3. Übung

Aufgabe 3.1 Quadratische Optimierung

Als Beispiel für die quadratische Optimierung wird im Folgenden die optimale Steuerung eines linearen zeitdiskreten Systems betrachtet. Das Optimierungsproblem ist gegeben durch

$$\min_{\boldsymbol{u}} \quad \sum_{k=0}^{N-1} \left(\boldsymbol{x}_k^\mathrm{T} \boldsymbol{Q} \boldsymbol{x}_k + \boldsymbol{u}_k^\mathrm{T} \boldsymbol{R} \boldsymbol{u}_k \right) + \boldsymbol{x}_N^\mathrm{T} \boldsymbol{S} \boldsymbol{x}_N \tag{1a}$$

$$\text{s.t.} \quad \boldsymbol{x}_{k+1} = \boldsymbol{A}\boldsymbol{x}_k + \boldsymbol{B}\boldsymbol{u}_k \,, \quad k = 1, \ldots, N-1 \tag{1b}$$

mit dem Zustand $\boldsymbol{x} \in \mathbb{R}^{n_x}$, dem Eingang $\boldsymbol{u} \in \mathbb{R}^{n_u}$, dem Horizont N und dem Anfangszustand $\boldsymbol{x}_0$. Die Kostenfunktion gewichtet die Zustände und Stellgrößen mit den Matrizen $\boldsymbol{Q}$ und $\boldsymbol{R}$ sowie den Endzustand mit der Matrix $\boldsymbol{S}$. Das Ziel besteht darin, das System in den Ursprung $(\boldsymbol{x}, \boldsymbol{u}) = (\boldsymbol{0}, \boldsymbol{0})$ zu überführen.

Liegen keine weiteren Beschränkungen vor, so entspricht die Problemstellung gerade dem bekannten linear-quadratischen Reglerentwurf. In diesem Fall ergibt sich die optimale Steuerung gemäß dem Regelgesetz

$$\boldsymbol{u}_k^* = \boldsymbol{K}_k \boldsymbol{x}_k \,, \quad k = 0, \ldots, N-1 \tag{2}$$

mit der zeitvarianten Rückführmatrix

$$\boldsymbol{K}_k = -(\boldsymbol{R} + \boldsymbol{B}^\mathrm{T} \boldsymbol{P}_{k+1} \boldsymbol{B})^{-1} (\boldsymbol{B}^\mathrm{T} \boldsymbol{P}_{k+1} \boldsymbol{A}) \tag{3}$$

und der Matrix $\boldsymbol{P}_k$, welche aus der Riccati-Gleichung

$$\boldsymbol{P}_k = (\boldsymbol{Q} + \boldsymbol{A}^\mathrm{T} \boldsymbol{P}_{k+1} \boldsymbol{A}) - (\boldsymbol{B}^\mathrm{T} \boldsymbol{P}_{k+1} \boldsymbol{A})^\mathrm{T} (\boldsymbol{R} + \boldsymbol{B}^\mathrm{T} \boldsymbol{P}_{k+1} \boldsymbol{B})^{-1} (\boldsymbol{B}^\mathrm{T} \boldsymbol{P}_{k+1} \boldsymbol{A}) \tag{4}$$

bestimmt werden kann, indem ausgehend von $\boldsymbol{P}_N = \boldsymbol{S}$ rückwärts iteriert wird. Skizzieren Sie die Herleitung der obigen Gleichungen, indem Sie die optimalen Überführungskosten

$$J^*(\boldsymbol{x}_k) = \min_{\boldsymbol{u}_k} \left(\boldsymbol{x}_k^\mathrm{T} \boldsymbol{Q} \boldsymbol{x}_k + \boldsymbol{u}_k^\mathrm{T} \boldsymbol{R} \boldsymbol{u}_k + J^*(\boldsymbol{x}_{k+1}) \right) \tag{5}$$

für den Schritt $k = N-1$ betrachten und den Zusammenhang $J^*(\boldsymbol{x}_N) = \boldsymbol{x}_N^\mathrm{T} \boldsymbol{S} \boldsymbol{x}_N$ ausnutzen.

Implementieren Sie den zeitdiskreten Riccati-Regler in MATLAB, indem Sie die Funktion `riccati` ergänzen. Als Beispielsystem wird der zeitdiskrete Doppelintegrator

$$\boldsymbol{x}_{k+1} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \boldsymbol{x}_k + \begin{bmatrix} 0 \\ \Delta t \end{bmatrix} u_k \tag{6}$$

mit $n_x = 2$, $n_u = 1$ und der Abtastzeit $\Delta t = T_\mathrm{hor}/N$ betrachtet. Die Gewichtungsmatrizen werden auf $\boldsymbol{S} = \mathrm{diag}(1, 0)$, $\boldsymbol{Q} = \mathrm{diag}(1, 0)\,\Delta t$ und $\boldsymbol{R} = 0.01\,\Delta t$ gesetzt. Berechnen Sie die optimale Steuerung mit dem Riccati-Regler und stellen Sie die Lösung grafisch dar.

Aufgabe 3.2

Alternativ zum Riccati-Ansatz kann das Optimierungsproblem (1) auch als ein gewöhnliches quadratisches Optimierungsproblem mit Gleichungsbeschränkungen betrachtet werden. Das Optimierungsproblem

$$\min_{\bar{\boldsymbol{x}}} \quad \frac{1}{2}\bar{\boldsymbol{x}}^\mathsf{T}\bar{\boldsymbol{G}}\bar{\boldsymbol{x}}^\mathsf{T} \tag{7a}$$

$$\text{u.B.v.} \quad \bar{\boldsymbol{A}}\bar{\boldsymbol{x}} = \bar{\boldsymbol{b}} \tag{7b}$$

wird dann bezüglich den zusammengefassten Zuständen und Stellgrößen $\bar{\boldsymbol{x}} = \left[\boldsymbol{x}_1^\mathsf{T},\, \boldsymbol{u}_0^\mathsf{T},\, \ldots,\, \boldsymbol{x}_N^\mathsf{T},\, \boldsymbol{u}_{N-1}^\mathsf{T}\right]^\mathsf{T}$ formuliert. Da der Anfangszustand $\boldsymbol{x}_0$ fest ist, muss dieser nicht als freie Variable betrachtet werden und es ergeben sich insgesamt $(n_x + n_u)\,N$ Optimierungsvariablen. Welche Struktur haben die Matrix $\bar{\boldsymbol{G}}$ sowie die Beschränkungen $\bar{\boldsymbol{A}}\bar{\boldsymbol{x}} = \bar{\boldsymbol{b}}$? Lösen Sie das quadratische Optimierungsproblem mit MATLAB und vergleichen Sie die Ergebnisse mit dem Riccati-Regler.

Aufgabe 3.3

Die Betrachtung als allgemeines quadratisches Optimierungsproblem bietet den Vorteil, dass die Problemstellung sehr einfach um weitere Beschränkungen erweitert werden kann. Von großer praktischer Relevanz sind insbesondere Zustands- und Stellgrößenbeschränkungen der Form

$$\boldsymbol{x}_{k+1} \in [\boldsymbol{x}^-, \boldsymbol{x}^+], \quad \boldsymbol{u}_k \in [\boldsymbol{u}^-, \boldsymbol{u}^+], \quad k = 0, \ldots, N-1\,. \tag{8}$$

Implementieren Sie ein aktive Restriktionen-Verfahren, indem Sie die MATLAB-Funktion `active_set_qp` ergänzen. Berechnen Sie die optimale Steuerung des Doppelintegrators für unterschiedliche Begrenzungen der Zustände und Stellgrößen und vergleichen Sie die Ergebnisse mit dem unbeschränkten Fall. Falls nur Stellgrößenbeschränkungen $\boldsymbol{u}_k \in [\boldsymbol{u}^-, \boldsymbol{u}^+]$ betrachtet werden, so könnte man als naiven Ansatz beim Riccati-Regler eine Projektion auf die zulässige Menge $\boldsymbol{u}_k = \mathbf{min}(\mathbf{max}(\boldsymbol{u}^-,\, \boldsymbol{K}_k\boldsymbol{x}_k),\, \boldsymbol{u}^+)$ vornehmen. Führt dieser Ansatz auf die optimale Lösung?

Aufgabe 3.4 Zusatzaufgabe

Implementieren Sie ein Interior Point-Verfahren in der MATLAB-Funktion `interior_point_qp` für Probleme der Form

$$\min_{\bar{\boldsymbol{x}}} \quad \frac{1}{2}\bar{\boldsymbol{x}}^\mathsf{T}\bar{\boldsymbol{G}}\bar{\boldsymbol{x}}^\mathsf{T} + \bar{\boldsymbol{c}}^\mathsf{T}\bar{\boldsymbol{x}} \tag{9a}$$

$$\text{u.B.v.} \quad \bar{\boldsymbol{A}}\bar{\boldsymbol{x}} = \bar{\boldsymbol{b}} \tag{9b}$$

$$\bar{\boldsymbol{x}} \geq 0\,. \tag{9c}$$

Welche Änderungen ergeben sich im Vergleich zum Interior Point-Verfahren für lineare Optimierungsprobleme in der Standardform? Berechnen Sie die optimale Steuerung des Doppelintegrators mit Beschränkungen für einen festen Zeithorizont T_hor und wachsender Anzahl Schritte N und vergleichen Sie die Rechenzeit mit dem aktive Restriktionen-Verfahren. Welche Unterschiede ergeben sich und warum?

Aufgabe 3.5 Nichtlineare Optimierung

Ein Benchmarkproblem in der nichtlinearen Optimierung ist das „Largest small polygon"-Problem, bei dem die Fläche eines Polygons mit n Ecken und einem Maximalabstand zwischen allen Punkten maximiert werden soll. Das Problem lässt sich in Abhängigkeit der Polarkoordinaten $\boldsymbol{r} = [r_1, \ldots, r_n]^\mathsf{T}$ und $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_n]^\mathsf{T}$ in der folgenden Form ausdrücken:

$$\min_{(\boldsymbol{r},\boldsymbol{\theta})} \quad f(\boldsymbol{r},\boldsymbol{\theta}) = -\frac{1}{2}\sum_{i=1}^{n-1} r_{i+1} r_i \sin(\theta_{i+1} - \theta_i) \tag{10a}$$

$$\text{u.B.v.} \quad r_n = 0\,, \quad \theta_n = \pi\,, \tag{10b}$$

$$r_i^2 + r_j^2 - 2 r_i r_j \cos(\theta_j - \theta_i) \leq 1\,, \qquad i = 1, \ldots, n-2\,, \quad j = i+1, \ldots, n-1\,, \tag{10c}$$

$$\theta_i \leq \theta_{i+1}\,, \qquad i = 1, \ldots, n-2\,, \tag{10d}$$

$$0 \leq \theta_i \leq \pi\,, \qquad i = 1, \ldots, n-1\,, \tag{10e}$$

$$0 \leq r_i \leq 1\,, \qquad i = 1, \ldots, n-1\,. \tag{10f}$$

Das Problem ist sowohl in der Kostenfunktion als auch in den Beschränkungen nichtlinear und besitzt mehrere lokale Minima - vor allem für größere Werte von n. Veranschaulichen Sie den Aufbau des Polygons und die Bedeutung der Polarkoordinaten $\boldsymbol{r}$ und $\boldsymbol{\theta}$ in einer Skizze. Erläutern Sie die Bedeutung der Kostenfunktion (10a) sowie der Beschränkungen (10c)–(10f) und leiten Sie die einzelnen Beziehungen her.

Aufgabe 3.6

Das beschränkte Optimierungsproblem (10) soll im Folgenden unter MATLAB mit Hilfe der Funktion `fmincon` gelöst werden. Implementieren Sie dazu die Kostenfunktion (10a) als auch die Beschränkungen (10c)–(10f) in geeigneter Weise. Zunächst sollen die verhältnismäßig kleinen Werte $n = 6$ und $n = 10$ betrachtet werden. Lösen Sie das Problem numerisch mit Hilfe von `fmincon` für verschiedene Startpunkte $(\boldsymbol{r}^0, \boldsymbol{\theta}^0)$. Stellen Sie die berechneten Polygone grafisch dar und diskutieren Sie die Ergebnisse im Hinblick auf die berechnete Kostenfunktion (also die zu maximierende Fläche des Polygons) und die Existenz lokaler Minima.

Aufgabe 3.7 Zusatzaufgabe

Vor allem für größere Werte von n ist die Wahl des Startpunkts $(\boldsymbol{r}^0, \boldsymbol{\theta}^0)$ von elementarer Bedeutung zur Lösung dieses Problems. Berechnen Sie den Startpunkt $(\boldsymbol{r}^0, \boldsymbol{\theta}^0)$ aus einem gleichförmigen Polygon mit n Kanten und lösen Sie damit das Problem für die Werte $n = 6, 10, 20, 100$. Stellen Sie das Startpolygon als auch die berechnete Lösung grafisch dar. Aufgrund der zahlreichen lokalen Minima kann es des Weiteren notwendig sein, den berechneten Startpunkt $(\boldsymbol{r}^0, \boldsymbol{\theta}^0)$ zusätzlich zu stören, z.B. um 0.01 in jeder Variablen. Die folgende Tabelle zeigt Referenzwerte für die maximale Polygonfläche $|f(\boldsymbol{r}^*, \boldsymbol{\theta}^*)|$ für $n = 6, 10, 20, 100$, die mit verschiedenen Optimierern erzielt wurden:

Anzahl Ecken n	6	10	20	100
Maximale Fläche $\lvert f(\boldsymbol{r}^*, \boldsymbol{\theta}^*)\rvert$	0.67498	0.74913	0.77685	0.78506