

Philipp Braun* und Lars Grüne

Verteilte Optimierung: Anwendungen in der Modellprädiktiven Regelung

Zusammenfassung: Verteilte Optimierungsverfahren wie die duale Dekomposition oder die Alternating Direction Method of Multipliers (ADMM) erleben in den letzten Jahren ein erneutes steigendes Interesse in den unterschiedlichsten Anwendungen. Die zunehmende Vernetzung von Servern oder Mikrocontrollern weltweit sowie die Größe von heutigen Datensätzen liefern dabei die Grundlage für die Nachfrage nach iterativen, parallelisierbaren Optimierungsverfahren. In dieser Arbeit stellen wir verteilte Optimierungsalgorithmen und ihre Anwendungen bei der Berechnung von Zustandsrückführungen mithilfe der Modellprädiktiven Regelung vor. Wir konzentrieren uns auf die Systemdynamik sowie die Vernetzung der Systeme bei der Anwendbarkeit der Algorithmen. Darüber hinaus untersuchen wir die Algorithmen auf ihre Kommunikationsstruktur, den Austausch sensibler Daten, die Skalierbarkeit und die Flexibilität.

1 Einleitung

Verteilte Optimierungsverfahren wie die duale Dekomposition oder die Alternating Direction Method of Multipliers (ADMM) erleben in den letzten Jahren ein erneutes steigendes Interesse in den unterschiedlichsten Anwendungen. Ursprünge der dualen Dekomposition gehen dabei auf Arbeiten aus den 1960er Jahren zurück [13, 3, 16], und ADMM geht aus Weiterentwicklungen in den späten 1970er [20] und aus den 1980er Jahren [18, 19] hervor. Während die Methoden sowie zugehörige Konvergenzeigenschaften schon seit längerem bekannt sind, scheint sich ihr Potential erst heute zu entfalten. Die einfache Vernetzung von Servern weltweit bietet dabei die Grundlage für die Implementierung von iterativen parallelisierbaren Optimierungsalgorithmen, während Fragestellungen aus den Bereichen Maschinelles Lernen und der Bildverarbeitung mit immer größeren Datensätzen für einen steigenden Bedarf sorgen.

In dieser Arbeit stellen wir das Potential verteilter Optimierungsalgorithmen in der Modellprädiktiven Rege-

lung (MPC) vor. Hierbei legen wir den Fokus nicht auf sehr große Optimierungsprobleme, die möglichst geschickt aufgeteilt werden sollen. Stattdessen betrachten wir dynamische Systeme, die über ihre Systemdynamik und/oder eine Zielfunktion miteinander gekoppelt sind. Beim verteilten Lösen der zugehörigen Optimalsteuerungsprobleme im MPC-Algorithmus ist daher die Struktur der Aufteilung über die Systeme bereits vorgegeben. Wir gehen davon aus, dass jedes Einzelsystem einen eigenen Regler zur Berechnung der Zustandsrückführung mittels MPC besitzt, jedoch der Austausch von Informationen zum Erreichen von Optimalität und Zulässigkeit notwendig ist. Wir stellen hier beispielhaft die duale Dekomposition, ADMM sowie primale verteilte Ansätze im Bezug auf MPC vor. Zur Illustration der Konzepte verweisen wir auf weiterführende Literatur aus dem Bereich Optimierung und MPC. Beim Einbetten der Optimierungsalgorithmen in MPC steht vor allem die Plug-and-Play Fähigkeit, das heißt die Flexibilität bezüglich der Systemdynamik und der Zielfunktion, sowie die Skalierbarkeit und die Kommunikationsstruktur im Vordergrund. Bei der Kommunikation ist entscheidend, welche Informationen zwischen den Systemen ausgetauscht werden müssen und welche Daten geheim bleiben können.

Als Referenz für (die verteilte) Optimierung verweisen wir auf [4]. Für ADMM im speziellen verweisen wir zusätzlich auf die Arbeiten [7, 33]. Für allgemeine Resultate der Modellprädiktiven Regelung beziehen wir uns auf die Monographien [24, 34]. In Abschnitt 2 stellen wir zunächst die Grundform von MPC vor, und führen das Optimalsteuerungsproblem ein, das der MPC-Formulierung zugrunde liegt. Zum Lösen des Optimalsteuerungsproblems diskutieren wir in Abschnitt 3, dem Hauptteil der Arbeit, die duale Dekomposition, ADMM sowie primale Optimierungsansätze und ihre Erweiterungen. Zum Abschluss der Arbeit erläutern wir in Abschnitt 4 Eigenschaften wie Warm-Start in der Anwendung im geschlossenen Regelkreis.

Wir verwenden die folgende Notationen: Für die Zahlen von 1 bis $i \in \mathbb{N}$ verwenden wir $\mathbb{N}_i := \{1, \dots, i\}$. Genauso verwenden wir $\mathbb{N}_{i,j} = \{i, \dots, j\}$ für $i, j \in \mathbb{N}$. Für Vektoren a_i und Matrizen A_i , $i \in \mathbb{N}_{\mathcal{I}}$ verwenden wir die Notation $a = (a_1^T, \dots, a_{\mathcal{I}}^T)^T$ und $A = (A_1, \dots, A_{\mathcal{I}})$. Für eine Folge verwenden wir die Notation $(a_i)_{i \in \mathbb{N}}$ und $(a_i)_{i \in \mathbb{N}_{j,j+k}}$, $j, k \in \mathbb{N}$ bezeichnet Folgenglieder von j bis k . $\dim(a) \in \mathbb{N} \cup \{0\}$ gibt die Länge eines Vektors an.

*Korrespondenzautor: Philipp Braun, University of Newcastle, School of Electrical Engineering and Computing, Newcastle, Australia, E-Mail: philipp.braun@newcastle.edu.au
Lars Grüne, Universität Bayreuth, Lehrstuhl für Angewandte Mathematik, Bayreuth, Germany, E-Mail: lars.gruene@uni-bayreuth.de

2 Modellprädiktive Regelung gekoppelter Systeme

2.1 Systemdynamik

Wir betrachten $\mathcal{I} \in \mathbb{N}$ zeitdiskrete dynamische Systeme

$$x_i(k+1) = f_i(x_i(k), u_i(k), y_i(k)) \quad (1a)$$

mit Zuständen $x_i \subset \mathbb{X}_i$, Eingängen $u_i \subset \mathbb{U}_i$, und Kopplungsvariablen $y_i = \{y_{i,j} | j \in \mathbb{N}_{\mathcal{I}} \setminus \{i\}\}$. Die Mengen \mathbb{X}_i und \mathbb{U}_i definieren Beschränkungen der Variablen passender Dimension. Wir verwenden die Notation $\mathbb{X} = \mathbb{X}_1 \times \dots \times \mathbb{X}_{\mathcal{I}}$ und $\mathbb{U} = \mathbb{U}_1 \times \dots \times \mathbb{U}_{\mathcal{I}}$. Die Einzelsysteme sind über die Variablen $y_{i,j}$, $i, j \in \mathbb{N}_{\mathcal{I}}$, $i \neq j$, miteinander verbunden, wobei $y_{i,j}$ die Kopplung von System j zu System i beschreibt. Dabei machen wir die Annahme, dass $y_{i,j}(k)$, $j \in \mathbb{N}_{\mathcal{I}} \setminus \{i\}$, linear von $x_i(k)$, $u_i(k)$ sowie von weiteren externen bekannten Parametern $v_{i,j}(k)$ abhängen kann. Das heißt, für alle $i, j \in \mathbb{N}_i$, $i \neq j$ ist System i mit System j über

$$y_{i,j}(k) = C_{i,j}x_i(k) + D_{i,j}u_i(k) + v_{i,j}(k). \quad (1b)$$

gekoppelt. Wir weisen explizit darauf hin, dass $y_{i,j}$ keine Ausgänge bezeichnen. Um unnötige Notation zu vermeiden, verzichten wir hier auf die Angabe der Dimension der Variablen. Wir weisen aber darauf hin, dass alle Variablen unterschiedliche Dimensionen aus $\mathbb{N} \cup \{0\}$ besitzen können.

Zur Beschreibung der Vernetzung zwischen den Einzelsystemen können gerichtete Graphen $G(\mathbb{N}_{\mathcal{I}}, \Gamma)$ mit den Systemen $i \in \mathbb{N}_{\mathcal{I}}$ als Knoten und Kanten $\Gamma = \{(i, j) | i, j \in \mathbb{N}_{\mathcal{I}}, i \neq j, \dim(y_{i,j}) \geq 1\}$ verwendet werden. Unterschiedliche Strukturen sind in Abb. 1 dargestellt.

Beispiel 2.1. *Das lineare System*

$$x_1(k+1) = x_1(k) + 2x_2(k) + u_1(k) \quad (2a)$$

$$x_2(k+1) = 3x_2(k) + u_1(k) + u_2(k) \quad (2b)$$

kann alternativ durch die Gleichungen

$$x_1(k+1) = x_1(k) + u_1(k) + 2y_{2,1}(k), \quad y_{2,1}(k) = x_2(k)$$

$$x_2(k+1) = 3x_2(k) + u_2(k) + y_{1,2}(k), \quad y_{1,2}(k) = u_1(k)$$

in der Form (1) beschrieben werden.

Beispiel 2.2. *Die linearen Systeme*

$$x_i(k+1) = x_i(k) + u_i(k), \quad (3a)$$

$$y_{i,\mathcal{I}}(k) = u_i(k) + v_{i,\mathcal{I}}(k), \quad (3b)$$

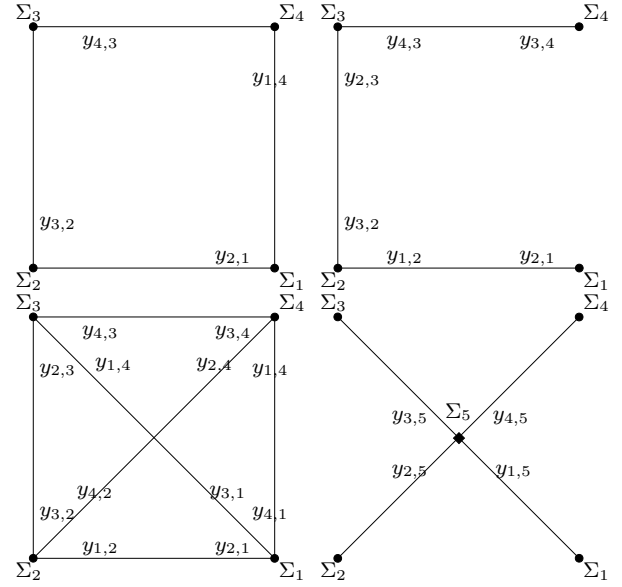


Abb. 1: Visualisierung der Vernetzung unterschiedlicher Systeme Σ_i , $i \in \mathbb{N}_{\mathcal{I}}$.

für $i \in \mathbb{N}_{\mathcal{I}-1}$ beschreiben die hierarchische Struktur in Abb. 1 unten rechts. Alle Systeme $i \in \mathbb{N}_{\mathcal{I}-1}$ sind mit einem zentralen System \mathcal{I} ohne eigene Dynamik verbunden. Varianten von (3) werden zum Beispiel in [2, 27, 17, 9] verwendet um Elektrizitätsnetze zu modellieren. Hierbei beschreibt $y_i(k)$ den Energieverbrauch zum Zeitpunkt k , wobei der Grundbedarf $v_{i,\mathcal{I}}(k)$ durch Laden/Entladen einer Speichereinheit mit Zustand $x_i(k)$ über $u_i(k)$ beeinflusst werden kann.

2.2 Modellprädiktive Regelung

Die Modellprädiktive Regelung (MPC) ist ein Verfahren zur (schrittweisen) Berechnung von Zustandsrückführungen. In jedem Zeitpunkt k wird dafür ein Optimalsteuerungsproblem (OCP) (in der Regel auf endlichem Zeithorizont) gelöst. Das OCP ist über laufende Kosten $(x_i, u_i, y_i) \mapsto \ell_i(x_i, u_i, y_i) \in \mathbb{R}$, $i \in \mathbb{N}_{\mathcal{I}}$, einen gegebenen Prädiktionshorizont $K \in \mathbb{N}_{\geq 2}$ und die Systemdynamik (1) definiert. Zum Zeitpunkt $k \in \mathbb{N}$ und einem gegebenen Anfangswert x_0 sowie bekannten externen Signalen $(v(k))_{k \in \mathbb{N}}$ wird das folgende OCP verwendet

$$\min_{\substack{u_i(t), x_i(t), y_i(t) \\ i \in \mathbb{N}_{\mathcal{I}}, t \in \mathbb{N}_{k, k+K-1}}} \sum_{t=k}^{k+K-1} \left(\sum_{i=1}^{\mathcal{I}} \ell_i(x_i(t), u_i(t), y_i(t)) \right) \quad (4a)$$

$$\text{s.t. } x_i(k) = x_{i,0}, \quad (4b)$$

$$x_i(t+1) = f_i(x_i(t), u_i(t), y_i(t)), \quad (4c)$$

$$y_{i,j}(t) = C_{i,j}x_i(t) + D_{i,j}u_i(t) + v_{i,j}(t), \quad (4d)$$

$$x_i(t+1) \in \mathbb{X}_i, \quad u_i(t) \in \mathbb{U}_i, \quad (4e)$$

$$\forall i \in \mathbb{N}_{\mathcal{I}}, \forall j \in \mathbb{N}_{\mathcal{I}} \setminus \{i\}, \forall t \in \mathbb{N}_{k, k+K-1}.$$

Wir verzichten bei der Formulierung auf Endbedingungen und Endbeschränkungen (siehe zum Beispiel [34], [24]). Diese lassen sich jedoch in das OCP integrieren. Wir nehmen an, dass (nicht notwendigerweise eindeutige) Optimallösungen $u_i^*(t)$, $(x_i^*(t), y_i^*(t))$, $i \in \mathbb{N}_{\mathcal{I}}$, $t \in \mathbb{N}_{k, k+K-1}$ zu jedem Zeitpunkt $k \in \mathbb{N}$ existieren. Zur Berechnung einer Zustandsrückführung $\mu : \mathbb{X} \rightarrow \mathbb{U}$ mittels MPC kann damit die folgende Iterationsvorschrift definiert werden:

1. Für $k \in \mathbb{N}$, einen Prädiktionshorizont $K \in \mathbb{N}$, $(v(t))_{t \in \mathbb{N}_{k, k+K-1}}$ und $x_0 = x(k)$ löse das OCP (4).
2. Definiere $\mu(x(k)) = u^*(k)$ und berechne $x(k+1)$ unter Verwendung des Eingangs $\mu(x(k))$.
3. Setze $k = k + 1$ und gehe zu Schritt 1.

Arbeiten die sich mit MPC-Algorithmen beschäftigen konzentrieren sich in der Regel auf die Regelgüte der Zustandsrückführung μ , Zulässigkeit des OCPs (4) im geschlossenen Regelkreis oder auf Algorithmen zur effizienten Berechnung von $u_i^*(t)$, $i \in \mathbb{N}_{\mathcal{I}}$, $t \in \mathbb{N}_{k, k+K-1}$. Wie bereits in der Einleitung motiviert konzentrieren wir uns in dieser Arbeit auf verteilte Optimierungsverfahren zur Berechnung von u^* für die wir zunächst die Notation des OCPs (4) vereinfachen.

2.3 Das Optimalsteuerungsproblem

Um die Notation des OCPs zu vereinfachen konzentrieren wir uns auf einen festen Zeitschritt $k \in \mathbb{N}$ und verzichten auf die explizite Darstellung des Zeitindex. Zunächst fassen wir die Unbekannten des OCPs für $i \in \mathbb{N}_{\mathcal{I}}$ in einer Variablen, ohne spezielle Reihenfolge, zusammen

$$z_i \doteq \{[x_i(t), u_i(t), y_i(t)] \mid t \in \mathbb{N}_{k, k+K-1}\}. \quad (5)$$

Damit lassen sich die laufenden Kosten über Funktionen der Form

$$\phi_i(z_i) = \sum_{t=k}^{k+K-1} \ell_i(x_i(t), u_i(t), y_i(t)),$$

$i \in \mathbb{N}_{\mathcal{I}}$, vereinfachen. Für die Nebenbedingungen lassen sich die Mengen

$$\mathbb{D}_i = \left\{ z_i \mid \begin{array}{l} x_i(k) = x_{i,0} \\ x_i(t+1) = f_i(x_i(t), u_i(t), y_{j,i}(t)) \\ x_i(t+1) \in \mathbb{X}_i, u_i(t) \in \mathbb{U}_i \\ \forall i \in \mathbb{N}_{\mathcal{I}}, \forall t \in \mathbb{N}_{k, N-1} \end{array} \right\} \quad (6)$$

definieren und aus (4d) ergeben sich lineare Kopplungsnebenbedingungen $\sum_{i=1}^{\mathcal{I}} E_i z_i - b = 0$ für geeignete Matrizen E_i , $i \in \mathbb{N}_{\mathcal{I}}$, und einen Vektor b . Insgesamt ergibt sich das Optimierungsproblem

$$\min_{z_i \in \mathbb{D}_i} \sum_{i=1}^{\mathcal{I}} \phi_i(z_i) \quad (7a)$$

$$\text{s.t.} \quad \sum_{i=1}^{\mathcal{I}} E_i z_i - b = 0 \quad (7b)$$

für das wir in Abschnitt 3 Ideen der verteilten Optimierung vorstellen.

Bemerkung 2.1. Die Formulierung des Optimierungsproblems (7) kann auch auf andere Weise erfolgen. Es gibt zum Beispiel für MPC für lineare Systeme gängige Formulierungen mit und ohne x als Unbekannte im Optimierungsproblem (siehe zum Beispiel [34]). Dieser Unterschied spielt für diese Arbeit keine Rolle und wirkt sich nur auf die Definition (insbesondere die Dimension) der Variablen z_i , $i \in \mathbb{N}_{\mathcal{I}}$, aus.

Beispiel 2.3. Wir betrachten erneut das lineare System (2). Für das OCP mit den laufenden Kosten $\ell_i(x_i, u_i, y_i) = x_i^2 + u_i^2$, $i = 1, 2$, und $K = 2$ definieren wir die Unbekannten (o.B.d.A. für $k = 0$)

$$z_i = (x_i(0), u_i(0), y_{j,i}(0), x_i(1), u_i(1), y_{j,i}(1))^T,$$

$i = 1, 2$, $j \in \{1, 2\} \setminus \{i\}$, und die lokalen Nebenbedingungen

$$\mathbb{D}_1 = \left\{ z_1 \in \mathbb{R}^6 \mid \begin{array}{l} z_1 \in \mathbb{X}_1 \times \mathbb{U}_1 \times \mathbb{R} \times \mathbb{X}_1 \times \mathbb{U}_1 \times \mathbb{R} \\ 0 = -z_{1,1} - z_{1,2} - 2z_{1,3} + z_{1,4} \\ x_{1,0} = z_{1,1}, z_{1,4} + z_{1,5} + 2z_{1,6} \in \mathbb{X}_1 \end{array} \right\},$$

$$\mathbb{D}_2 = \left\{ z_2 \in \mathbb{R}^6 \mid \begin{array}{l} z_2 \in \mathbb{X}_2 \times \mathbb{U}_2 \times \mathbb{R} \times \mathbb{X}_2 \times \mathbb{U}_2 \times \mathbb{R} \\ 0 = -3z_{2,1} - z_{2,2} - z_{2,3} + z_{2,4} \\ x_{2,0} = z_{2,1}, 3z_{2,4} + z_{2,5} + z_{2,6} \in \mathbb{X}_2 \end{array} \right\}.$$

Für die Nebenbedingungen (7b) definieren wir die Matrizen

$$E_1 = \begin{pmatrix} 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$E_2 = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \end{pmatrix}$$

und den Vektor $b = (0, 0, 0, 0)^T$. Aus den laufenden Kosten erhalten wir die Zielfunktion $\phi_i(z_i) = z_{i,1}^2 + z_{i,2}^2 + z_{i,4}^2 + z_{i,5}^2$, $i = 1, 2$, für das Optimierungsproblem (7).

Beispiel 2.4. Für das System aus Beispiel 2.2 kann das künstliche System \mathcal{I} zum Koppeln der Variablen $y_{i,\mathcal{I}}$, $i \in \mathbb{N}_{\mathcal{I}-1}$, in der Zielfunktion genutzt werden. Für den Prädiktionshorizont $K = 1$ und die Definition

$$z_{\mathcal{I}} = (y_{1,\mathcal{I}}, \dots, y_{\mathcal{I}-1,\mathcal{I}})^T, \quad z_{\mathcal{I}} \mapsto \phi_{\mathcal{I}}(z_{\mathcal{I}}), \quad (8)$$

der Gesamtverbrauch im Netzwerk optimiert werden [8, 9]. Alternativ kann zum Beispiel über die Definition

$$z_{\mathcal{I}} = \sum_{i=1}^{\mathcal{I}-1} y_{i,\mathcal{I}}, \quad \phi_{\mathcal{I}}(z_{\mathcal{I}}) = \begin{cases} 0, & \text{für } z_{\mathcal{I}} = 0 \\ \infty, & \text{für } z_{\mathcal{I}} \neq 0 \end{cases} \quad (9)$$

sichergestellt werden, dass Stromverbrauch und Stromerzeugung im Gesamtsystem übereinstimmen (siehe zum Beispiel [2]). Die Bedingung der Indikatorfunktion (9) führt auf die lineare Nebenbedingung $\sum_{i=1}^{\mathcal{I}-1} y_{i,\mathcal{I}} = z_{\mathcal{I}} = 0$ der Form (7b) im OCP. Für $K > 1$ lassen sich die Ideen für jeden Zeitschritt im Prädiktionshorizont verallgemeinern.

3 Verteilte Optimierung

In diesem Kapitel stellen wir Ideen der verteilten Optimierung zum Lösen von Optimierungsproblemen der Form (7) vor. Zur vereinfachten Darstellung nutzen wir die zusätzlichen Definitionen $\mathbb{D} = \mathbb{D}_1 \times \dots \times \mathbb{D}_{\mathcal{I}} \subset \mathbb{R}^n$ und $\mathbb{G} = \{z \in \mathbb{R}^n \mid Ez - b = 0\} \subset \mathbb{R}^n$. Damit kann das Optimierungsproblem (7) kompakt durch

$$\min_{z \in \mathbb{D} \cap \mathbb{G}} \phi(z) \quad (10)$$

beschrieben werden.¹ Im Gegensatz zum vorherigen Kapitel geben wir in diesem Abschnitt Dimensionen an, da sie für die Kommunikation zwischen den Systemen entscheidend sind. Genauso wie bei der MPC-Formulierung (4) gehen wir davon aus, dass das Optimierungsproblem (7) zulässig ist, d.h. es gibt $z \in \mathbb{D} \cap \mathbb{G}$. Beim Vorstellen der Algorithmen sollen vor allem grundlegende Ideen der verteilten Optimierung dargestellt werden. Für Details und Erweiterungen verweisen wir auf Literatur.

3.1 Duale Dekomposition

Ein einfaches Verfahren zum Lösen des Optimierungsproblems (10) basiert auf der dualen Dekomposition, einem dualen Gradientenverfahren. Durch Einführen von Lagrange-Multiplikatoren $\lambda \in \mathbb{R}^m$ und Definition der Lagrange-Funktion $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}$,

$$\begin{aligned} \mathcal{L}(z, \lambda) &= \phi(z) + \lambda^T (Ez - b) \\ &= \sum_{i=1}^{\mathcal{I}} (\phi_i(z_i) + \lambda^T E_i z_i) - \lambda^T b, \end{aligned} \quad (11)$$

lassen sich die linearen Nebenbedingungen in die Zielfunktion aufnehmen. Der Vorteil der Lagrange-Funktion verglichen mit dem Optimierungsproblems (10) ist, dass für feste λ , die Summenterme einzeln optimiert werden können. Mithilfe der Lagrange-Funktion lässt sich implizit

¹ Alternativ findet sich in der Literatur auch die Schreibweise $\min_{z \in \mathbb{R}^n} \phi(z) + I_{\mathbb{D} \cap \mathbb{G}}(z)$ wobei $I_{\mathbb{D} \cap \mathbb{G}} : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{\infty\}$ eine erweiterte reellwertige Indikatorfunktion mit $I_{\mathbb{D} \cap \mathbb{G}}(z) = 0$ für $z \in \mathbb{D} \cap \mathbb{G}$ und $I_{\mathbb{D} \cap \mathbb{G}}(z) = \infty$ sonst, definiert.

die duale Funktion $\psi : \mathbb{R}^m \rightarrow \mathbb{R} \cup \{-\infty\}$,

$$\psi(\lambda) = \inf_{z \in \mathbb{D}} \mathcal{L}(z, \lambda), \quad (12)$$

und das duale Problem

$$\max_{\lambda \in \mathbb{R}^m} \psi(\lambda) \quad (13)$$

formulieren. Hierbei ist ψ eine erweiterte reellwertige Funktion, da das Infimum nicht für alle $z \in \mathbb{D}$ existieren muss. Zwischen dem Optimierungsproblem (10), der Lagrange-Funktion (11) und dem dualen Problem (13) bestehen die folgenden Zusammenhänge. Die zugehörigen Beweise können zum Beispiel in [5, Appendix C] gefunden werden.

Theorem 3.1 (Dualität). *Falls das primale Problem (10) eine Optimallösung z^* besitzt, dann besitzt auch das duale Problem (13) eine Optimallösung λ^* und die Zielfunktionswerte stimmen überein, d.h.*

$$\phi(z^*) = \phi^* = \psi^* = \psi(\lambda^*).$$

Theorem 3.2 (Sattelpunkte). *Für das primale Problem (10) und das duale Problem (13) gilt die folgende Beziehung: z^* ist optimal für (10) und λ^* ist optimal für (13) genau dann wenn die Sattelpunkt Bedingung*

$$\mathcal{L}(z^*, \lambda) \leq \mathcal{L}(z^*, \lambda^*) \leq \mathcal{L}(z, \lambda^*) \quad (14)$$

für alle $z \in \mathbb{D}$ und für alle $\lambda \in \mathbb{R}^m$ erfüllt ist.

Aus Theorem 3.2 folgt, dass basierend auf dem Wissen von λ^* ein z^* berechnet werden kann. Durch die Struktur der Lagrange-Funktion kann hierbei z_i^* , $i \in \mathcal{N}_{\mathcal{I}}$ parallel durch Minimierung der Lagrange-Funktion,

$$z_i^* \in \operatorname{argmin}_{z_i \in \mathbb{D}_i} \phi_i(z_i) + z_i^T E_i^T \lambda^* \quad (15)$$

für alle $i \in \mathcal{N}_{\mathcal{I}}$ berechnet werden. Unter gewissen Voraussetzungen an die Funktionen ϕ_i , die Mengen \mathbb{D}_i und die Schrittweite $c > 0$ kann damit über die Iterationsvorschrift

$$z_i^\ell \in \operatorname{argmin}_{z_i \in \mathbb{D}_i} (\phi_i(z_i) + z_i^T E_i^T \lambda^\ell) \quad (16a)$$

$$\lambda^{\ell+1} = \lambda^\ell + c \left(\sum_{i=1}^{\mathcal{I}} E_i z_i^\ell - b \right) \quad (16b)$$

eine Optimallösung λ^* für das duale Problem (13) gefunden werden. Über (15) ergibt sich damit eine Optimallösung z^* für das ursprüngliche Problem (10).

Iterationsvorschrift (16) beschreibt ein duales Gradienten- beziehungsweise Aufstiegsverfahren. Bevor wir eine Interpretation des Namens liefern, geben wir ein allgemeines Konvergenzresultat an, dass sich auf [4, Prop. 1.2.3] zurückführen lässt.

Theorem 3.3 (Konvergenz: Duale Dekomposition). *Sei ϕ stark konvex mit Parameter $\alpha > 0$.² Sei (10) zulässig und sei \mathbb{D}_i konvex und abgeschlossen für alle $i \in \mathbb{N}_{\mathcal{I}}$. Sei $c \in (0, 2\alpha/\|E\|_2^2)$ eine konstante Schrittweite und seien $(z^\ell)_{\ell \in \mathbb{N}}$ und $(\lambda^\ell)_{\ell \in \mathbb{N}}$ für beliebige Anfangswerte $\lambda^0 \in \mathbb{R}^m$, $z^0 \in \mathbb{R}^n$ definiert über die Iterationsvorschrift (16). Dann gilt $z_i^\ell \rightarrow z_i^*$ und $\lambda^\ell \rightarrow \lambda^*$ für alle $i \in \mathbb{N}_{\mathcal{I}}$ und $\ell \rightarrow \infty$.*

Alternativ kann die konstante Schrittweite durch eine andere Schrittweite, zum Beispiel eine verschwindende Schrittweite (diminishing stepsize) mit der Bedingung

$$c^\ell \geq 0, \quad c^\ell \xrightarrow{\ell \rightarrow \infty} 0 \quad \text{und} \quad \sum_{\ell=0}^{\infty} c^\ell = \infty, \quad (17)$$

ersetzt werden [4, Prop. 1.2.4]. Starke Konvexität und die Annahmen an \mathbb{D}_i garantieren Eindeutigkeit der Optimallösungen z_i^ℓ in (16a) für alle $i \in \mathbb{N}_{\mathcal{I}}$. Mit diesen Bedingungen lässt sich zeigen, dass die duale Funktion ψ stetig differenzierbar ist und für den Gradienten gilt [4, Prop. 6.1.1]

$$\nabla \psi(\lambda) = \sum_{i=1}^{\mathcal{I}} E_i \left(\underset{z_i \in \mathbb{D}_i}{\operatorname{argmin}} (\phi_i(z_i) + z_i^T E_i^T \lambda) \right) - b.$$

Auch wenn eine explizite Form von ψ und $\nabla \psi$ nicht hergeleitet werden kann, erhält man damit $\psi(\lambda^\ell) = \sum_{i=1}^{\mathcal{I}} E_i z_i^\ell - b$ und $\psi(\lambda^{\ell+1}) = \psi(\lambda^\ell + c \nabla \psi(\lambda^\ell)) \geq \psi(\lambda^\ell)$ für eine geeignete Schrittweite $c > 0$, woraus sich der Name duales Gradientenverfahren erklärt.

Kommunikationsstruktur:

Die duale Dekomposition (16) basiert auf dem abwechselnden Lösen lokaler Optimierungsprobleme (16a) und dem Aktualisieren der Lagrange-Multiplikatoren (16b). Zum Ausführen der Iterationsvorschrift muss $E_i z_i^\ell$, $i \in \mathbb{N}_{\mathcal{I}}$, in jeder Iteration zwischen den Systemen kommuniziert werden. Eine Kommunikation zwischen System i und System j findet nur statt wenn $\dim(y_{i,j}) > 0$ oder $\dim(y_{j,i}) > 0$ gilt. Maximal müssen m (die Anzahl der Spalten der Matrix E) Variablen kommuniziert werden. In Beispiel 2.3 müssen 4 Variablen zwischen den Systemen ausgetauscht werden um λ lokal aktualisieren zu können. Falls einzelne Systeme nur mit wenigen anderen Systemen gekoppelt sind reduziert sich die Kommunikation, da die Matrizen E_i einige Nullzeilen enthalten. Alternativ zur lokalen Berechnung, kann λ bei hierarchischen Systemen global berechnet und kommuniziert werden.

² Eine Funktion $\phi : \mathbb{R}^n \rightarrow \mathbb{R}$ ist stark konvex mit Parameter $\alpha > 0$ falls $\phi(\kappa x + (1-\kappa)y) \leq \kappa f(x) + (1-\kappa)f(y) - \frac{\alpha}{2} \kappa(1-\kappa) \|x-y\|_2^2$ für alle $x, y \in \mathbb{R}^n$, für alle $\kappa \in [0, 1]$.

Die Systemdynamik (1a), die Nebenbedingungen \mathbb{X}_i , \mathbb{U}_i , sowie die Zielfunktionen ϕ_i , für $i \in \mathbb{N}_{\mathcal{I}}$, spielen bei der Kommunikation keine Rolle und müssen nur dem lokalen Regler i bekannt sein. Die Anwendung ist dadurch sehr flexibel gegenüber lokalen Änderungen. Der Rechenaufwand hängt von der Komplexität der lokalen Optimierungsprobleme (16a) ab. Der Aufwand der Kommunikation ändert sich mit der Vernetzung der Systeme untereinander.

Konvergenz & Zulässigkeit:

Falls geeignete Voraussetzungen erfüllt sind (zum Beispiel die Voraussetzungen von Theorem 3.3), dann konvergiert die duale Dekomposition linear gegen eine Optimallösung. Die Iterationen z^ℓ sind dabei in der Regel nicht zulässig, das heißt $\sum_{i=1}^{\mathcal{I}} E_i z_i^\ell - b \neq 0$ für alle $\ell \in \mathbb{N}$. Die lokalen Nebenbedingungen $z^\ell \in \mathbb{D}$ sind dagegen in jedem Schritt erfüllt. Falls die Systeme nur über die Zielfunktion gekoppelt sind (siehe Beispiel 2.2), ist z^ℓ zulässig für alle $\ell \in \mathbb{N}$ und $(z^\ell)_{\ell \in \mathbb{N}}$ konvergiert monoton gegen eine Optimallösung z^* , bei geeigneter Wahl der Schrittweite.

Erweiterungen:

Die Voraussetzung der starken Konvexität an die Zielfunktion ϕ ist sehr stark. Um die Voraussetzungen zu erfüllen kann eine konvexe Funktion ϕ durch die Funktion $\phi(z) + \frac{\varepsilon}{2} \|z\|_2^2$ für $\varepsilon > 0$ approximiert werden. Für $\varepsilon \searrow 0$ ergibt sich eine gute Approximation des ursprünglichen Optimierungsproblems. Darüber hinaus ist $z \mapsto \frac{\varepsilon}{2} \|z\|_2^2$ stark konvex mit Parameter $\alpha = \varepsilon$. Alternativ kann eine Reihe von Problemen

$$z^{k+1} = \underset{z \in \mathbb{D} \cap \mathbb{G}}{\operatorname{argmin}} \phi(z) + \frac{\rho^k}{2} \sum_{i=1}^{\mathcal{I}} \|z_i - z_i^k\|_2^2 \quad (18)$$

für Parameter $\rho^k \in \mathbb{R}_{>0}$ parallel über die duale Dekomposition gelöst werden um eine Optimallösung für das ursprüngliche Problem (10) zu finden. Eine geeignete Wahl der Parameter ρ^k garantiert, dass die Zielfunktion von (18) durch den zusätzlichen quadratischen Strafterm stark konvex ist. Die Idee findet sich in der Literatur unter dem Namen Proximal Algorithms oder Penalty Methods wobei wir auf [33] und [4] für Einzelheiten und Konvergenzeigenschaften verweisen.

Literatur & Anwendungsbeispiele:

Für eine genauere Einführung und weitere Resultate der dualen Dekomposition verweisen wir auf die Monographien [4, 5] sowie auf die Arbeit [31]. Erweiterungen der dualen Dekomposition mit verbesserten Konvergenzeigenschaften finden sich zum Beispiel in [30, 15]. Hierbei werden auch die Konvergenzeigenschaften bezüglich der Ver-

netzung der Systeme im Graphen sowie bei asynchronen Updates der Lagrange-Multiplikatoren untersucht. Verteilte allgemeine MPC-Algorithmen basierend auf der dualen Dekomposition finden sich in [21, 22, 29, 12]. Aktuelle Anwendungsbeispiele im Bereich der dynamischen Systeme finden sich in unterschiedlichen Smart Grid Anwendungen [27, 2, 10, 17]. In [27, 2] wird mittels dualer Dekomposition die Energieerzeugung und der Energieverbrauch optimiert. In [2] wird dabei eine Form der dualen Dekomposition bezüglich (18) verwendet. In [10] wird die duale Dekomposition verwendet um zeitvariante Strompreise zur Steuerung des Energieverbrauchs zu generieren und in [17] wird das Laden von Elektrofahrzeugen koordiniert. Weitere Anwendungen finden sich zum Beispiel bei der Regelung des Wasserstands in Behältern [29, 22].

3.2 Alternating Direction Method of Multipliers

Um die Konvergenzeigenschaften der dualen Dekomposition zu verbessern wurden Verfahren basierend auf der sogenannten Augmented-Lagrange-Funktion $\mathcal{L}_\rho : \mathbb{R}^{n+m} \rightarrow \mathbb{R}$,

$$\mathcal{L}_\rho(z, \lambda) = \phi(z) + z^T E^T \lambda + \frac{\rho}{2} \|Ez - b\|_2^2 \quad (19)$$

für $\rho \in \mathbb{R}_{>0}$ entwickelt, wobei hier ähnlich zu (18) ein zusätzlicher Term eingeführt wird. Da für eine Optimallösung z^* die Bedingung $Ez^* - b = 0$ erfüllt ist, stimmen die Optimallösungen von

$$\min_{z \in \mathbb{D} \cap \mathbb{G}} \phi(z) + \frac{\rho}{2} \|Ez - b\|_2^2 \quad (20)$$

und (10) überein. Durch den zusätzlichen quadratischen Strafterm lassen sich die Konvergenzeigenschaften der dualen Dekomposition verbessern, indem (16a) durch

$$z^\ell \in \operatorname{argmin}_{z \in \mathbb{D}} \left(\phi(z) + z^T E^T \lambda^\ell + \frac{\rho}{2} \|Ez - b\|_2^2 \right) \quad (21)$$

ersetzt wird (siehe zum Beispiel [4, Sec. 4.2]). Allerdings ist eine Dekomposition in separate Optimierungsprobleme und damit eine parallele Implementierung aufgrund der Kopplung von z_i in $\|\sum_{i=1}^{\mathcal{I}} E_i z_i - b\|_2^2$ nicht möglich.

Basierend auf der Augmented-Lagrange-Funktion kann allerdings die Alternating Direction Method of Multipliers (ADMM) beschrieben werden, die sich für eine verteilte Implementierung eignet. Für ADMM wird das Optimierungsproblem (10) zunächst in die Form

$$\min_{z_i \in \mathbb{D}_i; s_i \in \mathbb{R}^m} \sum_{i=1}^{\mathcal{I}} \phi_i(z_i) \quad (22a)$$

$$\text{s.t.} \quad \sum_{i=1}^{\mathcal{I}} s_i - b = 0 \quad (22b)$$

$$s_i - E_i z_i = 0 \quad \forall i \in \mathbb{N}_{\mathcal{I}} \quad (22c)$$

gebracht, indem neue Variablen $s_i \in \mathbb{R}^m$, $i \in \mathbb{N}_{\mathcal{I}}$ eingeführt werden. (Es reicht aus Variablen s_i für Zeilen von E_i einzuführen, die Einträge ungleich 0 besitzen. Zur Vereinfachung der Notation verwenden wir dennoch $s_i \in \mathbb{R}^m$.) Für die Nebenbedingungen der neuen Variablen definieren wir die Menge

$$\mathbb{S} = \{s \in \mathbb{R}^{\mathcal{I}m} \mid \sum_{i=1}^{\mathcal{I}} s_i - b = 0\}$$

und stellen die zugehörige Augmented-Lagrange-Funktion $\mathcal{L}_\rho : \mathbb{R}^n \times \mathbb{R}^{\mathcal{I}m} \times \mathbb{R}^{\mathcal{I}m} \rightarrow \mathbb{R}$,

$$\mathcal{L}_\rho(z, s, \lambda) = \sum_{i=1}^{\mathcal{I}} \phi_i(z_i) + (s_i^T - z_i^T E_i^T) \lambda_i + \frac{\rho}{2} \|s_i - E_i z_i\|_2^2 \quad (23)$$

auf. Beachte das im Gegensatz zu den Lagrange-Funktionen (11) und (19) für die Dimension der Lagrange-Multiplikatoren $\lambda^T = (\lambda_1^T, \dots, \lambda^T) \in \mathbb{R}^{\mathcal{I}m}$ gilt. Im Gegensatz zu (19) lässt sich die Funktion (23) für feste s und feste λ parallel über z_i optimieren. Damit ergibt sich die Iterationsvorschrift für ADMM aus der dualen Dekomposition über einen zusätzlichen Zwischenschritt:

$$z_i^{\ell+1} \in \operatorname{argmin}_{z_i \in \mathbb{D}_i} \phi_i(z_i) - z_i^T E_i^T \lambda_i^\ell + \frac{\rho}{2} \|s_i^\ell - E_i z_i\|_2^2 \quad (24a)$$

$$s_i^{\ell+1} \in \operatorname{argmin}_{s_i \in \mathbb{S}} \sum_{i=1}^{\mathcal{I}} s_i^T \lambda_i^\ell + \frac{\rho}{2} \|s_i - E_i z_i^{\ell+1}\|_2^2 \quad (24b)$$

$$\lambda_i^{\ell+1} = \lambda_i^\ell + \rho \left(\sum_{i=1}^{\mathcal{I}} s_i^\ell - E_i z_i^\ell \right). \quad (24c)$$

Hierbei wird im Gegensatz zur dualen Dekomposition alternierend über z und s optimiert bevor ein Update der Lagrange-Multiplikatoren durchgeführt wird. Die Schritte (24a) und (24c) können hierbei parallel für alle z_i , λ_i durchgeführt werden. Durch das Optimierungsproblem (24b) sieht die Iterationsvorschrift (24) im Vergleich zum ursprünglichen Optimierungsproblem (10) zunächst nicht nach einer Vereinfachung aus. Für manche Anwendungen lässt sich (24b) jedoch effizient lösen. Zum Beispiel besitzt das Optimierungsproblem nur lineare Gleichungsnebenbedingungen und keine Ungleichungsnebenbedingungen (versteckt in \mathbb{D}_i), was das Lösen im Allgemeinen vereinfacht. Falls keine Kopplung in den Nebenbedingungen vorliegen, sondern die Systeme nur über die Zielfunktion miteinander verbunden sind, ergibt sich ein unbeschränktes Optimierungsproblem für das sich die Optimallösung explizit berechnen lässt. In Spezialfällen lässt sich dabei auch die Dimension von s und λ reduzieren. Im Beispiel der Elektrizitätsnetze, Beispiel 2.4, wenn die Systeme über (9) miteinander verbunden sind, kann die Dimension der Variablen s sowie die Dimension der Lagrange-Multiplikatoren λ , unabhängig von \mathcal{I} gewählt werden (siehe zum Beispiel [7, 8]). Da die Kopplung (1b)

keine Dynamik besitzt, lässt sich das Optimierungsproblem (24b) darüber hinaus über den Prädiktionshorizont parallelisieren. Je nach Kopplung der Einzelsysteme (das heißt je nach Definition der Matrizen E_i , $i \in \mathbb{N}_{\mathcal{I}}$) sind weitere Parallelisierungen beim Lösen von (24b) möglich.

Der Vorteil von ADMM gegenüber der dualen Dekomposition ergibt sich aus der Konvergenz des Verfahrens. Im Bezug auf das Konvergenzresultat 3.3 lassen sich die Voraussetzungen an die Zielfunktion ϕ für ADMM von stark konvex auf konvex abschwächen. Zwar garantieren konvexe Zielfunktionen keine Eindeutigkeit der Optimallösungen, dafür lässt sich zeigen, dass alle Häufungspunkte von $(z^\ell)_{\ell \in \mathbb{N}}$ Optimallösungen von (10) sind. Konvergenz gilt hierbei unabhängig von der Wahl von $\rho > 0$. Die Konvergenzgeschwindigkeit hängt allerdings von der Wahl der Schrittweite beziehungsweise der Gewichtung des Strafterms ab. Wir verweisen auf [7] für ein allgemeines Konvergenzresultat mit Beweis.

Kommunikationsstruktur & Zulässigkeit

Bezüglich der Kommunikationsstruktur und der Zulässigkeit lassen sich die gleichen Aussagen treffen wie bei der dualen Dekomposition. Das gekoppelte Problem (24b) kann entweder von allen Systemen basierend auf $E_i z_i^{\ell+1}$ gelöst werden oder in einer hierarchischen Implementierung von einer zentralen Einheit. Bei einer hierarchischen Implementierung müssen zusätzlich zu den Lagrange-Multiplikatoren die Variablen s_i kommuniziert werden.

Literatur, Erweiterungen & Anwendungsbeispiele

Für eine detaillierte Einführung und weiterführende Literatur verweisen wir auf die Arbeiten [7, 32, 33]. Hierbei möchten wir vor allem auf die momentane Forschung zu ADMM für nichtkonvexe Optimierungsprobleme aufmerksam machen, sowie auf die Verbindung zu anderen Algorithmen wie Douglas-Ratchford-Splitting und auf die Einordnung in die Klasse der Proximal Point Algorithmen.

In [26] wird eine Augmented-Lagrangian based Alternating Direction Inexact Newton Method (ALADIN) entwickelt mit verbesserten Konvergenzeigenschaften in Bezug auf die Konvergenzgeschwindigkeit sowie der Anwendung auf nichtkonvexe Probleme unter der Annahme, dass sowohl die Zielfunktion ϕ_i als auch die Funktionen zur Beschreibung der Mengen \mathbb{D}_i , $i \in \mathbb{N}$, zweimal stetig differenzierbar sind. Die Ableitungen werden im gekoppelten Optimierungsschritt, dem Äquivalent zu Problem (24b), verwendet und müssen daher zwischen den Einzelsystemen kommuniziert werden und bekannt sein.

MPC-Algorithmen basierend auf ADMM finden sich mittlerweile in zahlreichen Arbeiten [37, 12, 28, 6, 1, 8, 25].

In [12] werden neben der dualen Dekomposition auch die Konvergenzeigenschaften von ADMM für verschiedene Graphen in der verteilten Optimierung untersucht. In [28] liegt der Fokus auf der Kommunikation und der effizienten Lösung eines Optimierungsproblems und nicht auf der Vernetzung von Einzelsystemen. In [6] werden, im Gegensatz zu den anderen Arbeiten, zeitkontinuierliche Systeme betrachtet. Zusätzliche Anwendungsbeispiele finden sich in [1, 8, 25] zur Wasserstandregelung, zur Optimierung des Stromverbrauchs in Smart Grids und zur Gebäudeautomatisierung.

3.3 Primale Algorithmen

Die duale Dekomposition sowie die Erweiterungen in ADMM basieren auf dem dualen Problem und der Berechnung der optimalen dualen Variablen λ^* . Zum Abschluss dieses Abschnittes wollen wir uns noch kurz mit Ideen der verteilten Optimierung beschäftigen, die nur auf den primalen Variablen basieren. Dafür gehen wir davon aus, dass die Systeme nur über die Zielfunktion miteinander gekoppelt sind. Dies kann wie in Beispiel 2.4 über die Funktion (8) modelliert werden, oder indem wir die linearen Nebenbedingungen (7b) als weiche Beschränkungen interpretieren und in die Zielfunktion aufnehmen

$$\sum_{i=1}^{\mathcal{I}} E_i z_i - b \mapsto \bar{\phi} \left(\sum_{i=1}^{\mathcal{I}} E_i z_i - b \right).$$

Zum Lösen des resultierenden Optimierungsproblems

$$\operatorname{argmin}_{z_i \in \mathbb{D}_i} \bar{\phi} \left(\sum_{i=1}^{\mathcal{I}} E_i z_i - b \right) + \sum_{i=1}^{\mathcal{I}} \phi_i(z_i) \quad (25)$$

definieren wir dafür die Iterationsvorschrift

$$z_i^{\ell+\frac{1}{2}} \in \operatorname{argmin}_{z_i \in \mathbb{D}_i} \phi_i(z_i) + \bar{\phi} \left(E_i z_i + \sum_{\substack{j=1 \\ j \neq i}}^{\mathcal{I}} E_j z_j^\ell - b \right) \quad (26a)$$

$$z_i^{\ell+1} = c_i^\ell z_i^{\ell+\frac{1}{2}} + (1 - c_i^\ell) z_i^\ell. \quad (26b)$$

Die Optimierung (26a) sowie das Update (26b) können parallel von den einzelnen Systemen durchgeführt werden. Für konvexe und kompakte nichtleere Mengen \mathbb{D}_i , $i \in \mathbb{N}_{\mathcal{I}}$, und konvexe Zielfunktionen ϕ und $\bar{\phi}$ kann unter geeigneter Wahl der Schrittweiten $(c_i^\ell)_{\ell \in \mathbb{N}} \subset \mathbb{R}$, $i \in \mathbb{N}_{\mathcal{I}}$, monotone Konvergenz gegen den optimalen Zielfunktionswert nachgewiesen werden. Im Verbindung mit verteilter Modellprädiktiver Regelung finden sich diese Ansätze zum Beispiel in [35, 36, 9]. Konvexität der Mengen \mathbb{D}_i , $i \in \mathbb{N}$, ist notwendig um Zulässigkeit von $(z^\ell)_{\ell \in \mathbb{N}}$ in jedem Schritt für einen zulässigen Anfangswert $z^0 \in \mathbb{D}$ zu garantieren.

Unter den genannten Voraussetzungen sind Häufungspunkte von $(z^\ell)_{\ell \in \mathbb{N}}$ globale Minima des Optimierungsproblems (25). Dabei garantiert zum Beispiel die konstante

Schrittweite $c_i^\ell = \frac{1}{\mathcal{I}}$ für alle $i \in \mathbb{N}_{\mathcal{I}}$, $\ell \in \mathbb{N}$, monotone Konvergenz. Alternativ kann zum Beispiel zentral

$$(c_1^\ell, \dots, c_{\mathcal{I}}^\ell) \in \underset{c_i \in [0,1], i \in \mathbb{N}_{\mathcal{I}}}{\operatorname{argmin}} \sum_{i=1}^{\mathcal{I}} \phi_i(c_i^\ell z_i^{\ell+\frac{1}{2}} + (1-c_i^\ell)z_i^\ell) + \bar{\phi} \left(\sum_{i=1}^{\mathcal{I}} \left(E_i c_i^\ell z_i^{\ell+\frac{1}{2}} + (1-c_i^\ell)z_i^\ell \right) - b \right),$$

oder ein einzelnes $c \in [0, 1]$ (d.h. $c_1 = \dots = c_{\mathcal{I}}$) optimiert werden. Falls die Zielfunktion nicht konvex ist kann das Verfahren mit variabler Schrittweite genutzt werden um eine monoton fallende Folge bezüglich des Zielfunktionswertes zu garantieren, auch wenn Konvergenz zu einem lokalen oder globalen Minimum nicht ohne weitere Voraussetzungen garantiert werden kann.

Kommunikationsstruktur und ergänzende Literatur

Ähnlich wie in den anderen vorgestellten Algorithmen muss hier die Lösung $E_i z_i^{\ell+\frac{1}{2}}$ zwischen den Systemen kommuniziert werden um ein lokales Optimierungsproblem in jeder Iteration lösen zu können. Hier wird mit dieser Information allerdings nur die primale Lösung aktualisiert. Bei der Optimierung muss die Funktion $\bar{\phi}$ allen Systemen bekannt sein. Die lokalen Informationen ϕ_i und D_i müssen nicht kommuniziert werden oder den anderen Systemen bekannt sein. Die Iterationsvorschrift (26) kann auch angewendet werden wenn die Variablen z_i , $i \in \mathbb{N}_{\mathcal{I}}$, in $\bar{\phi}$ nichtlinear gekoppelt sind.

Anwendungen und die Analyse verwandter Algorithmen im MPC-Kontext finden sich zum Beispiel in [36], [34, Kap. 6] und [9]. In [11, 26] finden sich weiterführende verteilte Optimierungsalgorithmen die sich sowohl die primale als auch die duale Optimierung zunutze machen.

4 Optimierung im geschlossenen Regelkreis

Im geschlossenen Regelkreis muss zur iterativen Berechnung der Zustandsrückführung zu jedem Zeitschritt k ein OCP gelöst werden. Dabei basieren die vorgestellten Optimierungsansätze auf einer iterativen Berechnung von z^* . Hierfür bieten sich insbesondere Konzepte des Warm-Starts an. Zum Vorstellen des Warm-Starts nutzen wir in Übereinstimmung mit Definition (5) die Notation

$$z^{*k} = \{[x^{*k}(t), u^{*k}(t), y^{*k}(t)] | t \in \mathbb{N}_{k, k+N-1}\}$$

für eine Optimallösung von (10) zum Zeitpunkt $k \in \mathbb{N}$. Optimierungsprobleme zum Zeitpunkt k und $k+1$ unterscheiden sich nur durch die Anfangswerte $x(k)$ und $x(k+1)$

sowie die externen Daten $v(k)$ und $v(k+N)$. Dadurch lässt sich erwarten, dass im Allgemeinen für ausreichend großen Prädiktionshorizont $K \in \mathbb{N}$ die Eigenschaft

$$\begin{aligned} & \{[x^{*k}(t), u^{*k}(t), y^{*k}(t)] | t \in \mathbb{N}_{k+1, k+N-1}\} \\ & \approx \{[x^{*k+1}(t), u^{*k+1}(t), y^{*k+1}(t)] | t \in \mathbb{N}_{k+1, k+N-1}\} \end{aligned} \quad (27)$$

erfüllt ist. Im Zusammenhang mit ökonomischen MPC Anwendungen lässt sich diese Näherung zum Beispiel über die Turnpike-Eigenschaft analysieren [23]. Durch diese Eigenschaft bietet sich die Initialisierung

$$\begin{aligned} & \{[x^0(t), u^0(t), y^0(t)] | t \in \mathbb{N}_{k+1, k+N-1}\} \\ & := \{[x^{*k}(t), u^{*k}(t), y^{*k}(t)] | t \in \mathbb{N}_{k+1, k+N-1}\} \end{aligned}$$

und $[x^0(k+N), u^0(k+N), y^0(k+N)] = 0$ an. Auf die gleiche Weise lassen sich die Lagrange-Multiplikatoren λ , und die Variablen s der ADMM-Iterationsvorschrift (24b) definieren. Durch geeignete Initialisierung lässt sich damit die Anzahl der Iterationen zum Erreichen eines Abbruchkriteriums signifikant verringern.

Zusätzlich kann man die Anzahl der Iterationen $\ell \in \mathbb{N}$ pro Zeitschritt $k \in \mathbb{N}$ limitieren um den Rechenaufwand und die notwendige Kommunikation zu verringern. Dies bietet sich zum Beispiel an, wenn die Systeme nicht durch Gleichungsnebenbedingungen der Form (7b) miteinander gekoppelt sind, sondern nur über die Zielfunktion wie in Beispiel 2.4. Da keine Kopplungsnebenbedingungen existieren, ist sichergestellt, dass z^ℓ zulässig ist für alle $\ell \in \mathbb{N}$. Durch frühzeitiges Abbrechen der verteilten Optimierungsalgorithmen büßt man also lediglich Optimalität ein. In Kombination mit Warm-Start und dadurch dass die Zustandsrückführung nur von $u^{*k}(k)$ Gebrauch macht, lassen sich auch für wenige Iterationen ℓ gute Ergebnisse erwarten. Das Konzept der unvollständigen Optimierung findet sich in Suboptimal-MPC [24, Kap. 10.9] und in Real-Time-Iterationsverfahren für MPC [34, Kap. 8.9],[14]. Für verteiltes MPC wird es zum Beispiel in [36, 35, 9] im Allgemeinen, sowie für chemische Prozesse und Smart-Grid Anwendungen untersucht.

Literatur

- [1] M. Annergren, A. Hansson, and B. Wahlberg. An ADMM algorithm for solving ℓ_1 regularized MPC. In *Proc. 51st IEEE Conference on Decision and Control*, pages 4486–4491, 2012.
- [2] I. Atzeni, L. G. Ordóñez, G. Scutari, D. P. Palomar, and J. R. Fonollosa. Demand-side management via distributed energy generation and storage optimizati-

- on. *IEEE Transactions on Smart Grid*, 4(2):866–876, 2013.
- [3] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.
- [4] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.
- [5] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, Belmont, MA, USA, 1989.
- [6] A. Bestler and K. Graichen. Distributed model predictive control for continuous-time nonlinear systems based on suboptimal ADMM. arXiv preprint, arXiv:1706.09599, 2017.
- [7] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.
- [8] P. Braun, T. Faulwasser, L. Grüne, C. M. Kellett, S. R. Weller, and K. Worthmann. Hierarchical distributed ADMM for predictive control with applications in power networks. *IFAC Journal of Systems and Control*, 3:10 – 22, 2018.
- [9] P. Braun, L. Grüne, C. M. Kellett, S. R. Weller, and K. Worthmann. A distributed optimization algorithm for the predictive control of smart grids. *IEEE Transactions on Automatic Control*, 61(12):3898–3911, 2016.
- [10] P. Braun, L. Grüne, C. M. Kellett, S. R. Weller, and K. Worthmann. Towards price-based predictive control of a small-scale electricity network. *International Journal of Control*, 2017. <https://doi.org/10.1080/00207179.2017.1339329>.
- [11] T.-H. Chang, A. Nedić, and A. Scaglione. Distributed constrained optimization by consensus-based primal-dual perturbation method. *IEEE Transactions on Automatic Control*, 59(6):1524–1538, 2014.
- [12] C. Conte, T. Summers, M. N. Zeilinger, M. Morari, and C. N. Jones. Computational aspects of distributed optimization in model predictive control. In *Proc. 51st IEEE Conference on Decision and Control*, pages 6819–6824, 2012.
- [13] G. B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8(1):101–111, 1960.
- [14] M. Diehl, H. G. Bock, and J. P. Schlöder. A real-time iteration scheme for nonlinear optimization in optimal feedback control. *SIAM Journal on Control and Optimization*, 43(5):1714–1736, 2005.
- [15] J. C. Duchi, A. Agarwal, and M. J. Wainwright. Dual averaging for distributed optimization: Convergence analysis and network scaling. *IEEE Transactions on Automatic Control*, 57(3):592–606, 2012.
- [16] H. Everett. Generalized Lagrange multiplier method for solving problems of optimum allocation of resources. *Operations Research*, 11(3):399–417, 1963.
- [17] C. Le Floch, F. Belletti, S. Saxena, A. M. Bayen, and S. Moura. Distributed optimal charging of electric vehicles for demand response and load shaping. In *Proc. 54th IEEE Conference on Decision and Control*, pages 6570–6576, 2015.
- [18] M. Fortin and R. Glowinski. On decomposition-coordination methods using an augmented Lagrangian. In *Augmented Lagrangian Methods: Applications to the numerical solution of Boundary-Value Problems*. North-Holland: Amsterdam, 1983.
- [19] D. Gabay. Applications of the method of multipliers to variational inequalities. In *Augmented Lagrangian Methods: Applications to the Solution of Boundary-Value Problems*. North-Holland: Amsterdam, 1983.
- [20] D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976.
- [21] P. Giselsson, M. D. Doan, T. Keviczky, B. De Schutter, and A. Rantzer. Accelerated gradient methods and dual decomposition in distributed model predictive control. *Automatica*, 49(3):829–833, 2013.
- [22] P. Giselsson and A. Rantzer. Distributed model predictive control with suboptimality and stability guarantees. In *Proc. 49th IEEE Conference on Decision and Control*, pages 7272–7277, 2010.
- [23] L. Grüne. Approximation properties of receding horizon optimal control. *Jahresbericht der Deutschen Mathematiker-Vereinigung*, 118(1):3–37, 2016.
- [24] L. Grüne and J. Pannek. *Nonlinear Model Predictive Control. Theory and Algorithms*. Springer, 2 edition, 2017.
- [25] X. Hou, Y. Xiao, J. Cai, J. Hu, and J. E. Braun. Distributed model predictive control via proximal jacobian admm for building control applications. In *Proc. IEEE American Control Conference*, pages 37–43, 2017.
- [26] B. Houska, J. Frasch, and M. Diehl. An augmented Lagrangian based algorithm for distributed nonconvex optimization. *SIAM Journal on Optimization*, 26(2):1101–1127, 2016.
- [27] G. K. H. Larsen, N. D. van Foreest, and J. M. A. Scherpen. Distributed control of the power supply-demand balance. *IEEE Transactions on Smart Grid*, 4(2):828–836, 2013.

- [28] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Püschel. Distributed optimization with local domains: Applications in mpc and network flows. *IEEE Transactions on Automatic Control*, 60(7):2004–2009, 2015.
- [29] I. Necoara and D. Clipici. Efficient parallel coordinate descent algorithm for convex optimization problems with separable constraints: Application to distributed MPC. *Journal of Process Control*, 23(3):243 – 253, 2013.
- [30] I. Necoara and J. A. K. Suykens. Application of a smoothing technique to decomposition in convex optimization. *IEEE Transactions on Automatic Control*, 53(11):2674–2679, 2008.
- [31] A. Nedić and A. Ozdaglar. *Cooperative distributed multi-agent optimization*, page 340–386. Cambridge University Press, 2009.
- [32] B. O’Donoghue, G. Stathopoulos, and S. Boyd. A splitting method for optimal control. *IEEE Transactions on Control Systems Technology*, 21(6):2432–2442, 2013.
- [33] N. Parikh and S. P. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2013.
- [34] J. B. Rawlings, D. Q. Mayne, and M. M. Diehl. *Model Predictive Control: Theory, Computation, and Design*. Nob Hill Publishing, 2 edition, 2017.
- [35] B. T. Stewart, A. N. Venkat, J. B. Rawlings, S. J. Wright, and G. Pannocchia. Cooperative distributed model predictive control. *Systems & Control Letters*, 59(8):460 – 469, 2010.
- [36] A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright. Distributed mpc strategies with application to power system automatic generation control. *IEEE Transactions on Control Systems Technology*, 16(6):1192–1206, 2008.
- [37] Z. Wang and C. J. Ong. Distributed model predictive control of linear discrete-time systems with local and global constraints. *Automatica*, 81:184 – 195, 2017.