozentigen Prognosefehlern noch akzeptable Ergebnisse.



(a) Quadratisch bestrafte Änderung über eine Woche

(b) Verbrauchsprognosen für einen Tag vor und nach Störung bzw. Glättung

Abbildung 4.3: Vergleich der ungestörten mit der einfach gestörten und geglätteten Prognose

Das Stören der Verbrauchsvorhersagen spiegelt sich kaum in der Anzahl der Iterationen wieder, die der DMPC-Algorithmus in seiner inneren Schleife benötigt. Lediglich bei einer Fehlerrate von bis zu 30 % benötigt der Algorithmus im Durchschnitt eine Iteration mehr, um ebenfalls eine Genauigkeit von 10^{-5} zu erlangen. Der Algorithmus muss sich also erst bei großen Prognosefehlern bemühen, eine gute Lösung zu finden.

4.3.3 Glätten der Verbrauchsprognosen mittels Splines

Um das oben genannte Problem des schlechten ASF-Werts zu umgehen, werden wir die Störung der Verbrauchsprognosen etwas erweitern. Echte Verbrauchsdaten sind im Vergleich zu unseren gestörten Prognosen relativ glatt. Um diesen Effekt darzustellen, betrachten wir die Originalprognose im Vergleich zu einer bis zu 30 Prozent gestörten Prognose. In Abbildung 4.3a sind die Abweichungen zum Verbrauch des vorherigen Zeitpunkts analog zum Bewertungskriterium ASF quadratisch bestraft dargestellt. Für eine Prognose \mathbf{w} ergibt sich dieser Performance-Wert als

$$ASF(\mathbf{w}) := \frac{1}{\mathcal{N}-1} \sum_{j=1}^{\mathcal{N}-2} (\hat{w}(j+1) - \hat{w}(j))^2.$$

Es fällt auf, dass die Abweichungen nach dem Stören der Prognosen viel drastischer ausfallen. Ignoriert man Stromerzeugung und Batteriesteuerungen und wendet die Berechnung des ASF-Werts auf die Prognosen an, so wird dieser durch das bis zu 30 prozentige Stören fast verdoppelt (siehe Tabelle 4.3, nicht geglättet).

Um diesen Effekt, den unsere Art der Störungsberechnung mit sich bringt, zu verringern, verwenden wir einen Spline². Diesen legen wir mit dem Ziel, die entstehende Kurve zu glätten, über die gestörten Werte. Mit dieser Methode können wir den

²Wir verwenden hierfür Matlabs *Smoothing Spline*. Dabei wird ein Spline s gesucht, welcher anhand der gegebenen Daten x einen Term der Form $p \sum_i (y_i - s(x_i))^2 + (1-p) \int \left(\frac{d^2 s}{dx^2} \right)^2 dx$ minimiert. Für mehr Informationen dazu siehe [18]

Geglättet	0 %	5 %	10 %	15 %	20 %	30 %
Nein	0.08452	0.08950	0.09984	0.11410	0.13101	0.16245
Ja	-	0.06443	0.06853	0.06920	0.07352	0.08436

Tabelle 4.3: ASF der Verbrauchsprognosen bei steigendem Fehler vor und nach der Glättung mittels Splines

Fehler	PTP	MQD	ASF	NRMS	Anzahl Iterationen
5 %	0.19778	0.00110	0.03761	0.042	6.65
10 %	0.20387	0.00138	0.03990	0.051	6.74
15 %	0.27153	0.00208	0.04564	0.071	6.65
20 %	0.31874	0.00292	0.05152	0.088	6.65
30 %	0.47882	0.00493	0.07011	0.126	7.27

Tabelle 4.4: Performance-Werte, Prognosegenauigkeiten und durchschnittlich notwendige innere Schleifeniterationen des DMPCs (mit Warmstart) für eine Genauigkeit von 10^{-5} bei gestörten und danach geglätteten Prognosen

ASF-Wert der Prognose reduzieren und erhalten so eine vernünftige Vorhersage, die trotzdem von der Originalprognose abweicht. In Abbildung 4.3b sind die drei unterschiedlichen Prognosen dargestellt. Die Kurve, die wir nach der Glättung mittels eines Splines erhalten, ähnelt der Originalkurve in ihrer Form viel mehr als die ursprünglich gestörte Prognose.

In Tabelle 4.4 sind die Performance-Werte dargestellt, die der DMPC mit gestörten und danach geglätteten Verbrauchsprognosen erzeugt. Zu beobachten ist, dass ab einem Fehler von zehn Prozent alle Performance-Werte besser sind als vor dem Glätten. Allerdings sind die Prognosen auch genauer, was an den niedrigeren NRMS Werten erkennbar ist. Besonders zu beachten ist aber, dass der ASF Wert deutlich niedriger als vor dem Glätten ausfällt. Nur bei fünf Prozent Abweichung sind alle Werte größer. Dies liegt allerdings daran, dass die Genauigkeit der Verbrauchsprognose durch den Spline deutlich verschlechtert wird: Ihr NRMS Wert ist um 0.014 größer.

Die durch die Optimierung erzeugten Nachfragen sind in Abbildung 4.4 dargestellt. Erkennbar ist, dass die Schwankungen weniger extrem ausfallen als vor dem Glätten. Zusammenfassend lässt sich sagen, dass der DMPC mit vernünftigen Verbrauchsprognosen gute Ergebnisse liefert, obwohl sich Fehler in aggregierten Prognosen nicht gänzlich vermeiden lassen. Die Simulation mit geglätteter, bis zu 20 Prozent schwankender Verbrauchsprognose stellt mit ihrem NRMS Wert von 0.088 eine realistisch genaue Prognose dar. In dieser Simulation kann der PTP Wert auf ein Viertel, der MQD Wert auf deutlich weniger als ein Zehntel und der ASF Wert immerhin um ca. 30 Prozent im Vergleich zum ungesteuerten System reduziert werden.

Für alle weiteren Kapitel werden wir ungeachtet dieser Resultate auf die Nutzung perfekter Verbrauchsprognosen zurückgreifen, um die Konvergenzgeschwindigkeit des DMPCs nicht unnötig zu beeinflussen. Im nächsten Kapitel entfernen wir uns wieder von dem Fokus auf Verbrauchsschätzung und untersuchen die Erweiterung des Modells um Elektroautos.

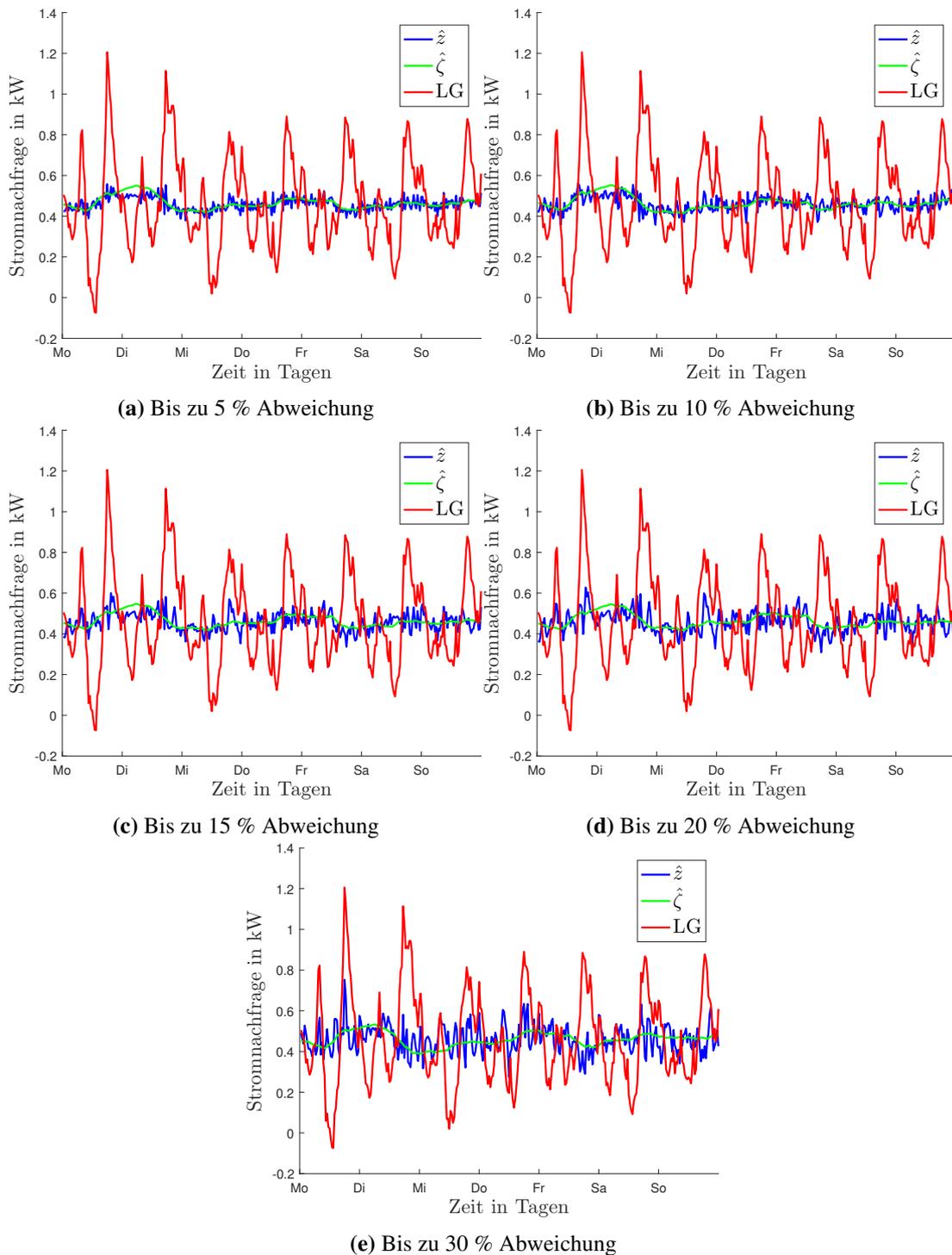


Abbildung 4.4: Durchschnittliche Stromnachfrage von 50 Haushalten über eine Woche mit gestörten, aggregierten und danach geglätteten Verbrauchsprognosen

Das Elektroauto im Smart Grid

Nachdem wir den DMPC für das Grundmodell, so wie er in [1] eingeführt wurde, in Abschnitt 3.3 bereits ausreichend beleuchtet haben, richten wir unsere Aufmerksamkeit im Folgenden auf die Erweiterung des Modells um Elektroautos. Insbesondere werden wir darauf eingehen, an welchen Stellen neue Variablen hinzugefügt werden müssen und wie sich die Änderungen auf den DMPC Algorithmus auswirken.

5.1 Erweiterung des Modells

Die Idee der folgenden Modellierung basiert darauf, dass das Elektroauto für uns nichts weiter als eine zusätzliche Batterie ist, für die einige Besonderheiten gelten. Zum Einen kann sie nur dann verwendet werden, wenn das Auto ans Netz angeschlossen ist. Dies ist dann der Fall, wenn es zu Hause in der Garage, in der Arbeit oder an einer öffentlichen Ladestation steht, solange diese Teil des gleichen Smart Grids ist. Zu jedem anderen Zeitpunkt ist unser Elektroauto “unterwegs” und kann nicht ge- oder entladen werden. Zum Anderen verbraucht das Fahren geladenen Strom. Daher gehen wir davon aus, dass die Batterie nach einer “Unterwegs-Phase”¹ mit geringerer Ladung wieder ins Netz kommt. Wir erweitern das Modell pro RES $i \in \mathbb{N}_{\mathcal{I}}$ um eine zusätzliche Zustandsvariable $x_{v_i} \in \mathbb{R}$ für den Ladestand in kWh sowie eine Steuerungsmöglichkeit $u_{v_i} \in \mathbb{R}$ für die Laderate des Elektroautos in kW.

Bemerkung 5.1.1. Ob ein RES tatsächlich ein Elektroauto oder eine Haushaltsbatterie besitzt, kann stets durch die maximale Kapazität der Batterien bestimmt werden. Ist diese gleich 0, so entspricht dies einem RES ohne den entsprechenden Speicher.

Wann das Auto unterwegs ist, wird mit den Parametern $c_{v_i}^{\text{off}}(k) \in \{0, 1\}$ realisiert. Dabei steht $c_{v_i}^{\text{off}}(k) = 1$ dafür, dass das Fahrzeug des RES i zum Zeitpunkt $k \in \mathbb{N}$ unterwegs ist. Bei der Modellierung stellen wir fest, dass das Update der Autobatterieladung im Fall, dass das Auto am Netz angeschlossen ist, genau dem Update der normalen Batterie entspricht. Nur wenn es gerade unterwegs ist, müssen wir die Dynamik so ändern, dass während des Fahrens Ladung verloren geht und die gewählte

¹Also einem Zeitraum, in dem das BEV nicht am Netz angeschlossen ist.

Steuerung keinen Einfluss auf das System hat. Dafür führen wir analog zur Vorhersage der Stromerzeugung und des -verbrauchs den Parameter $w_{v_i}(k)$ für alle $i \in \mathbb{N}_{\mathcal{I}}$ und für jeden Zeitpunkt $k \in \mathbb{N}_{\mathcal{N}}$ ein. Dieser besagt, wie viel Strom das Elektroauto von RES i zum Zeitpunkt k während des Fahrens verbraucht. Diese Werte könnten beispielsweise durch Analyse des Fahrverhaltens des Fahrers prognostiziert werden. Für RES ohne Elektroauto setzen wir $w_{v_i}(k)$ auf 0.

Die Systemdynamik ergibt sich dann wie folgt:

$$\begin{aligned} x_{v_i}(k+1) &= f_{v_i}(x_{v_i}(k), u_{v_i}(k)) := \\ & x_{v_i}(k) + T \cdot u_{v_i}(k) \cdot (1 - c_{v_i}^{\text{off}}(k)) - w_{v_i}(k) \cdot c_{v_i}^{\text{off}}(k) \end{aligned} \quad (5.1)$$

$$z_i(k) = w_i(k) - g_i(k) + u_i(k) + (1 - c_{v_i}^{\text{off}}(k)) \cdot u_{v_i}(k) \quad (5.2)$$

Da durch den Einsatz des Parameters $c_{v_i}^{\text{off}}$ sichergestellt ist, dass die Wahl von u_{v_i} für das System keine Rolle spielt, solange das Elektroauto unterwegs ist, muss die zulässige Menge der Steuerungsentscheidungen zu diesen Zeitpunkten nicht zusätzlich eingeschränkt werden. Somit entsprechen die Nebenbedingungen der Elektroauto-Batterie denen der Haushaltsbatterie. Allerdings wollen wir eine weitere Funktionalität integrieren: Da eine leere Autobatterie - im Gegensatz zu einer leeren Haushaltsbatterie - zu Problemen, wie beispielsweise dem Stehenbleiben auf dem Heimweg, führen kann, wollen wir zum Zeitpunkt des Losfahrens eine Mindestladung erzwingen. Um dies zu realisieren, führen wir neben dem Parameter für die Kapazität der Batterie $\bar{C}_{v_i} \in \mathbb{R}_{\geq 0}$ und den minimal und maximal Laderaten $\underline{u}_{v_i} \in \mathbb{R}_{\leq 0}$ bzw. $\bar{u}_{v_i} \in \mathbb{R}_{\geq 0}$ eine zeitabhängige untere Schranke für den Batteriestand $\underline{C}_{v_i} \in [0, \bar{C}_{v_i}]^{\mathcal{N}}$ ein. Da diese Einschränkung allerdings nur zum Zeitpunkt des Losfahrens notwendig ist, ist diese Schranke meistens gleich null. Die Nebenbedingungen für Elektroautos lauten demnach für alle $k \in \mathbb{N}_{\mathcal{N}}$

$$\underline{C}_{v_i}(k) \leq x_{v_i}(k) \leq \bar{C}_{v_i} \quad (5.3)$$

$$\underline{u}_{v_i} \leq u_{v_i}(k) \leq \bar{u}_{v_i}. \quad (5.4)$$

Ob ein RES eine Hausbatterie und/oder ein Elektroauto besitzt, kann direkt mit der Wahl von \bar{C}_i bzw. \bar{C}_{v_i} gesteuert werden. Analog zu (3.3) ergeben sich die zulässigen Zustände und Steuerungen als

$$\mathbb{X}_{v_i}(k) := \{x_{v_i} \in \mathbb{R} \mid \underline{C}_{v_i}(k) \leq x_{v_i} \leq \bar{C}_{v_i}\}, \quad \mathbb{U}_{v_i} := \{u_{v_i} \in \mathbb{R} \mid \underline{u}_{v_i} \leq u_{v_i} \leq \bar{u}_{v_i}\}.$$

Die in (3.4) bzw. (3.5) definierten Mengen lassen sich nun unter Berücksichtigung der neuen Systemdynamik (5.1) bzw. (5.2) erweitern. Als Ergebnis erhalten wir die für die Erweiterung zulässigen Kontrollfolgen

$$\mathbb{U}_{v_i}^{k,N}(x_{i,0}, x_{v_i,0}) := \left\{ \left(\begin{array}{c} u_i(k; N) \\ u_{v_i}(k; N) \end{array} \right) \in \mathbb{R}^{2N} \left| \begin{array}{l} x_i(k) = x_{i,0}, \quad x_{v_i}(k) = x_{v_i,0} \\ x_i(j+1) = x_i(j) + T u_i(j) \\ x_{v_i}(j+1) = x_{v_i}(j) - w_{v_i}(j) \cdot c_{v_i}^{\text{off}}(j) \\ \quad + T \cdot u_{v_i}(j) \cdot (1 - c_{v_i}^{\text{off}}(j)) \\ u_i(j) \in \mathbb{U}_i, \quad u_{v_i}(j) \in \mathbb{U}_{v_i} \\ x_i(j) \in \mathbb{X}_i, \quad x_{v_i}(j) \in \mathbb{X}_{v_i}(j) \\ \forall j = k, \dots, k + N - 1 \end{array} \right. \right\}, \quad (5.5)$$

bzw. Nachfragefolgen

$$\mathbb{D}_{v_i}^{k,N}(x_{i,0}, x_{v_i,0}) := \left\{ z_i(k; N) \in \mathbb{R}^N \left| \begin{array}{l} x_i(k) = x_{i,0}, \quad x_{v_i}(k) = x_{v_i,0} \\ x_i(j+1) = x_i(j) + Tu_i(j) \\ x_{v_i}(j+1) = x_{v_i}(j) - w_{v_i}(j) \cdot c_{v_i}^{\text{off}}(j) \\ \quad + T \cdot u_{v_i}(j) \cdot (1 - c_{v_i}^{\text{off}}(j)) \\ z_i(j) = w_i(j) - g_i(j) + u_i(j) \\ \quad + (1 - c_{v_i}^{\text{off}}(j)) \cdot u_{v_i}(j) \\ u_i(j) \in \mathbb{U}_i, \quad u_{v_i}(j) \in \mathbb{U}_{v_i} \\ x_i(j) \in \mathbb{X}_i, \quad x_{v_i}(j) \in \mathbb{X}_{v_i}(j) \\ \forall j = k, \dots, k + N - 1 \end{array} \right. \right\}. \quad (5.6)$$

Genauso wie in (3.6) definieren wir die Mengen $\mathbb{U}_v^{k,N}(x_0, x_{v_0})$ und $\mathbb{D}_v^{k,N}(x_0, x_{v_0})$ für das Gesamtsystem als kartesisches Produkt der jeweiligen Mengen der Einzelhaushalte.

Im folgenden Beispiel werden wir zeigen, dass die Erweiterung um Elektroautos in gewissen Fällen dazu führt, dass die Existenz optimaler Lösungen nicht mehr gewährleistet werden kann. Tatsächlich kann es sogar passieren, dass jede Steuerung unzulässig wird.

Beispiel 5.1.1. Für alle RES $i \in \{1, \dots, \mathcal{I}\}$ werden dieselben Parameter gewählt. Wir betrachten Intervalllängen von 30 Minuten, also $T = 0.5$. Zu Beginn seien sowohl die Haus- als auch Autobatterien leer, d.h. $x_i(0) = x_{v_i}(0) = 0$. Nun können Daten so gewählt werden, dass die Elektroautos bereits im ersten Zeitpunkt unterwegs sind. Dies entspricht $w_{v_i}(0) > 0$ und $c_{v_i}^{\text{off}}(0) = 1$. Die Autos fahren also mit leerem Akku los und somit erhalten wir für alle $u_{v_i}(0)$ einen Widerspruch dazu, dass die Ladestände nicht negativ werden dürfen:

$$x_{v_i}(1) = 0 + 0.5 \cdot u_{v_i}(0) \cdot 0 - w_{v_i}(0) = -w_{v_i}(0) < 0.$$

Damit diese Problematik im weiteren Verlauf der Arbeit nicht auftritt, werden wir im Folgenden Annahmen treffen, die auch mit Elektroautos die Existenz einer zulässigen Lösung garantieren. Hierfür führen wir den Begriff der Unterwegs-Phase ein.

Definition 5.1.1 (Unterwegs-Phase). Ein Paar von Zeitpunkten (a, b) mit $a, b \in \mathbb{N}$ heißt Unterwegs-Phase für ein RES $i \in \mathbb{N}_{\mathcal{I}}$, wenn gilt:

$$a \leq b \quad (5.7)$$

$$c_{v_i}^{\text{off}}(a-1) = 0 \quad (5.8)$$

$$c_{v_i}^{\text{off}}(b+1) = 0 \quad (5.9)$$

$$c_{v_i}^{\text{off}}(t) = 1 \quad \forall t \in \{a, \dots, b\} \quad (5.10)$$

Der Gesamtstromverbrauch einer Unterwegs-Phase ist gegeben durch

$$W_{v_i}(a, b) := \sum_{j=a}^b w_{v_i}(j).$$

Annahme 5.1.1. Um zu gewährleisten, dass immer eine zulässige Kontrolle existiert, müssen die möglichen maximalen Laderaten sowie die Kapazitäten der Autobatterien in einem sinnvollen Verhältnis zum Stromverbrauch während der Unterwegs-Phasen stehen. Formal bedeutet dies:

Zu jedem Zeitpunkt $k \in \mathbb{N}_{\mathcal{N}}$ und für jedes RES $i \in \mathbb{N}_{\mathcal{I}}$ muss gelten: Es existiert eine Kontrollfolge $\mathbf{u}_{v_i}(k; N)$, sodass für jede Unterwegs-Phase (a, b) nach Definition 5.1.1 mit $a \in \{k, \dots, k + N - 1\}$ gilt:

$$x_{v_i}(a - 1) \geq W_{v_i}(a, b).$$

Dann gilt für alle $j \in \{a, \dots, b\}$:

$$x_{v_i}(j) = x_{v_i}(a - 1) - \sum_{l=k}^j w_{v_i}(l) \geq x_{v_i}(a - 1) - W_{v_i}(a, b) \geq 0.$$

Wir werden in den Simulationen sehen, dass die obige Eigenschaft ohne Schwierigkeiten eingehalten werden kann wenn wir uns auf den Einsatz von BEVs als Zweitwagen, welche für das tägliche Pendeln zur Arbeit und zurück verwendet werden, konzentrieren. Diese Einschränkung lässt sich dadurch begründen, dass Elektroautos noch nicht die Reichweite bzw. Infrastruktur besitzen, die notwendig wäre, um Verbrenner komplett ersetzen zu können. Außerdem liegen die Stärken des Elektromotors vor allem im Stadtverkehr.

5.2 DMPC mit Elektroautos

Algorithmus 3 aus Kapitel 3 muss an mehreren Stellen angepasst werden, um auch mit der Erweiterung um Elektroautos zu funktionieren. Wie die erweiterte Version aussieht, ist in Algorithmus 4 zu sehen. Da die relevanteren Anpassungen in den einzelnen RES passieren, betrachten wir zunächst Phase 2 des Algorithmus. In Schritt 6 müssen neben \mathbf{u}_i und \mathbf{z}_i nun zudem die Kontrollvariablen des Elektroautos aktualisiert werden. Dies geschieht analog zur Aktualisierung der normalen Laderaten:

$$\mathbf{u}_{v_i}^{l+1} = \theta^l \mathbf{u}_{v_i}^{*l} + (1 - \theta^l) \mathbf{u}_{v_i}^l.$$

Danach werden im siebten Schritt zusätzlich zu den Ladeständen der Hausbatterien auch die Ladestände der BEVs berechnet:

$$x_{v_i}^{l+1}(j + 1) = f_{v_i}(x_{v_i}^{l+1}(j), u_{v_i}^{l+1}(j)), \quad \forall j \in \{k, \dots, k + N - 1\}.$$

In Schritt 8 werden weitere Zulässigkeitsindikatoren eingeführt. Dabei wird $\nu_{v_i}^l$ auf 1 gesetzt, wenn $\mathbf{x}_{v_i}^{l+1} \in \mathbb{X}_{v_i}^N$, und auf 0 falls $\mathbf{x}_{v_i}^{l+1} \notin \mathbb{X}_{v_i}^N$. Diese werden in Schritt 11

gemeinsam mit den anderen Indikatoren an die CE übergeben. Die größte Änderung erfolgt jedoch im zehnten Schritt. Wir ersetzen das Optimierungsproblem durch

$$\mathbf{z}_i^{*l+1} = \arg \min_{\mathbf{z}_i \in \mathbb{D}_{V_i}^{k,N}} g_{V_i}(\mathbf{z}_i; \mathbf{p}_i^{l+1}), \quad (5.11)$$

wobei nicht nur über die neue Menge der zulässigen Nachfragefolgen $\mathbb{D}_{V_i}^{k,N}$, welche die Nutzung der Elektroautos berücksichtigt, optimiert wird, sondern auch eine geänderte Kostenfunktion minimiert wird. Durch den Einsatz von Elektroautos ändert sich neben der Kommunikationsvariable z ein weiterer entscheidender Punkt: Der tatsächliche Stromverbrauch. Bisher wurde für die Optimierung der Abstand von der durchschnittlichen Stromnachfrage $\hat{\zeta}$ bestraft. Elektroautos verbrauchen jedoch zusätzlichen Strom beim Fahren, der bisher zwar in der Systemdynamik, nicht aber in der Kostenfunktion berücksichtigt wird. Steuerungen sind nur dann zulässig, wenn sie sicherstellen, dass die Ladungen der Fahrzeuge nie unter Null fallen. Daher muss das Optimierungsziel in Abhängigkeit des erwarteten, durch die Elektroautos höheren Stromverbrauchs angepasst werden. Andernfalls würde der Optimierer versuchen, eine zu niedrige Nachfrage anzustreben, die unter den gegebenen Bedingungen nicht einzuhalten ist.

Um die neue Kostenfunktion aufzustellen, fügen wir zur angestrebten Nachfrage die durchschnittlichen erwarteten Stromkosten der Elektroautos hinzu:

$$\hat{\zeta}_v(k) := \hat{\zeta} + \frac{1}{\mathcal{I}N} \sum_{j=k}^{k+N-1} \sum_{i=1}^{\mathcal{I}} w_{V_i}(j) \quad (5.12)$$

Die lokale Kostenfunktion wird also ersetzt durch

$$g_{V_i}(\mathbf{z}_i; \mathbf{p}_i) := \sum_{j=0}^{N-1} \left(\hat{z}(j) - \hat{\zeta}_v(k) \right)^2. \quad (5.13)$$

In Phase 1 muss das Optimierungsziel im ersten Schritt in gleichem Maße angepasst werden. Die optimale Schrittweite in der Erweiterung ist gegeben durch:

$$\theta^l := \arg \min_{\theta \in [0,1]} G_v(\theta \mathbf{z}^{*l} + (1-\theta) \mathbf{z}^l) \quad \text{mit} \quad G_v(\mathbf{z}) := \sum_{j=0}^{N-1} \left(\hat{z}(j) - \hat{\zeta}_v(k) \right)^2 \quad (5.14)$$

Auch für die Bestimmung der Performance-Werte im zweiten Schritt werden nun die veränderten Kosten G_v verwendet. In den Abbruchbedingungen aus Schritt 3 und 4 müssen alle Bedingungen auch für die neuen Zulässigkeitsindikatoren $\nu_{V_i}^l$ geprüft werden. Ein Abbruch in Schritt 3 erfolgt also nur dann, wenn entweder $\nu_i^l = 1$ und $\nu_{V_i}^l = 1$ für alle $i \in \mathbb{N}_{\mathcal{I}}$ erfüllt ist oder es ein $i \in \mathbb{N}_{\mathcal{I}}$ gibt mit $\nu_i^l = 0$ oder $\nu_{V_i}^l = 0$ und gleichzeitig $|G^{*l} - G^l| < \varepsilon_2$ gilt. Die Abbruchkriterien im vierten Schritt werden analog angepasst.

Algorithmus 4 DMPC mit Elektroautos

Input: Lege die Anzahl der RES $\mathcal{I} \in \mathbb{N}$, den MPC-Horizont $N \in \mathbb{N}$, die maximale Iterationszahl $l_{\max} \in \mathbb{N} \cup \{\infty\}$ und die gewünschten Genauigkeiten $\varepsilon_1, \varepsilon_2, \delta > 0$ fest. Setze die zulässigen Mengen $\mathbb{D}_{v_i}^{k,N}(x_i(k), x_{v_i}(k))$ abhängig von den Startzuständen $x_i(k) \in \mathbb{X}_i, x_{v_i}(k) \in \mathbb{X}_{v_i}$ und den zeitabhängigen Parametern $\mathbf{g}_i(k; N), \mathbf{w}_i(k; N), \mathbf{w}_{v_i}(k; N)$. Setze $l = 0$.

Hauptschleife:

Phase 1 (CE): Erhalte $\mathbf{z}_i^{*l}, \nu_i^l$ und $\nu_{v_i}^l, i \in \mathbb{N}_{\mathcal{I}}$.

1. Berechne die Schrittweite $\theta^l = \arg \min_{\theta \in [0,1]} G_v(\theta \mathbf{z}^{*l} + (1 - \theta) \mathbf{z}^l)$.
2. Berechne $\mathbf{z}^{l+1} = \theta^l \mathbf{z}^{*l} + (1 - \theta^l) \mathbf{z}^l$ und bestimme die Performance-Werte $G_v^{*l} = G_v(\mathbf{z}^{*l})$ und $G_v^{l+1} = G_v(\mathbf{z}^{l+1})$.
3. Falls $|G_v^{l+1} - G_v^l| < \varepsilon_1$,
 - Wenn $\nu_i^l = 1$ und $\nu_{v_i}^l = 1$ für alle $i \in \mathbb{N}_{\mathcal{I}}$, gib θ^l zurück und beende den Algorithmus.
 - Wenn es ein $i \in \mathbb{N}_{\mathcal{I}}$ mit $\nu_i^l = 0$ oder $\nu_{v_i}^l = 0$ gibt und $|G_v^{*l} - G_v^l| < \varepsilon_2$, gib θ^l zurück und beende den Algorithmus.
 - Sonst übertrage θ^l und \mathbf{z}^{l+1} an alle RES.
4. Falls $l = l_{\max}$ oder $\theta^l < \delta$,
 - Wenn $\nu_i^l = 1$ und $\nu_{v_i}^l = 1$ für alle $i \in \mathbb{N}_{\mathcal{I}}$, gib θ^l zurück und beende den Algorithmus.
 - Wenn es ein $i \in \mathbb{N}_{\mathcal{I}}$ mit $\nu_i^l = 0$ oder $\nu_{v_i}^l$ gibt, gib $\theta^l = 1$ zurück und beende den Algorithmus.
5. Sonst übertrage θ^l und \mathbf{z}^{l+1} an alle RES.

Phase 2 (RES $i, i \in \mathbb{N}_{\mathcal{I}}$): Erhalte θ^l und \mathbf{z}^{l+1} .

6. Aktualisiere die Kontroll- und Kommunikationsvariablen, also

$$\mathbf{u}_i^{l+1} = \theta^l \mathbf{u}_i^{*l} + (1 - \theta^l) \mathbf{u}_i^l, \quad \mathbf{u}_{v_i}^{l+1} = \theta^l \mathbf{u}_{v_i}^{*l} + (1 - \theta^l) \mathbf{u}_{v_i}^l \quad \text{und} \quad \mathbf{z}_i^{l+1} = \theta^l \mathbf{z}_i^{*l} + (1 - \theta^l) \mathbf{z}_i^l.$$

7. Berechne $\mathbf{x}_i^{l+1}, \mathbf{x}_{v_i}^{l+1}$ basierend auf \mathbf{u}_i^{l+1} bzw. $\mathbf{u}_{v_i}^{l+1}$, d.h.:

$$\begin{aligned} x_i^{l+1}(j+1) &= f_i(x_i^{l+1}(j), u_i^{l+1}(j)), \\ x_{v_i}^{l+1}(j+1) &= f_{v_i}(x_{v_i}^{l+1}(j), u_{v_i}^{l+1}(j)) \quad \forall j \in \{k, \dots, k + N - 1\} \end{aligned}$$

8. Falls $\mathbf{x}_i^{l+1} \in \mathbb{X}_i^N$ (also \mathbf{x}_i^{l+1} zulässig ist), so setze $\nu_i^{l+1} = 1$. Sonst setze $\nu_i^{l+1} = 0$. Falls $\mathbf{x}_{v_i}^{l+1} \in \mathbb{X}_{v_i}^N$ (also $\mathbf{x}_{v_i}^{l+1}$ zulässig ist), so setze $\nu_{v_i}^{l+1} = 1$. Sonst setze $\nu_{v_i}^{l+1} = 0$.
9. Setze \mathbf{p}_i^{l+1} wie in Gleichung (3.17).
10. Löse das Optimierungsproblem $\mathbf{z}_i^{*l+1} = \arg \min_{\mathbf{z}_i \in \mathbb{D}_{v_i}} g_{v_i}(\mathbf{z}_i; \mathbf{p}_i^{l+1})$.
11. Übermittle $\mathbf{z}_i^{*l+1}, \nu_i^{l+1}$ und $\nu_{v_i}^{l+1}$ an die CE.

Setze $l = l + 1$, gehe zu Phase 1.

Numerische Untersuchungen des DMPC mit Elektroautos

In diesem Kapitel wollen wir uns zum einen mit der Frage beschäftigen, ob der DMPC mit der Erweiterung um Elektroautos immer noch notwendige Eigenschaften erfüllt, um eine Konvergenz der berechneten Lösungen zu einem Optimum hin zu gewährleisten. Zum anderen werden wir darauf eingehen, wie der Algorithmus ein System mit unterschiedlichen Anzahlen von Haushaltsbatterien und Elektroautos steuert.

Um vernünftige Vergleiche anstellen zu können, werden im Folgenden einigen Rahmenbedingungen festgelegt, welche als Grundlage für alle weiteren Simulationen dienen sollen.

6.1 Grundmodell

Wir untersuchen 50 Haushalte über eine Woche hinweg. Darunter gibt es abhängig vom Szenario Haushalte, die eine Batterie und/oder ein Elektroauto besitzen sowie einige, bei denen beides vorhanden ist. Da die zugrunde liegenden Verbrauchs- und Erzeugungsdaten aus [19] aus dem Jahr 2013 stammen, legen wir Batterieeigenschaften fest, die möglichst zu dieser Zeit passen. Außerdem nehmen wir vereinfachend an, dass es bei den Batterie- und Fahrzeugtypen keine Unterschiede zwischen den einzelnen Haushalten gibt¹. Eine klassische Batterie zur Speicherung von Solarstrom besitzt eine Kapazität von 4 kWh und eine maximale Lade- bzw. Entladerate von 2.5 kW². Für die Daten der Elektroautos richten wir uns nach dem Mitte 2013 eingeführten und im Jahr 2015 meistverkauften Auto, dem Renault ZOE, dessen Batterie in der üblichen Ausstattung 22 kWh fasst [21]. Für die maximalen Lade- und Entladeraten beziehen wir uns auf die Leistung der in Deutschland am häufigsten vertretenen Ladestation³, welche bei maximal 22 kW liegt.

¹Es wäre jedoch technisch jederzeit möglich, Kapazitäten und Laderaten individuell anzupassen.

²Zum Beispiel die sonnenBatterie eco 8/4 aus dem Jahr 2014 [20].

³Dabei handelt es sich um eine Art der Ladestation IEC Typ 2, welcher 2013 eingeführt wurde. Heute sind mehr als ein Drittel aller Ladepunkte in Deutschland dieser Art (Stand Juli 2018). [23]

Weiter legen wir fest, dass die Besitzer eines Elektroautos damit von Montag bis Freitag zur Arbeit und wieder nach Hause fahren. Dafür soll die Batterie des Autos am Morgen ausreichend Strom geladen haben. Laut einer Studie des Statistischen Bundesamtes aus dem Jahr 2014 [24] beträgt der durchschnittliche einfache Arbeitsweg der Deutschen ca. 10 km. Da man aufgrund von Umwegen durch Baustellen, Staus oder diversen Besorgungen manchmal etwas weiter fährt, zählen wir noch einmal 10 km hinzu. Bei einem Verbrauch von knapp 20 kWh pro 100 km [21] erreichen wir somit einen durchschnittlichen Tagesverbrauch von 6 kWh. Bei einem “normalen” Auto befindet sich bei leerer Kraftstoffanzeige als Reserve ca. 5 Liter im Tank [22]. Dies entspricht bei einem geschätzten Tankvolumen von 40 Litern einer Reserve von 12.5 Prozent. Um diese Sicherheit für das Elektroauto zu erreichen, fügen wir noch 2.75 kWh Reserve hinzu. Zum Zeitpunkt des Losfahrens müssen sich also mindestens 8.75 kWh Strom in der Batterie befinden.

Für die Optimierung mittels MPC legen wir eine Diskretisierung mit halbstündigen Intervallen sowie einen Horizont von 24 Stunden fest. Wir gehen demnach davon aus, dass für die Dauer eines Tages zuverlässige Vorhersagen über Verbrauch und Erzeugung gemacht werden können. Für die tatsächlichen Verbrauchs- und Erzeugungsdaten nehmen wir eine Woche aus den in [19] vom australischen Stromanbieter Ausgrid zur Verfügung gestellten Daten her.

6.2 Die Turnpike Eigenschaft

6.2.1 Ein Indikator für Stabilität

Die Turnpike Eigenschaft ist in vielen kontrolltheoretischen Problemen nachweisbar, die auf einem ökonomischen Kostenmodell basieren. Ein System besitzt die Turnpike Eigenschaft, wenn sich Lösungen zu unterschiedlichen Startwerten in der Nähe einer bestimmten Trajektorie bewegen. Der Name *Turnpike*⁴ ergibt sich aus einer einfachen Anschauung: Um auf schnellstem Weg von einem Ort zu einem anderen zu kommen, fährt man in der Regel bei der nächsten Autobahnauffahrt auf und bei der Abfahrt, die dem Zielort am nächsten kommt, wieder herunter. Legt man den Startort nun weiter weg, wird man trotzdem einen großen Teil der Strecke auf derselben Autobahn fahren. Auf Kontrollsysteme übertragen bedeutet dies, dass Lösungen zu unterschiedlichen Startzuständen zu den meisten Zeitpunkten übereinstimmen.

Es existieren mehrere mathematische Formulierungen, die diese Eigenschaft beschreiben. Für unsere Arbeit werden wir uns auf die in [?] beschriebene Formulierung festlegen. Dafür führen wir modifizierte Stufenkosten ein, die den tatsächlichen Kosten entsprechen, jedoch um die Stufenkosten eines optimalen Trajektorien-Paars reduziert sind.

Definition 6.2.1.

Sei (x^*, u^*) ein optimales Trajektorien-Paar für ein zeitabhängiges Optimalsteuerungs-

⁴Aus dem Englischen: *Schnellstraße*

problem mit Startzustand x_0 . Dann sind die *modifizierten Stufenkosten* einer Trajektorie (x, u) gegeben als

$$\hat{g}(k, x(k), u(k)) := g(k, x(k), u(k)) - g(k, x^*(k), u^*(k)).$$

Damit lässt sich nun die Turnpike Eigenschaft formal definieren.

Definition 6.2.2.

Sei (x^*, u^*) ein optimales Trajektorien-Paar für ein zeitabhängiges Optimalsteuerungsproblem. Dann besitzt das Problem mit modifizierten Stufenkosten \hat{g} die *zeitabhängige Turnpike-Eigenschaft bei (x^*, u^*)*

- a) *auf unendlichem Horizont*, falls gilt: Es existiert ein $\rho \in \mathcal{L}$ ⁵, sodass für jedes $k \in \mathbb{N}_0$, für jede optimale Trajektorie $x_{u_\infty^*}(\cdot)$ zum Startzustand $x_0 \in \mathbb{X}(k)$ und für jedes $P \in \mathbb{N}$ eine Menge $\mathcal{Q}(k, x_0, P, \infty) \subseteq \mathbb{N}_0$ mit $\#\mathcal{Q}(k, x_0, P, \infty) \leq P$ existiert, sodass

$$|(x_{u_\infty^*}(j, x), u_\infty^*(j)) - (x^*(k+j), u^*(k+j))| \leq \rho(P)$$

gilt für alle $j \in \mathbb{N}_0 \setminus \mathcal{Q}(k, x_0, P, \infty)$.

- b) *auf endlichem Horizont*, falls für die modifizierten Stufenkosten \hat{l} gilt: Es existiert ein $\rho \in \mathcal{L}$, sodass für jedes $k \in \mathbb{N}_0$, für jede optimale Trajektorie $x_{u_N^*}(\cdot)$ zum Startzustand $x_0 \in \mathbb{X}(k)$ und für jede $N, P \in \mathbb{N}$ eine Menge $\mathcal{Q}(k, x_0, P, N) \subseteq \{0, \dots, N\}$ mit $\#\mathcal{Q}(k, x_0, P, N) \leq P$ existiert, sodass

$$|(x_{u_N^*}(j, x), u_N^*(j)) - (x^*(k+j), u^*(k+j))| \leq \rho(P)$$

gilt für alle $j \in \{0, \dots, N\} \setminus \mathcal{Q}(k, x_0, P, \infty)$.

Diese Eigenschaft für ein System analytisch nachzuweisen kann sehr schwierig sein. Allerdings können Simulationen zumindest numerisch einen Hinweis darauf geben, dass die Turnpike Eigenschaft erfüllt ist.

6.2.2 Numerische Untersuchungen zur Turnpike Eigenschaft

Um die Anzahl der Variablen gering zu halten, betrachten wir in diesem Abschnitt nur zehn Haushalte, wobei acht davon eine Hausbatterie und zwei ein Elektroauto besitzen. Dabei beschränken wir uns außerdem auf einen Optimierungszeitraum von zwei Tagen, um die Lösungen mit der optimalen Trajektorie vergleichen zu können. Um die Turnpike Eigenschaft zu untersuchen, führen wir die Simulation mit unterschiedlichen Startladungen durch⁶. In Abbildung 6.1a sind die resultierenden durchschnittlichen Stromnachfragen \hat{z} (blau) im Vergleich zur ungesteuerten Nachfrage LG (rot) und der optimalen Nachfrage, also dem laufenden Durchschnitt $\hat{\zeta}(\cdot)$ (grün), zu sehen. Beginnend mit komplett leeren Batterien werden die Startladungen der Haus- und Autobatterien sukzessive in zehn Prozent Schritten erhöht (das entspricht also 0.4 bzw.

⁵Siehe Definition 2.1.1

⁶Alle weiteren Eigenschaften des Systems entsprechen denen, die in Abschnitt 6.1 vorgestellt wurden.

2.2 kWh). Dabei kann man sehr gut erkennen, dass die Trajektorien bis auf den Anfang und das Ende gleich verlaufen. Diesen Unterschied kann man auch sehr gut in den resultierenden Zustands- bzw. Kontrolltrajektorien, in Abbildung 6.1 erkennen. Dass sich die Kontrollfolgen bei unterschiedlichen Startwerten so sehr ähneln, ist ein starkes Indiz, dass die Turnpike Eigenschaft auch mit der Erweiterung um Elektroautos gewährleistet ist.

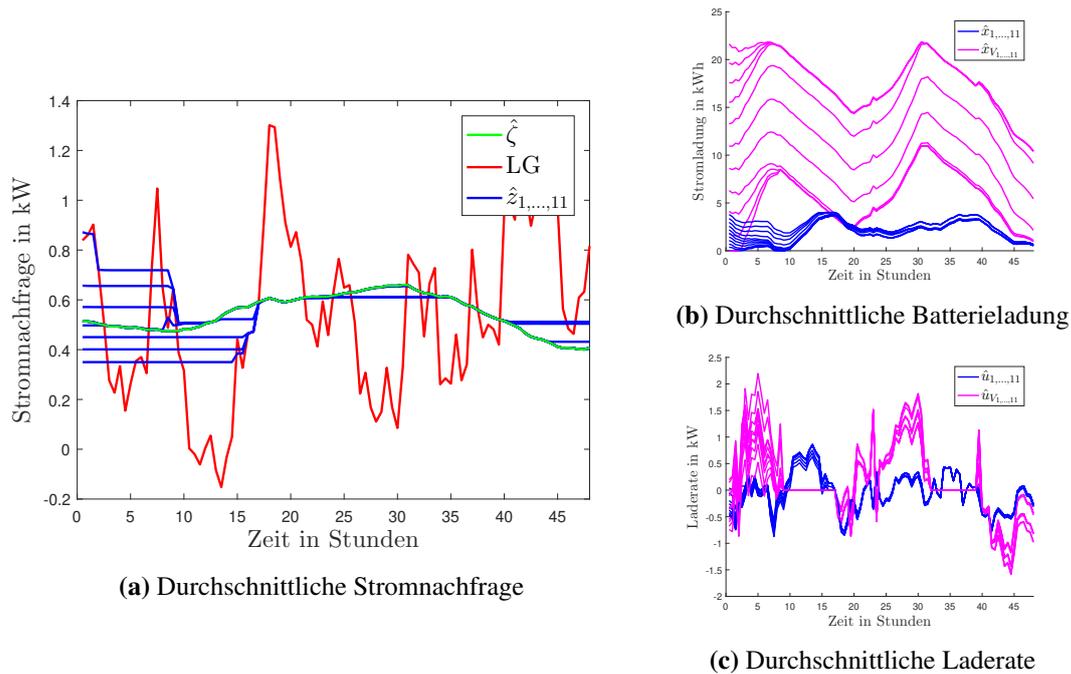


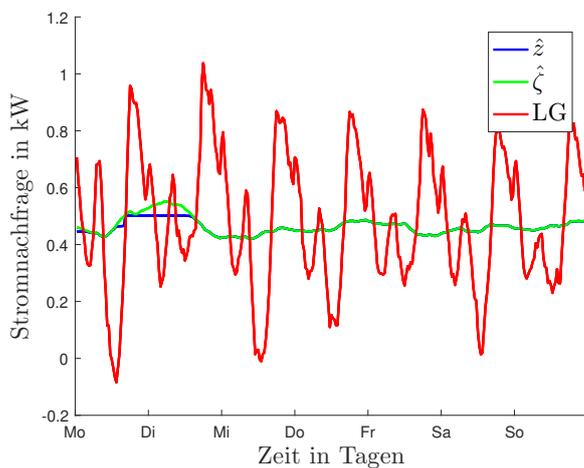
Abbildung 6.1: Simulation von zehn Haushalten mit unterschiedlichen Startladungen

6.3 Numerische Simulation des DMPC mit Elektroautos

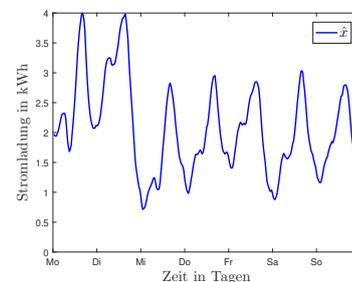
Im Folgenden werden wir verschiedene Simulationen des Grundmodells aus Abschnitt 6.1 durchführen und dabei nach und nach Haushaltsbatterien durch Elektroautos ersetzen. Dabei steigt durch den, von den Elektroautos verbrauchten, Strom wie in Kapitel 5.2 beschrieben auch die angestrebte Stromnachfrage $\hat{\zeta}$. Wir gehen davon aus, dass Besitzer von Elektroautos in der Früh zwischen 7 und 9 Uhr zur Arbeit fahren und nachmittags bzw. abends nach 8 bis 11 Stunden Arbeit wieder nach Hause kommen. Wir nehmen an, dass die BEVs während dieser Zeit vom Netz getrennt sind. Um den täglichen Arbeitsweg zurückzulegen, verbrauchen die BEVs wie in 6.1 beschrieben im Schnitt 6 kWh. Außerdem ist mit der eingebauten Reserve eine Mindestladung von 8,75 kWh zum Zeitpunkt des Losfahrens notwendig.

6.3.1 Keine Elektroautos - Optimale Nachfrage nicht erreichbar

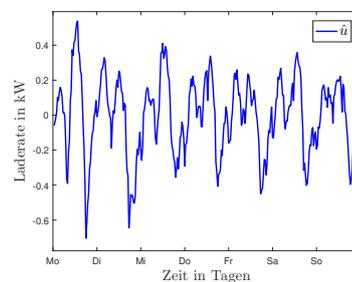
Unsere Ausgangslage entspricht der Situation in [1]: Jeder der Haushalte ist mit einer Batterie versehen, aber noch keiner besitzt ein Elektroauto. In Abbildung 6.2a ist die Stromnachfrage zu sehen, die vom DMPC berechnet wurde. An der ungesteuerten Nachfrage LG in rot können wir die uhrzeitbedingten Schwankungen sehr gut erkennen. Morgens und abends ist der Stromverbrauch in der Regel höher als am Nachmittag und Photovoltaikanlagen erzeugen den meisten Strom nachmittags, wenn die Sonne am stärksten scheint. Dies führt zu zwei positiven Peaks und einem Negativpeak pro Tag. Der laufende Durchschnitt $\hat{\zeta}$ schwankt zwischen 0.42 und 0.50 kW. Dass \hat{z} fast immer gleich $\hat{\zeta}$ ist, zeigt, dass das System bis auf ein paar Stunden perfekt gesteuert werden kann. An der durchschnittlichen Batterieladung in Abbildung 6.2b kann man erkennen, dass die Batteriekapazität nicht ausreicht, um den Energieüberschuss am Dienstag Nachmittag aufzunehmen. Dies führt dazu, dass die Stromnachfrage unter der gewünschten Menge bleibt. Unsere Hoffnung ist also, dass durch den Einsatz von Elektroautos die Kapazität weit genug erhöht wird, um die überschüssige Strommenge aufzunehmen. Da die eingesetzten BEVs jedoch auch zum Fahren aufgeladen werden müssen, erhält das System zusätzliche Nebenbedingungen. Außerdem steigt der Gesamtverbrauch und das Netz wird dadurch stärker belastet. Wir stellen uns also die Frage, wie groß der Anteil an Elektroautos in unserem Modell sein muss, damit die Stromnachfrage optimal gesteuert werden kann.



(a) Durchschnittliche Stromnachfrage



(b) Durchschnittliche Batterieladung



(c) Durchschnittliche Laderate

Abbildung 6.2: Simulation von 50 Haushalten über eine Woche mit Batterien und ohne Elektroautos

6.3.2 Zehn Prozent BEVs - Starkstromanschluss für Aufladen ausreichend

Wir ersetzen nun bei zehn Prozent der Haushalte die Haushaltsbatterien durch Elektroautos. Wie in Abbildung 6.3a zu sehen ist, kann das System mit der zusätzlichen Batteriekapazität noch besser gesteuert werden. In den beiden Graphen 6.3b und 6.3c sind nun auch die durchschnittlichen Batterieladungen und -laderaten der Elektroautos in Magenta dargestellt. An ihnen kann man erkennen, dass das Aufladen der Elektroautos über Nacht kein Problem ist, da zum Laden der Strom aus den Haushaltsbatterien verwendet werden kann, welche tagsüber zum Teil durch den überflüssigen Solarstrom aufgeladen werden. Eine weitere interessante Erkenntnis ist, dass sich die Laderaten der Elektroautos am Wochenende, an welchem sie in unserer Simulation nicht genutzt werden, denen der Haushaltsbatterien bis auf die Amplitude anpasst. Der Unterschied in der Stärke der Nutzung lässt sich dadurch erklären, dass die Elektroautos eine viel höhere Batteriekapazität und -leistung besitzen.

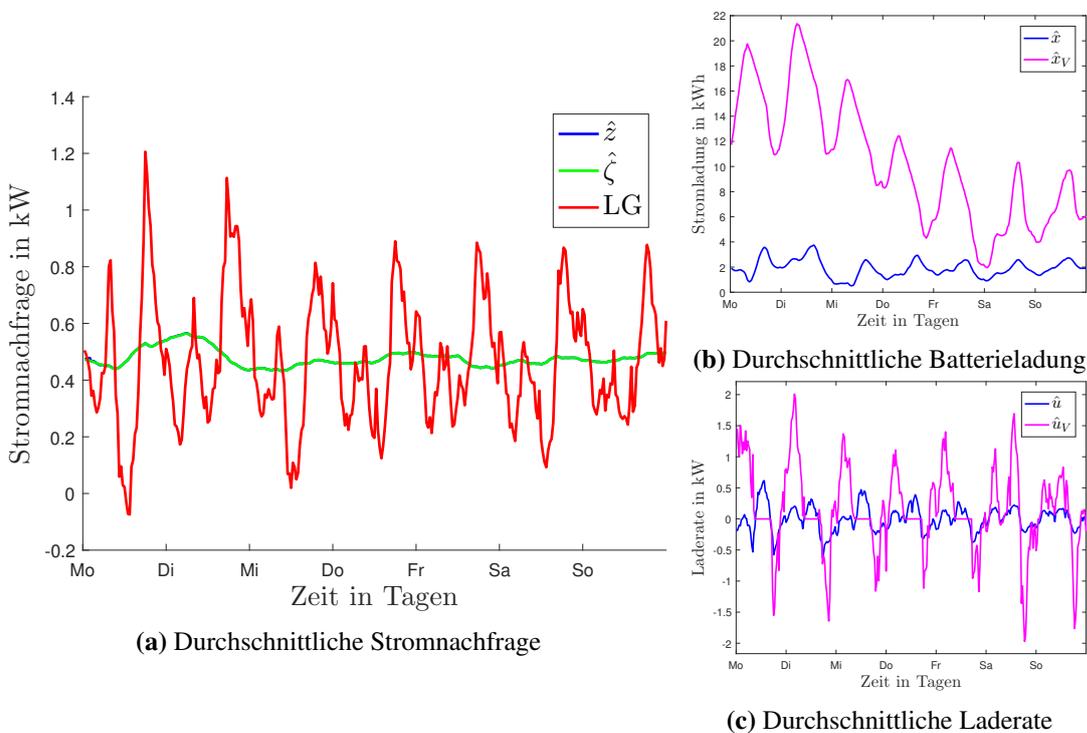


Abbildung 6.3: Simulation von 50 Haushalten über eine Woche mit 5 Elektroautos

Oft wird die Problematik angesprochen, dass Elektroautos nicht so schnell aufgeladen werden können wie man ein Auto mit Verbrennungsmotor auftanken kann. In unserer Simulation machen wir dazu eine spannende Beobachtung: Kein Elektroauto wird mit mehr als 3,7 kW aufgeladen bzw. entladen (der Durchschnitt über alle Elektroautos überschreitet sogar nur knapp die 2 kW Marke). Dieser Wert liegt weit unter den mit Schnellladestationen theoretisch möglichen 22 kW. Tatsächlich fällt noch etwas auf: Mit einem herkömmlichen Starkstrom-Anschluss sind bereits bis zu 3,7 kW Ladeleistung möglich [25]. Mithilfe des DMPC kann die Notwendigkeit einer Ladestation im Privathaushalt bei der Nutzung des BEVs als Zweitwagen in Frage gestellt werden.

6.3.3 Zwanzig Prozent BEVs - Großer Verlust durch das Ignorieren von BEVs in der Optimierung

Viele befürchten bei einem Ausbau der Elektromobilität eine Überlastung des Stromnetzes durch den stark steigenden Verbrauch. Begründet ist diese Angst dadurch, dass Besitzer von Elektroautos ihren Wagen in der Regel direkt nach dem Heimkommen an die Steckdose hängen und die Batterie mit maximal möglicher Geschwindigkeit aufladen lassen. Da die meisten Menschen tagsüber arbeiten und abends nach Hause kommen, erhöht sich der sowieso bereits hohe Verbrauch am Abend immens. Nehmen wir die selben Verbrauchswerte, wie im Grundmodell in 6.1 beschrieben, an und addieren den zusätzlichen Verbrauch auf die unkontrollierte Situation, so liefert sich ein Bild wie in Abbildung 6.4a. Die gestrichelte Linie LG_V zeigt, wie sehr die Peaks am Abend durch das Aufladen verstärkt werden. In der dargestellten Simulation wird beispielsweise am Freitag um 18:30 Uhr eine Nachfrage mit 1.29 kW erreicht. Dabei wird der Peak ohne Elektroautos um über fünfzig Prozent verstärkt.

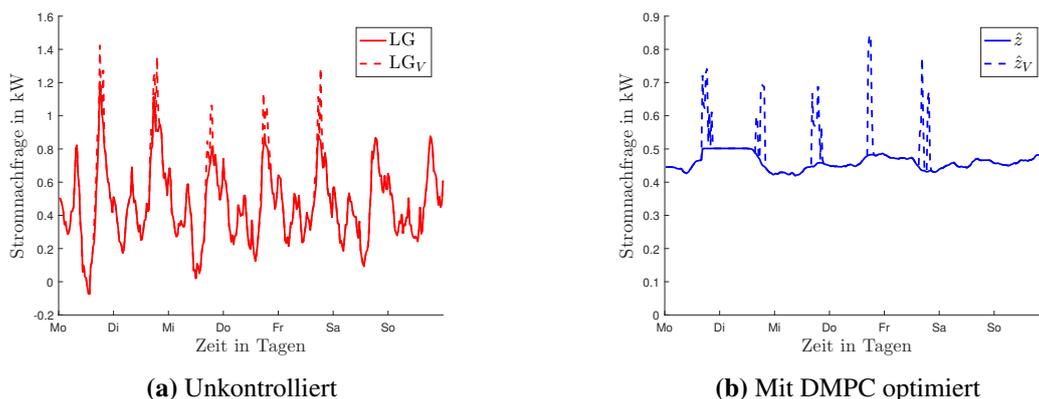


Abbildung 6.4: Durchschnittliche Stromnachfrage, falls der Verbrauch der Elektroautos im Nachhinein hinzuaddiert wird

Noch fataler verhält es sich, wenn das System ohne Rücksicht auf Elektroautos optimiert wird und diese erst hinterher in die Stromnachfrage einbezogen werden. In Abbildung 6.4b erkennen wir beispielsweise, dass am Freitag um 16:00 Uhr die Stromnachfrage um über 58 Prozent auf mehr als 0.77 kW ansteigt. Doch mithilfe unserer Erweiterung lässt sich auch bei zwanzig Prozent BEVs eine nahezu perfekte Steuerung finden, wie sie in Abbildung 6.5 dargestellt ist. Wie viel wir durch das Einbeziehen der Elektroautos in die Optimierung tatsächlich gewinnen, lässt sich auch sehr gut an Tabelle 6.1 erkennen. Der PTP Wert kann auf weniger als 20 Prozent, der RMS Wert auf fast ein Achtel und der ASF Wert auf knapp sechs Prozent im Vergleich zur Variante, in der ohne Elektroautos optimiert und deren Verbrauch im Nachhinein hinzugezählt wird, reduzieren.

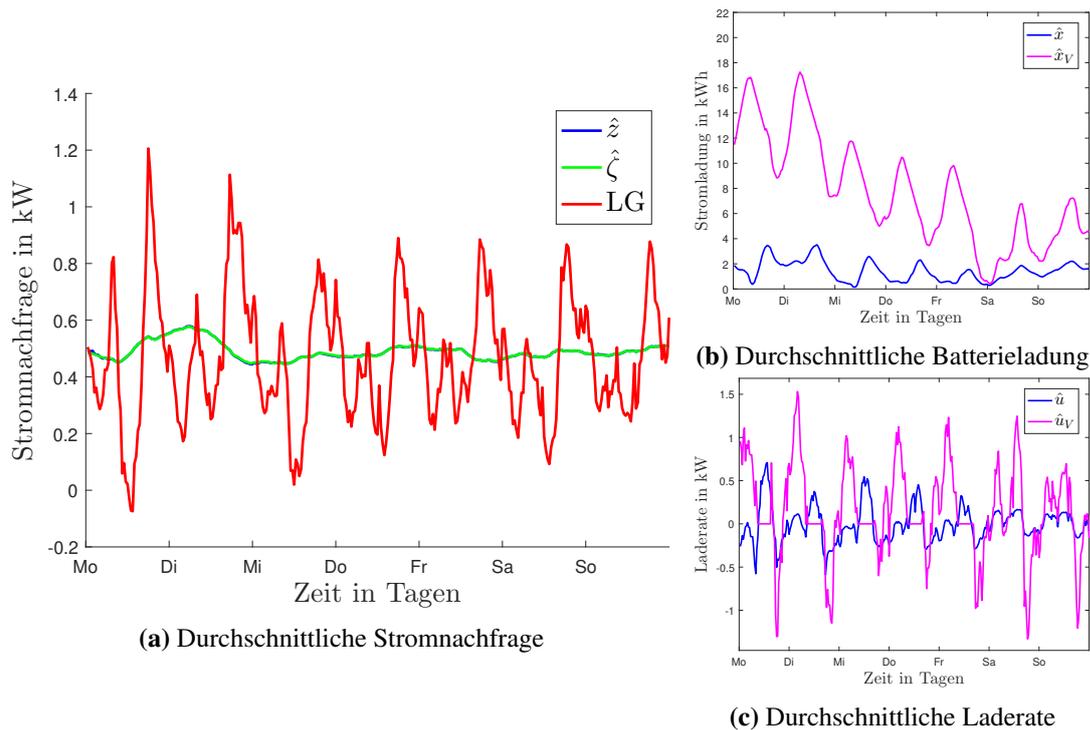


Abbildung 6.5: Simulation von 50 Haushalten über eine Woche mit 10 Elektroautos

Variante	PTP	RMS	ASF
Ungesteuert + BEV Verbrauch	1.41754	0.06353	0.10167
DMPC (ohne BEV) + BEV Verbrauch	0.45737	0.00403	0.04697
DMPC mit BEV	0.08193	0.00052	0.00290

Tabelle 6.1: Performance-Werte mit und ohne Einbezug von Elektroautos in die Optimierung

6.3.4 Fünfzig Prozent BEVs - Potential von Ladestationen an Arbeitsstellen

Besitzen die Hälfte der Haushalte keine Hausbatterie sondern ein Elektroauto, so treten erste Schwierigkeiten auf. Der Verbrauch der BEVs ist zu hoch und kann nicht komplett durch die deutlich kleineren Haushaltsbatterien getilgt werden. Diese werden zwar größtenteils weiterhin nachmittags durch Solarenergie aufgeladen, jedoch ist dies nicht ganz ausreichend. In Abbildung 6.6a ist zu sehen, dass die angestrebte Stromnachfrage $\hat{\zeta}$ zwischen Dienstag Nachmittag und Freitag Mittag nicht ganz eingehalten werden kann. Sehr schön erkennbar ist, dass der Algorithmus zu diesen Zeitpunkten durch den quadratischen Bestrafungsterm dazu gebracht wird, eine “plateauförmige” Nachfrage zu erzeugen und hohe Peaks zu vermeiden. Aus Abbildung 6.6b können wir ablesen, dass die Haushaltsbatterien bis auf Dienstag und am Wochenende komplett entladen werden müssen, um eine zu hohe Stromnachfrage zu verhindern. Des Weiteren ist die zu niedrige Nachfrage am Montag Nachmittag interessant. Diese geht darauf zurück, dass alle BEVs zu diesem Zeitpunkt unterwegs sind und die Haus-

batterien nicht genügend Kapazität besitzen, um den durch die erzeugte Solarenergie ausgelösten negativ Peak komplett auszugleichen.

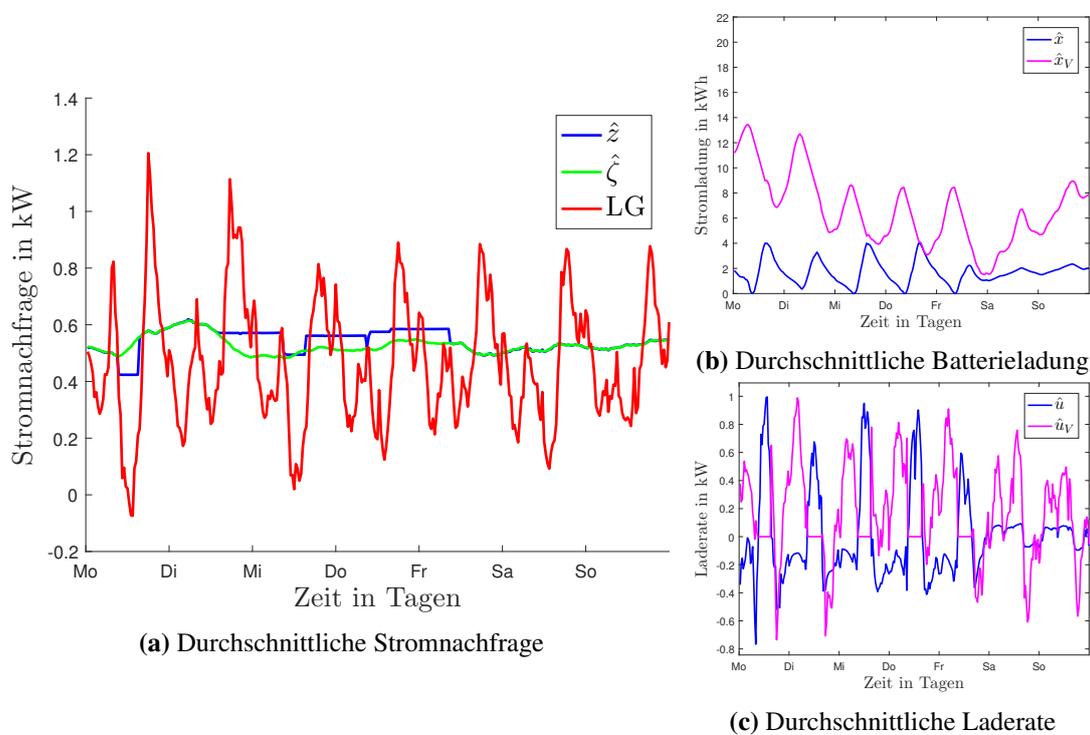


Abbildung 6.6: Simulation von 50 Haushalten über eine Woche mit 25 Elektroautos

Dieses Problem entsteht dadurch, dass wir bisher davon ausgegangen sind, dass BEVs solange vom Netz getrennt sind, bis deren Besitzer abends nach Hause kommen. Tatsächlich bieten einige Arbeitgeber jedoch bereits Parkplätze mit Ladestationen für Elektroautos an. Falls diese Ladestationen und die RES Teile desselben Smart-Grids sind, können wir die Unterwegs-Phasen drastisch verkürzen. Wir nehmen daher in der folgenden Simulation an, dass BEVs morgens und abends jeweils nur eine halbe Stunde unterwegs sind, anstatt mehrere Stunden vom Netz getrennt zu sein. Dadurch ermöglichen wir, dass nachmittags produzierte Solarenergie direkt in den größeren und leistungsfähigeren Elektroautobatterien gespeichert werden kann. Dass diese Änderung ein großes Potential besitzt, ist in dem vom DMPC berechneten Ergebnis in Abbildung 6.7 erkennbar. Die Nachfrage am Montag kann nun weit genug erhöht werden, um die angestrebte Menge zu erreichen. Außerdem fallen die Plateaus in der Nachfrage im Vergleich zu obiger Simulation niedriger aus bzw. verschwinden, wie z.B. dienstags und mittwochs, sogar gänzlich.

Sehr auffällig ist jedoch der hohe Peak am Mittwoch Nachmittag. Der Grund hierfür ist uns nicht klar. Wir stellen allerdings fest, dass der DMPC an manchen Zeitpunkten zu viele Iterationen benötigt, um eine gute und zulässige Lösung zu finden, und daher vorzeitig abbricht. Dies könnte daran liegen, dass der tägliche Verbrauch der BEVs von ca. 6 kWh, welcher vorher über mehrere Stunden verteilt von der Batterieladung abgezogen wurde, nun innerhalb von nur zwei Zeitschritten verloren geht. Außerdem

gelten zusätzliche Nebenbedingungen: Die BEVs fahren nun zweimal am Tag los. Das bedeutet, dass täglich einmal mehr eine Mindestladung eingehalten werden muss. Da sich die gefahrene Strecke pro Unterwegs-Phase jedoch halbiert, während die Reserveladung gleich bleibt, beträgt diese Mindestladung 5.75 kWh. Eventuell könnte dies der Grund dafür sein, dass der DMPC in 1000 Iterationen⁷ der inneren Schleife keine optimale Lösung findet. Um dieses Problem jedoch genauer zu verstehen, sind weitere Untersuchungen notwendig. Diese waren leider nicht mehr im Umfang dieser Arbeit umsetzbar.

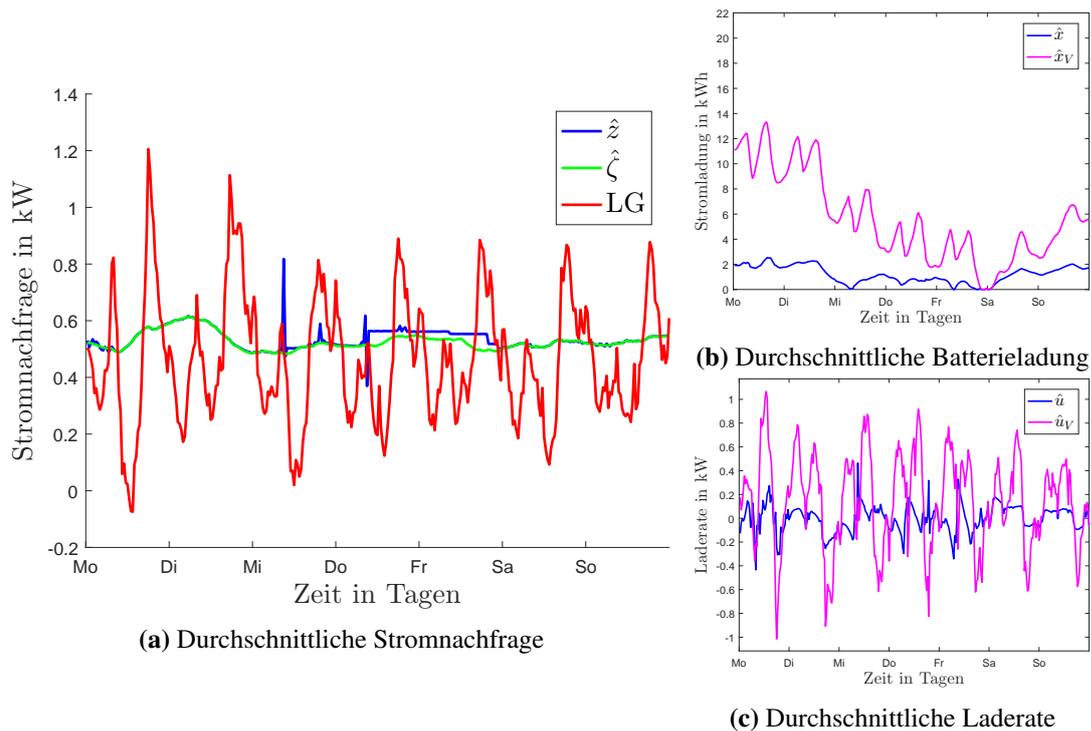


Abbildung 6.7: Simulation von 50 Haushalten über eine Woche mit 25 Elektroautos, wobei diese bei der Arbeitsstelle ans Netz angeschlossen sind

6.3.5 100 Prozent BEVs - Kombination von Haushaltsbatterie und BEV sinnvoll

Der DMPC Algorithmus stößt auch irgendwann an seine Grenzen. Abbildung 6.8 zeigt, dass bei der Optimierung erhebliche Probleme entstehen, wenn alle RES nur ein Elektroauto besitzen, aber keine Hausbatterie. In den Phasen, in denen die BEVs unterwegs sind, ist keine Steuerung des Systems möglich und daher entspricht die Nachfrage dem ungesteuerten Fall. Da dies jedoch nachmittags der Fall ist, kann die gewonnene Solarenergie nicht gespeichert werden. Über Nacht müssen die Batterien dann für den nächsten Tag aufgeladen werden, damit die Mindestladung von 8.75 kWh am Morgen erfüllt wird. Dies führt dazu, dass Plateaus mit hoher Nachfrage entstehen, zwischen

⁷In allen anderen Simulation wurde die obere Schranke von 300 Iterationen nie erreicht. Trotz des Erhörens dieses Werts traten hier jedoch Probleme auf.

welchen eine niedrige Nachfrage herrscht. Erst am Wochenende kann das System wieder gut gesteuert werden.

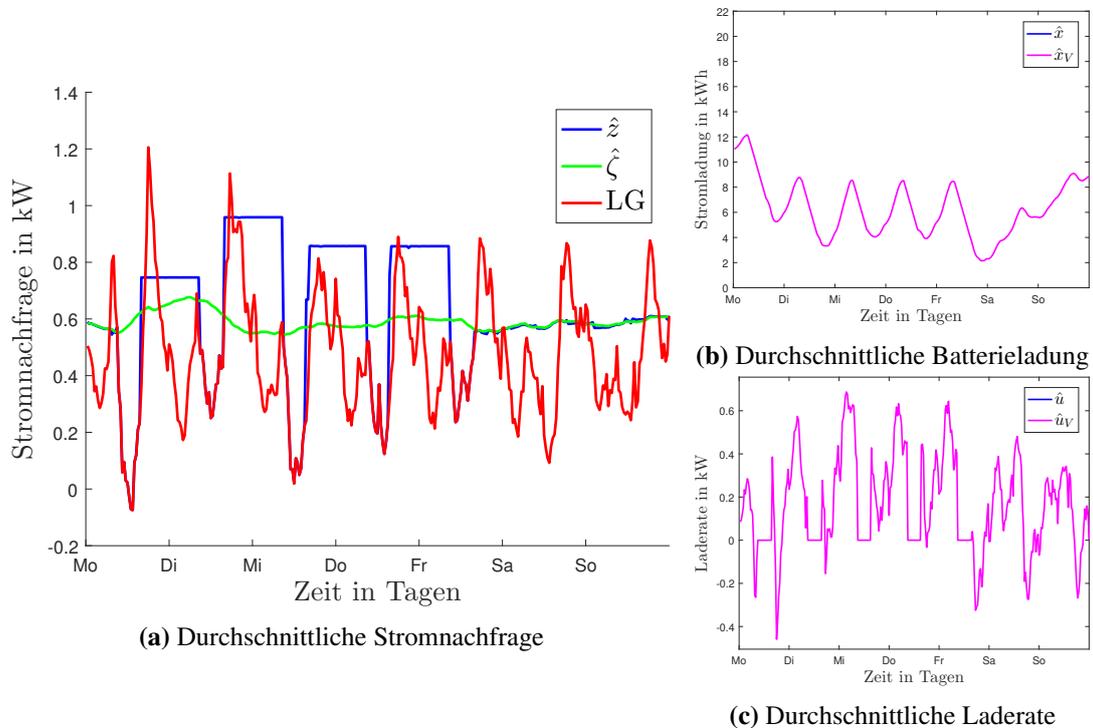


Abbildung 6.8: Simulation von 50 Haushalten über eine Woche ohne Hausbatterien und mit Elektroautos

In den bisherigen Simulationen haben wir nur RES betrachtet, die entweder eine Hausbatterie oder ein Elektroauto besitzen. In der Realität wird es jedoch häufig Fälle geben, in denen RES beides haben. In Abbildung 6.9 sehen wir den Extremfall, in dem alle RES zusätzlich ein Elektroauto nutzen. Auch hier sieht man, dass der Verbrauch der Elektroautos im Vergleich zu Größe und Stärke der Haushaltsbatterien zu hoch ist, um die gewünschte Nachfrage $\hat{\zeta}$ zu erfüllen. Jedoch liefert diese Simulation deutlich bessere Ergebnis als im Setting ohne Haushaltsbatterien. In Abbildung 6.9c sind die antizyklischen Laderaten von Haushaltsbatterien und Elektroautos nachmittags an Wochentagen sehr gut zu sehen.

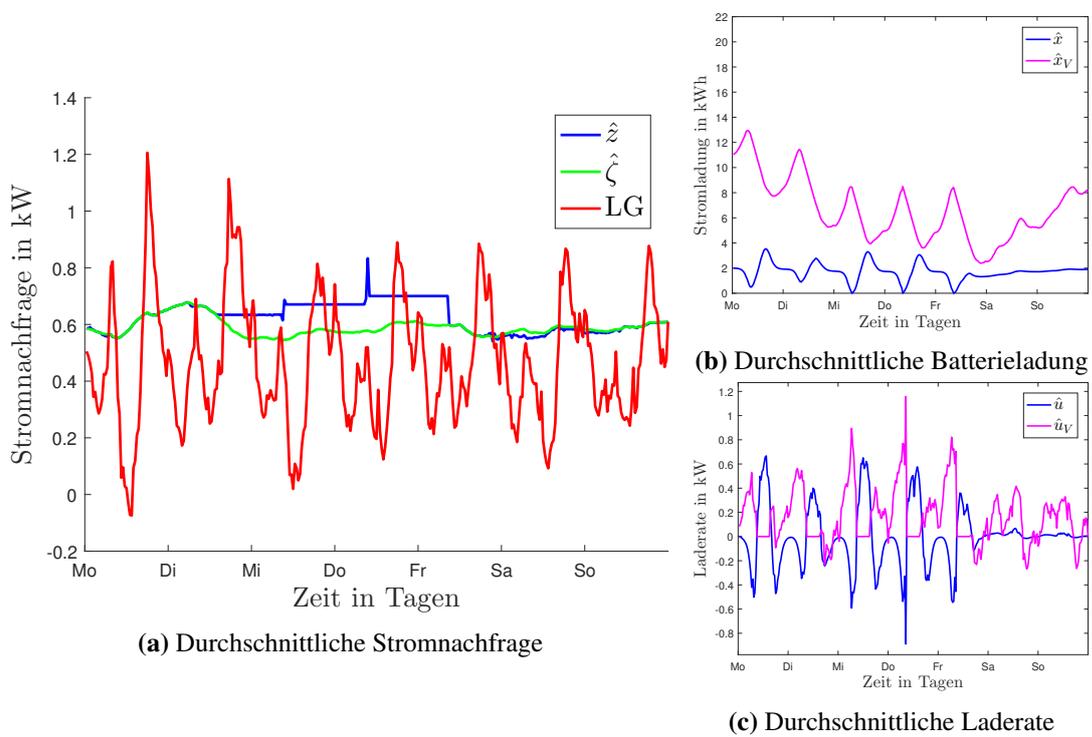


Abbildung 6.9: Simulation von 50 Haushalten über eine Woche mit Hausbatterien und Elektroautos

Fazit und Ausblick

7.1 MPC in Smart Grids als Chance für die Energiewende

Die Weiterentwicklung der Technik stellt die Menschheit immer wieder vor große Herausforderungen. So muss für eine umweltfreundlichere Zukunft eine Lösung für unsere Stromnetze gefunden werden. In dieser Arbeit haben wir gesehen, dass der Einsatz von Haushaltsstromspeichern zusammen mit einer intelligenten Steuerung des Stromnetzes einen Beitrag dazu liefern könnte. Um eine optimale Steuerung berechnen zu können, sind neben genauen Wettervorhersagen für die Stromerzeugung auch exakte Verbrauchsprognosen notwendig. Wir haben gezeigt, dass mithilfe des dargestellten Optimierungsalgorithmus aggregierte Verbrauchsvorhersagen, welche viel genauer sind als Prognosen für Einzelhaushalte, ausreichend sind. Außerdem liefert der DMPC auch bei gestörten Prognosen noch gute Ergebnisse.

Des Weiteren haben wir den bestehenden Algorithmus von Dr. Braun um den Einsatz von Elektroautos erweitert. In einem Modell, in dem achtzig Prozent der Haushalte eines Smart Grids Haushaltsbatterien und zwanzig Prozent der Haushalte BEVs als Zweitwagen nutzen, treten bereits Probleme auf, wenn die Elektroautos nicht in die Optimierung einbezogen werden. Erfolgt das Aufladen der Autobatterien ebenso kontrolliert wie das der Haushaltsbatterien, so könnte eine Überlastung des Stromnetzes bei einem Anstieg der Elektromobilität auch ohne einen umfangreichen Ausbau des Stromnetzes verhindert werden. Auch wenn für den Einsatz dieser Technik in der Realität noch viel getan werden muss, implizieren die Simulationsergebnisse, dass mit der Optimierung mittels MPC viel erreicht werden kann. Es bleibt jedoch die Aufgabe der Netzbetreiber oder des Gesetzgebers, Anreize zur Teilnahme an gesteuerten Smart Grids zu schaffen. So könnten beispielsweise Privatpersonen durch günstigere Strompreise belohnt werden, falls sie ihre Hausbatterie oder ihr Elektroauto durch einen Computer laden lassen, der in einem verteilten Netzwerk arbeitet und die Stromnachfrage im Gesamtnetz optimiert.

7.2 Ausblick auf künftige Arbeiten

In Kapitel 4 wurde das Verhalten des DMPCs bei fehlerhaften Verbrauchsprognosen untersucht. Der Algorithmus ist allerdings nicht speziell für den Gebrauch mit gestörten Daten ausgelegt. Das Ergebnis des Algorithmus könnte deutlich verbessert werden, wenn Techniken der robusten oder stochastischen modellprädiktiven Regelung verwendet werden, die z.B. in [26] beschrieben werden. Wie und ob sich dieses Vorgehen auf den vorgestellten DMPC übertragen lässt, ist noch genauer zu untersuchen.

In der vorliegenden Arbeit wurde ein Modell für Elektroautos aufgestellt, das wie in Bemerkung 3.1.3 angesprochen, bis auf die Kostenfunktion linear ist. Zusammen mit der quadratischen Kostenfunktion liegt ein sogenanntes *linear-quadratisches* System vor - eine Klasse von Kontrollproblemen, zu denen ausführliche Theorie existiert (z.B. in [11]). Es können jedoch in der Erweiterung genauso wie im Grundmodell aus [1] Verlustraten hinzugefügt werden, die das Problem auch in den Nebenbedingungen nichtlinear machen. Analysierbar wäre beispielsweise, wie sich diese Änderung auf die Laufzeit des DMPC auswirkt.

In den Simulationen wurde angenommen, dass Besitzer von Elektroautos regelmäßig eine bestimmte Strecke zur Arbeit fahren. In der Realität wird es jedoch öfter passieren, dass der Weg zur Arbeitsstelle ein anderer ist oder dass das Auto, z.B. im Krankheitsfall des Besitzers, gar nicht bewegt wird. Es wäre also sehr interessant zu untersuchen, wie robust der vorgestellte Algorithmus gegen Fehlinformationen bezüglich des tatsächlichen Stromverbrauchs der BEVs bzw. Unterwegs-Phasen ist. Außerdem kann die Frage geklärt werden, wieso der DMPC mit der Berechnung einer Lösung Probleme hat, falls die BEVs in der Arbeitsstelle ans Netz angeschlossen sind.

Wir stellen fest, dass das Potential des DMPCs bei der Minimierung von Stromschwankungen in Smart Grids noch weiter in Simulationen untersucht werden kann. Nichtsdestotrotz wäre es genauso interessant, zu sehen, ob und wie sich der DMPC in einem realen System umsetzen lassen würde.

Literaturverzeichnis

- [1] P. Braun; *Hierarchical distributed optimization and predictive control of a smart grid*; Universität Bayreuth, 2016.
- [2] Seite “*Die Beschlüsse des Bundestages am 30. Juni und 1. Juli*”; In: Textarchiv 2011, Deutscher Bundestag.
URL: https://www.bundestag.de/dokumente/textarchiv/2011/34915890_kw26_angenommen_abgelehnt/205788 (Abgerufen: 06.12.2018)
- [3] H. Wirth; *Aktuelle Fakten zur Photovoltaik in Deutschland*; S. 33-35; Fraunhofer ISE, 2018
- [4] H. Kuang, S. Li, Z. Wu; *Discussion on advantages and disadvantages of distributed generation connected to the grid*; 2011 International Conference on Electrical and Control Engineering. IEEE, 2011
- [5] Europäische Kommission; *Eine europäische Strategie für emissionsarme Mobilität*. In: Mitteilung der Kommission an das europäische Parlament, den Rat, den europäischen Wirtschafts- und Sozialausschuss und den Ausschuss der Regionen. Brüssel, 2016.
URL: https://eur-lex.europa.eu/resource.html?uri=cellar:e44d3c21-531e-11e6-89bd-01aa75ed71a1.0005.02/DOC_1&format=PDF (Abgerufen: 07.12.2018)
- [6] Artikel “*Elektromobilität in Deutschland*”. In: *Bundesministerium für Wirtschaft und Energie*. URL: <https://www.bmwi.de/Redaktion/DE/Dossier/elektromobilitaet.html> (Abgerufen: 07.11.2018)
- [7] Seite “*Energiespeicherung von morgen: Wissenschaftsministerin Marion Kiechle eröffnet das Bayerische Zentrum für Batterietechnik (BayBatt) an der Universität Bayreuth*”; In: *Universität Bayreuth*; Pressemitteilung vom 06.09.2018.
URL: <https://www.uni-bayreuth.de/de/universitaet/presse/pressemitteilungen/2018/109-Eroeffnung-BayBatt/index.html> (Abgerufen: 22.11.2018)
- [8] Oliver Wyman Strategieberatung; *E-Mobilität bedroht stabile Stromversorgung*, 2018

-
- [9] *Widerstand gegen Suedlink: Treffen von Stromtrassen-Gegnern*. In: *Süddeutsche Zeitung*;
URL: <https://www.sueddeutsche.de/news/wirtschaft/energie—bonn-widerstand-gegen-suedlink-treffen-von-stromtrassen-gegnern-dpa.urn-newsml-dpa-com-20090101-180322-99-587306> (Abgerufen: 18.11.2018)
- [10] L. Grüne. *Mathematische Kontrolltheorie II: Nichtlineare Systeme*; Vorlesungsskript, Universität Bayreuth, 2006.
- [11] E. Sontag; *Mathematical Control Theory*; Springer-Verlag New York Basel, 1998.
- [12] L. Grüne und J. Pannek; *Nonlinear Model Predictive Control*; Springer-Verlag London, 2017.
- [13] D. P. Bertsekas; *Dynamic Programming and Optimal Control*; Athena Scientific, Belmont, Mass., 2000.
- [14] P. Braun et al.; *A Distributed Optimization Algorithm for the Predictive Control of Smart Grids*; In *IEEE Transactions on Automatic Control*, Bd. 61 (2016) Heft 12, S. 3898-3911, 2015
- [15] D. Heinemann, E. Lorenz; *Vorhersage der Solarstrahlung und der Solarstromerzeugung*; In *Promet Jahrgang 39 Heft 3/4*; Deutscher Wetterdienst, Offenbach, 2015
- [16] S. Barker et al.; *SmartCap: Flattening peak electricity demand in smart homes*; In *Proc. of the International IEEE Conference on Pervasive Computing and Communications*, 2012
- [17] S. Humeau et al.; *Electricity Load Forecasting for Residential Customers: Exploiting Aggregation and Correlation between Households*; In *2013 Sustainable Internet and ICT for Sustainability (SustainIT)*; School of Computer and Communication Sciences École Polytechnique Fédérale de Lausanne (EPFL), 2018
- [18] Seite “Smoothing Splines”; In: *MathWorks Documentation*.
URL: <https://de.mathworks.com/help/curvefit/smoothing-splines.html>
(Abgerufen: 02.08.2018)
- [19] *Solar home energy data*; In: Ausgrid.
URL: <https://www.ausgrid.com.au/Industry/Innovation-and-research/Data-to-share/Solar-home-electricity-data> (Abgerufen: 27.11.2018)
- [20] Seite “sonnen eco 8/4”; In: *Enerkeep Echte unabhängige Solarberatung*.
URL: <https://enerkeep.com/DE/de/solarbatterie/sonnen-eco-8-4>
(Abgerufen: 06.11.2018)
- [21] *ADAC Autotest: Renault Zoe Intens*;
URL: https://www.adac.de/_ext/itr/tests/Autotest/AT5388_Renault_Zoe_Intens/Renault_Zoe_Intens.pdf (Abgerufen: 30.07.2018)

-
- [22] Seite “*Wie viel Sprit ist «auf Reserve» noch im Tank?*”; In: *Süddeutsche Zeitung*. Veröffentlicht am 14. Januar 2014, 10:51
URL: <https://www.sueddeutsche.de/news/leben/auto-wie-viel-sprit-ist-auf-reserve-noch-im-tank-dpa.urn-newsml-dpa-com-20090101-140114-99-02006>
(Abgerufen: 30.07.2018)
- [23] Seite “*Stromtankstellen Statistik für Deutschland*”; In: *GoingElectric*.
URL: <https://www.goingelectric.de/stromtankstellen/statistik/Deutschland/>
(Abgerufen: 30.07.2018)
- [24] Christian Wingerter; *Berufspendler: Infrastruktur wichtiger als Benzinpreis*; In: *STATmagazin: Arbeitsmarkt, 5/2014*; Statistisches Bundesamt, Wiesbaden, 2014.
- [25] Artikel “*Elektroauto zu Hause laden - die Voraussetzungen*”; In: *Stromschnell*.
URL: https://www.stromschnell.de/technik/elektroauto-zuhause-laden-voraussetzungen-und-kosten_5112770_5093776.html (Abgerufen: 14.11.2018)
- [26] B. Kouvaritakis und M. Cannon; *Model Predictive Control - Classical, Robust and Stochastic*; Springer International Publishing Switzerland, 2016.

Anhang

A Implementierung des DMPC in Matlab

Für alle Simulationen dieser Masterarbeit wurde ein selbstgeschriebenes Matlab-Programm verwendet. Wie die gesamte Arbeit, basiert auch das Programm auf einem von Dr. Braun geschriebenen Programm. Die größten Änderungen beinhalten das Aggregieren und Stören von Verbrauchsprognosen, wie in Kapitel 4 beschrieben, und die Erweiterung des Modells um den Einsatz von Elektroautos, wie in Kapitel 5 erklärt. Das Programm sowie einige Simulationsszenarien und Daten befinden sich auf der beigelegten CD.

A1 Bestandteile

Das Programm besteht im Wesentlichen aus 14 Dateien:

- *main_nmpc.m*: Das Main-Programm, in dem Einstellungen zur gewünschten Simulation getroffen werden und mit dem die Simulation gestartet wird.
- *nmpc_extended_n.m*: Die Hauptroutine, die sämtliche Initialisierungen und die MPC-Schleife enthält.
- *dynamic.m*: Eine Funktion, die die Systemdynamik realisiert.
- *loadLG.m*: Eine Subroutine, die Verbrauchs- und Erzeugungsdaten aus dem gegebenen Datensatz lädt.
- *globalOptimization.m*: Eine Funktion, welche die Optimierung der zentralen Einheit durchführt.
- *objectiveFunctionTheta.m*: Die Kostenfunktion der Optimierung der zentralen Einheit.
- *localOptimization.m*: Eine Funktion, welche die Optimierung eines Haushalts durchführt.
- *objectiveFunction.m*: Die Kostenfunktion der lokalen Optimierung.
- *inEqConstraints.m*: Eine Subroutine, welche die nichtlinearen Nebenbedingungen der lokalen Optimierung enthält.
- *hessinterior.m*: Eine Subroutine, in der die Hessematrizen der Kostenfunktion und Nebenbedingungen der lokalen Optimierung definiert sind.
- *visualization.m*: Eine Funktion, die die Live-Visualisierung der Open- und Closed-Loop Lösungen ermöglicht.

-
- *FeasibilityCheck.m*: Eine Subroutine, die eine gegebene Steuerung auf Zulässigkeit prüft.
 - *plot_output.m*: Eine Funktion, die einen durch Ausführen der *main_nmpc* Routine generierten Output plottet.
 - *LGData.mat*: Die Tabelle, die aufbereiteten Daten zur Stromerzeugung und -verbrauch von Haushalten enthält.

Im Ordner *examples* finden sich außerdem neben einigen Beispielsimulationen auch deren Output Daten sowie zugehörige Plots.

A2 Initialisierung der Simulation

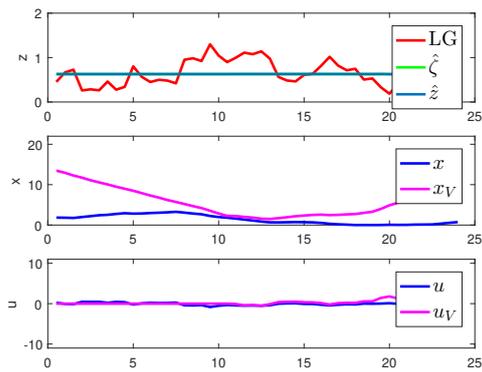
Um eine Simulation zu starten, müssen in *main_nmpc.m* einige Einstellungen festgelegt werden, welche in einem Struct namens *input* abgelegt werden. Im oberen Teil des Programms können Basiseinstellungen verändert werden: Zuallererst legt man die Länge eines Diskretisierungsintervalls T in Stunden (Defaultwert: 0.5), die Länge des MPC Horizonts N sowie die Anzahl an ganzen Tagen *wholedays*, für welche die optimale Lösung berechnet werden soll, fest. Alternativ kann die genaue Anzahl an Optimierungsschritten \mathcal{N} mit *MPCiterations* ausgewählt werden. Anschließend kann bestimmt werden, wieviele Haushalte nur eine Hausbatterie (*numHH*), nur ein Elektroauto (*numEV*) oder beides (*numHEV*) besitzen.

Die beiden Parameter *hcap* und *ecap* regeln die maximale Kapazität der Haus- bzw. Elektroautobatterien. Für die Laderaten sind analog die beiden Werte *hcha* und *echa* zuständig. Die Stromladung zu Beginn der Optimierung kann mit *startchargeH* bzw. *startchargeE* prozentual festgelegt werden. Die prozentuale Minimalladung, die Elektroautos beim Losfahren besitzen müssen, wird über den Parameter *eoffmin* geregelt. Mit den Parametern *minstartwork*, *maxstartwork*, *minworkingtime* und *maxworkingtime* werden die Arbeitszeiten der Elektroautobesitzer und damit die Unterwegs-Phasen der BEVs gesteuert. Wie viel Strom ein Elektroauto am Tag im Durchschnitt verbrauchen soll, kann mit *avgdailycons* festgelegt werden.

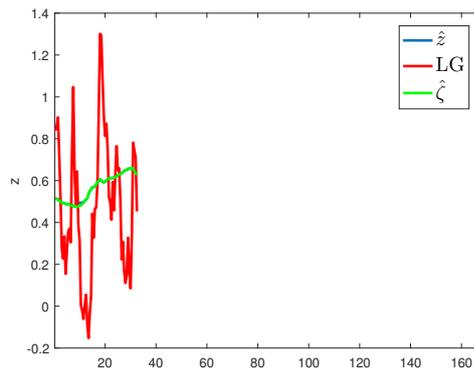
Im zweiten Abschnitt des Programms können genauere Einstellungen festgelegt werden. Allerdings ist hierfür ein besseres Verständnis des Programms notwendig, damit nicht durch eine falsche Eingabe Fehler erzeugt werden. Falls sich beispielsweise einzelne Haushalte unterscheiden sollen, können die Kapazitäten und Laderaten dort direkt (über die Parameter C_u , e_C_u , R_l , R_u , e_R_l , e_R_u) angepasst werden.

A3 Start der Simulation und Ausgabe

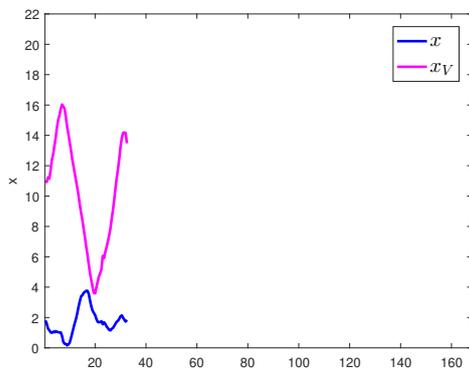
Nachdem alle Einstellungen festgelegt wurden, muss nur die *main_nmpc.m* Datei ausgeführt werden. Das Programm startet automatisch einen *parallel pool* der *Parallel Computing Toolbox*, um die lokalen Optimierungen zu beschleunigen. Sollte diese Toolbox nicht vorhanden sein, so muss in der Datei *nmpc_extended_n.m* die *parfor*-Schleife durch eine einfache *for*-Schleife ersetzt werden. Danach beginnt die erste Optimierung. Im Command Window wird die Dauer der inneren Schleifendurchläufe aus-



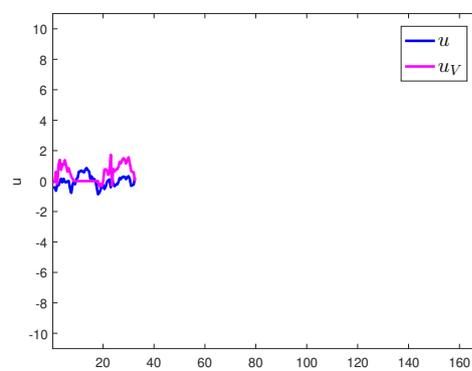
(a) Visualisierung der Open-Loop Lösung



(b) Closed-Loop Stromnachfrage in kW



(c) Closed-Loop Stromladung in kWh



(d) Closed-Loop Laderate in kW

Abbildung 7.1: Live Visualisierung des NMPC Programms

gegeben. Sobald eine der Abbruchbedingungen des Algorithmus 4 erfüllt ist, bricht die innere Schleife ab und der Algorithmus verschiebt den Horizont um eine Zeiteinheit. In der Ausgabe ist der aktuelle Zeitschritt, die Anzahl der nötigen inneren Schleifendurchläufe sowie die bisher vergangene Zeit zu sehen. Gleichzeitig öffnen sich 4 Plots. Wie in Abbildung 7.1a dargestellt, sind die Stromnachfrage, der Ladestand und die Laderaten der aktuellen Open-Loop Lösung zu sehen. Die Abbildungen 7.1b, 7.1c und 7.1d zeigen, wie die Stromnachfrage, der Ladestand und die Laderaten der iterativ erzeugten Closed-Loop Lösung visualisiert werden.

Nachdem die Optimierung abgeschlossen wurde, findet sich die Ausgabe des Programms in der Tabelle *out*. Die zur Closed-Loop Lösung gehörenden Daten sind sowie weitere Daten aus der Optimierung in den Feldern Z , X , e_X , U und e_U abgespeichert. Neben den in Abschnitt 3.1.2 definierten Performance Werten finden sich beispielsweise auch die Anzahl der benötigten inneren Schleifen Durchläufe $NIter$. Der für die Simulation verwendete *input*-Struct ist ebenfalls im Output enthalten.

Eidesstattliche Erklärung

Ich versichere hiermit an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, ganz oder in Teilen noch nicht als Prüfungsleistung vorgelegt und keine anderen als die angegebenen Hilfsmittel benutzt habe.

Sämtliche Stellen der Arbeit, die anderen Werken im Wortlaut oder dem Sinn nach entnommen sind, habe ich durch Quellenangaben kenntlich gemacht. Dies gilt auch für Zeichnungen, Skizzen, bildliche Darstellungen und dergleichen sowie für Quellen aus dem Internet.

Mir ist bewusst, dass jedes Zuwiderhandeln als Täuschungsversuch zu gelten hat und zu einer Bewertung der Arbeit mit "nicht ausreichend" führt.

Ort, Datum

Unterschrift