Fachgebiet Prozessoptimierung

TECHNISCHE UNIVERSITÄT ILMENAU

Fakultät für Informatik und Automatisierung
Institut für Automatisierungs- und Systemtechnik

Praktikum Dynamische Prozessoptimierung

Versuch DynPO-2, Sommersemester 2020

Programmierung und numerische Lösung von Optimalsteuerungsproblemen mittels Standardsoftware

Verantwortlicher Hochschullehrer: Prof. Dr.–Ing. habil. P. Li
Versuchsverantwortliche: Dr.–Ing. S. Hopfgarten, M. Sc. B. Juris

Name, Vorname, Matr.-Nr.	**Matrikel-Nr.**
Mitarbeiter in der Praktikumsgruppe	
Datum, Note, Unterschrift	

1 Einführung

Das Praktikum soll die Umsetzung verbal gegebener Aufgabenstellungen der Optimalsteuerung in mathematische Problemformulierungen, wie sie in der entsprechenden Lehrveranstaltung [7] behandelt werden, schulen. Nach dem Aufstellen des Problems in der Vorbereitungsphase auf den Praktikumsversuch wird eine Implementierung während der Durchführung des Versuchs mit Standardsoftware (z. B. MATLAB®, MATLAB® Optimization Toolbox^{TM})[1] [12] bzw. Octave [13] vorgenommen. Im Anhang sind einige Routinen aus der genannten Toolbox aufgeführt, die für die Lösung der Probleme in Frage kommen könnten. Die Effekte, die bei der Lösung von Optimalsteuerungsproblemen auftreten können, sind zu diskutieren, die erhaltenen Ergebnisse geeignet darzustellen sowie mit Bezug auf den jeweils zu untersuchenden Prozess zu interpretieren und unter praktischen Aspekten zu bewerten.

2 Problemformulierung

Ein allgemeines, nichtlineares und beschränktes Optimalsteuerungsproblem kann wie folgt notiert werden:

Zu minimierendes Zielfunktional:

$$\min_{\boldsymbol{u}(t)} \left\{ J = \Phi\left(\boldsymbol{x}(t_f)\right) + \int_0^{t_f} f_0(\boldsymbol{x}(t), \boldsymbol{u}(t), t)\, dt \right\}$$

Prozessbeschreibung (Zustandsdifferentialgleichung):

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t), t), \quad t \in [0, t_f]\,,\ \boldsymbol{x}(t) \in \mathbb{R}^n,\ \boldsymbol{u}(t) \in \mathbb{R}^m$$

Randbedingungen:

Fester Anfangszustand: $\boldsymbol{x}(0) = \boldsymbol{x}_0$

Fester Endzustand $\boldsymbol{x}(t_f) = \boldsymbol{x}_{t_f}$ (fest)

oder freier Endzustand: $\boldsymbol{x}(t_f)$ (frei)

Weitere Beschränkungen (Koordinatenbeschränkungen für Zustands- und Steuergrößen, allgemeine Beschränkungen für Zustands- und/oder Steuergrößen):

$$\begin{aligned} \boldsymbol{x}_{min} &\leq \boldsymbol{x}(t) \leq \boldsymbol{x}_{max} \\ \boldsymbol{u}_{min} &\leq \boldsymbol{u}(t) \leq \boldsymbol{u}_{max} \\ \boldsymbol{h}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) &\leq \boldsymbol{0} \end{aligned} \tag{1}$$

Die Endzeit t_f kann fest oder frei sein. Probleme mit freier Endzeit sind geeignet umzuformen. Typischerweise werden die beschränkten Optimalsteuerungsprobleme mit direkten Methoden gelöst. Das bedeutet, dass eine Diskretisierung aller Bestandteile des Optimalsteuerungsproblems vorgenommen und eine Parametrisierung von Steuerungen und/oder Zuständen angewendet wird. d. h. also eine Transformation eines Optimalsteuerungsproblems in ein (sehr oft) nichtlineares (quasi-statisches) Optimierungsproblem.

Eine mögliche Variante der numerischen Lösung von (1) besteht darin, Steuer- und Zustandsgrößen zeitlich zu diskretisieren, beispielsweise durch Anwendung eines Integrationsverfahrens zur Lösung der Zustandsdifferentialgleichung und damit zur Berechnung der Zustandsverläufe. Mit $\boldsymbol{x}^k = \boldsymbol{x}(t^k),\ \boldsymbol{u}^k = \boldsymbol{u}(t^k),\ 0 = t^0 < t^1 < \ldots < t^K = t_f$ wird aus (1) das zeitdiskrete

[1] MATLAB® und MATLAB® Optimization Toolbox^{TM} sind ein eingetragenes Warenzeichen bzw. eine eingetragene Handelsmarke der The Mathworks Inc.

Optimalsteuerungsproblem

$$
\begin{aligned}
&\boldsymbol{x}^{k+1} \;=\; \boldsymbol{f}^k(\boldsymbol{x}^k, \boldsymbol{u}^k), \quad k = 0, 1, \ldots, K-1 \\
&\min_{u^k} \left\{ J \;=\; \Phi(\boldsymbol{x}^K) + \sum_{k=0}^{K-1} \boldsymbol{f}_0^k(\boldsymbol{x}^k, \boldsymbol{u}^k) \right\} \\
&\boldsymbol{h}^k(\boldsymbol{x}^k, \boldsymbol{u}^k) \;\leq\; \boldsymbol{0}, \quad k = 0, 1, \ldots, K-1 \\
&\boldsymbol{h}^K(\boldsymbol{x}^K) \;\leq\; \boldsymbol{0}
\end{aligned}
\tag{2}
$$

Die zeitliche Diskretisierung könnte in einem einfachen Fall mit dem Euler-Verfahren und konstanter Schrittweite $\Delta t = t^{k+1} - t^k = \frac{t_f}{K}$ erfolgen. Wird $\boldsymbol{u}(t) = \boldsymbol{u}^k$ =const. für $t^k \leq t < t^{k+1}$ gewählt, so gilt

$$
\begin{aligned}
\boldsymbol{f}^k(\boldsymbol{x}^k, \boldsymbol{u}^k) \quad &= \quad \boldsymbol{x}^k + \Delta t \cdot \bar{\boldsymbol{f}}(\boldsymbol{x}^k, \boldsymbol{u}^k, t^k) \\
\Phi(\boldsymbol{x}^K) \quad &= \quad \bar{\Phi}\left(\boldsymbol{x}(t^K)\right) \\
\boldsymbol{f}_0^k(\boldsymbol{x}^k, \boldsymbol{u}^k) \quad &= \quad \Delta t \cdot \bar{\boldsymbol{f}}_0(\boldsymbol{x}^k, \boldsymbol{u}^k, t^k)
\end{aligned}
$$

Eine weitere Lösungsmöglichkeit besteht darin, durch Kollokationsverfahren zusätzlich zu den Steuerungen auch die Zustandsgrößen zeitlich zu diskretisieren und so ein Problem derart zu formulieren, dass in kompakter Schreibweise gilt:

$$
\begin{aligned}
&\min_{x^k, u^k} \left\{ J \;=\; \Phi(\boldsymbol{x}^K) + \sum_{k=0}^{K-1} \boldsymbol{f}_0^k(\boldsymbol{x}^k, \boldsymbol{u}^k) \right\} \\
&\boldsymbol{g}^k(\boldsymbol{x}^k, \boldsymbol{u}^k) \;=\; \boldsymbol{0}, \quad k = 0, 1, \ldots, K-1, (K) \\
&\boldsymbol{h}^k(\boldsymbol{x}^k, \boldsymbol{u}^k) \;\leq\; \boldsymbol{0}, \quad k = 0, 1, \ldots, K-1, (K)
\end{aligned}
\tag{3}
$$

In beiden Fällen werden bei nichtlinearen Optimierungsproblem typischerweise Sequentielle Quadratische Programmierungsverfahren (SQP-Verfahren) zu Lösung eingesetzt, die sich unterschiedlicher Varianten bedienen (z. B. Innere-Punkte-Verfahren (engl.: interior point methods), Aktive-Restriktionen-Strategien (engl.: active set methods)).

3 Aufgaben

3.1 Trinkwasserversorgungsprozess

Das Wasserversorgungssystems einer Gemeinde (siehe Abb. 1) besteht aus einem Hochbehälter mit einem Fassungsvermögen von 5000 m^3, einer Wasseraufbereitungsanlage und den Versorgungsleitungen zu den einzelnen Verbrauchern.

Das System kann durch die Differentialgleichung (Volumenbilanzgleichung)

$$
\dot{x}(t) \quad = \quad u(t) - z(t), \qquad x(0) = x_0 \tag{4}
$$

beschrieben werden. Der tägliche Verbrauch $\bar{z}$ beträgt 3000 m^3 und wird durch die Beziehung

$$
z(t) \quad = \quad \frac{\bar{z} + 1000\, sin(\frac{\pi(t-7.2}{12}) + 2000\, sin(\frac{\pi(t-6)}{6})}{24}\;, \; t \in [0, 24\; h] \tag{5}
$$

erfasst. Das Bewirtschaftungsziel besteht darin, die Versorgungssicherheit zu garantieren und durch einen möglichst gleichmäßigen Zufluss $u(t)$ eine hohe Qualität der Wasseraufbereitung zu

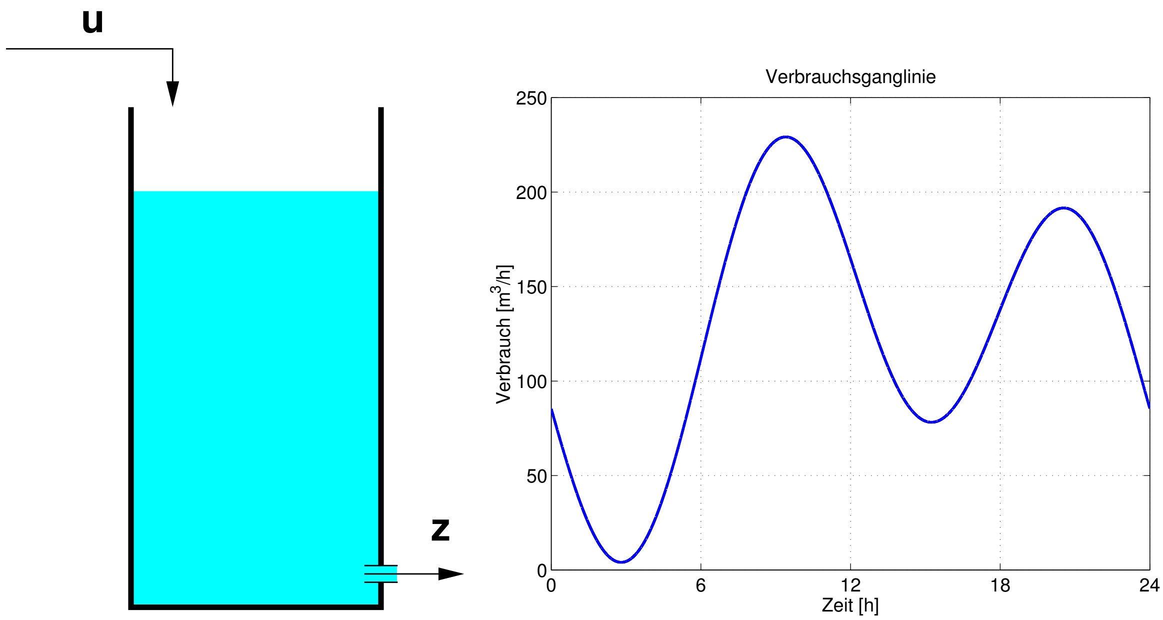

Abbildung 1: Wasserversorgungssystem und Verbrauch

sichern. Zum Zeitpunkt 6:00 Uhr soll der Behälter zu 90 % gefüllt sein. Für die Bewirtschaftungsaufgabe wird das folgende Gütefunktional

$$J = \frac{1}{2}\int_0^{24} \left(q(x(t)-x_s)^2 + r(u-z_S)^2\right) dt \tag{6}$$

formuliert, wobei mit x_S ein Sollinhalt und mit $z_S = \frac{\bar{z}}{24}$ ein mittlerer Verbrauch bezeichnet wird.

Versuchsvorbereitung:

3.1.1 Formulieren Sie zunächst das (zeitkontinuierliche) Optimalsteuerungsproblem mit allen Komponenten (Zielfunktional, Zeithorizont, Modell, Gleichungs- und Ungleichungsbeschränkungen) für den Fall der zu ermittelnden periodisch optimalen Steuerung unter der Bedingung auf, dass der Hochbehälter im Mittel zu 80 % gefüllt sein soll (Wichtungskoeffizienten $q = 1$ und $r = 10$, zeitliche Diskretisierung $\Delta t = 1$ h)! Periodisch optimal bedeutet, dass sich der Bewirtschaftungsvorgang mit einer Periode von einem Tag wiederholt und demzufolge zu Beginn und am Ende eines Zeithorizontes die Speicherinhalte gleich sein müssen. Der Zeithorizont soll von 6:00 Uhr bis 6:00 Uhr des Folgetages reichen.

3.1.2 Transformieren Sie das Optimalsteuerungsproblem in ein nichtlineares Optimierungsproblem (quasi-statisches Optimierungsproblem)! Diskretisieren Sie dazu mit der oben angegebenen festen zeitlichen Diskretisierung das Zielfunktional, die Modellgleichungen und die Beschränkungen! Stellen Sie dazu einen Optimierungsvariablenvektor, eine zeitdiskretisierte Zielfunktion entsprechend der Syntax des von Ihnen gewählten Lösungsverfahrens und anschließend eine Matrix-/Vektornotation der Gleichungs- und Ungleichungsnebendingungen auf! Belegen Sie alle Matrizen und Vektoren!

3.1.3 Implementieren Sie das unter 3.1.2 formulierte Problem unter MATLAB® oder octave! Lösen Sie das nichtlineare Optimierungsproblem!

Versuchsdurchführung:

3.1.4 Erläutern Sie Ihr Vorgehen bei der Lösung! Demonstrieren Sie die Ergebnisse und diskutieren Sie sie!

3.2 Hochwasserschutz

Eine wesentliche Aufgabe des Talsperrensystems nach Abb. 2 besteht im Hochwasserschutz. Einfache (normierte) Volumenbilanzgleichungen für die Sperren 1 und 2 berücksichtigen die

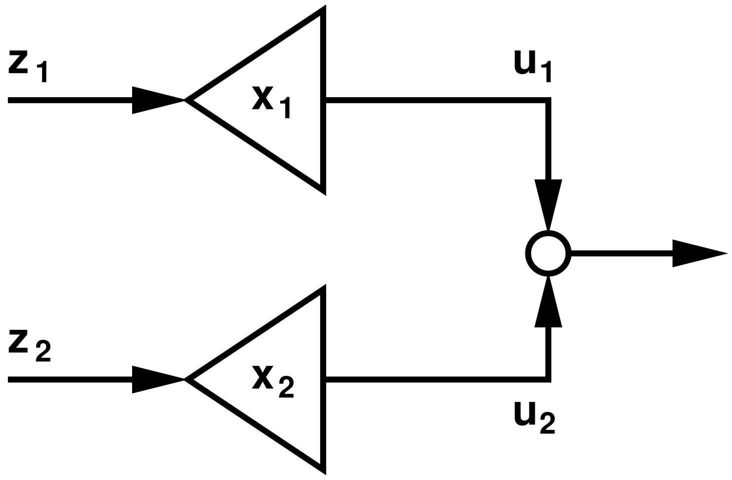

Abbildung 2: Talsperrensystem

gespeicherten Wasservolumina $x_i(t)$, die Volumenströme im Zulauf $z_i(t)$ und im Ablauf $u_i(t)$

$$\dot{x}_i(t) \quad = \quad z_i(t) - u_i(t), \qquad i = 1, 2\ , \qquad t \in [0, 1]. \tag{7}$$

Dabei wird angenommen, dass der Anfangszustand zum Zeitpunkt $t = 0$

$$\boldsymbol{x}(0) = \begin{pmatrix} 0.5 \\ 0.9 \end{pmatrix} \tag{8}$$

gegeben ist und der Zulauf im Optimierungshorizont $t \in [0, 1]$ durch

$$z_1(t) \quad = \quad \begin{cases} 0.1 + 10e^{-(t-0.15)/0.2} - 10e^{-(t-0.15)/0.1} & t > 0.15 \\ 0 & \text{sonst,} \end{cases} \tag{9}$$

$$z_2(t) \quad = \quad \begin{cases} 0.3 + 15e^{-(t-0.35)/0.2} - 15e^{-(t-0.35)/0.1} & t > 0.35 \\ 0 & \text{sonst,} \end{cases} \tag{10}$$

im Sinne einer Vorhersage beschrieben werden kann. Die Zeitverläufe der Abgaben $u_i(t)$ sind die Steuergrößen, die so bestimmt werden sollen, dass der Scheitelwert der Hochwasserwelle unterhalb der Talsperren (als Maß für ein mögliches Schadensereignis) minimal wird, d. h. das Maximum des Gesamtabflusses soll minimal werden bzw. der Gesamtabfluss soll unter Berücksichtigung der Zeitabhängigkeit möglichst niedrig sein. Dabei sind die minimalen bzw. maximalen Fassungsvermögen der beiden Talsperren

$$0 \leq x_i(t) \leq 1\ , \qquad i = 1, 2 \tag{11}$$

als auch die Nicht-Negativitätsforderung für die beiden Abflüsse $u_i(t),\ i = 1, 2$, zu beachten.

Versuchsvorbereitung:

3.2.1 Formulieren Sie zunächst das (zeitkontinuierliche) Optimalsteuerungsproblem!

3.2.2 Approximieren Sie das Optimalsteuerungsproblem durch ein nichtlineares Optimierungsproblem (quasi-statisches Optimierungsproblem) durch den Übergang zu einer zeitdiskreten Beschreibung (zeitliche Diskretisierungsschrittweite $\Delta t = 0.05$)! Stellen Sie dazu einen Optimierungsvariablenvektor, eine zeitdiskretisierte Zielfunktion, und anschließend eine Matrix-/Vektornotation der Gleichungs- und Ungleichungsnebendingungen auf!

Optional: Versuchsdurchführung:

3.2.3 Lösen Sie das entstehende Optimierungsproblem mit einem geeigneten Verfahren aus der MATLAB® Optimization Toolbox^{TM} bzw. mit octave!

3.2.4 Demonstrieren und diskutieren Sie die Ergebnisse! Diskutieren Sie Probleme, die sich bei einer praktischen Anwendung einer solchen Verfahrensweise ergeben!

Anhang: Auszug von Routinen aus der MATLAB® Optimization Toolbox^{TM} bzw. octave

```
MATLAB:
Nonlinear minimization of functions.
   fmincon    - Multidimensional constrained nonlinear minimization.
Nonlinear minimization of multi-objective functions.
   fminimax   - Multidimensional minimax optimization.
Minimization of matrix problems.
   linprog    - Linear programming.
   quadprog   - Quadratic programming.
Controlling defaults and options.
   optimset   - Create or alter optimization OPTIONS structure.
   optimget   - Get optimization parameters from OPTIONS structure.
Graphical user interface and plot routines
   optimtool  - Optimization Toolbox Graphical User
                Interface

octave:
Nonlinear minimization of functions.
   sqp    - Multidimensional constrained nonlinear minimization.
Minimization of matrix problems.
   lp     - Linear programming.
   qp     - Quadratic programming.
```

Literatur

[1] E. Arnold. *Optimale Steuerung 2: Numerische Verfahren und Beispiele.* `http://www.tu-ilmenau.de/fileadmin/media/simulation/Lehre/Beispiele/OS2_Verfahren.pdf`

[2] L.T. Biegler. *Nonlinear Programming.* MOS-SIAM Series on Optimization. SIAM. 2010.

[3] R. Fletcher. *Practical Methods of Optimization. Vol. 2: Constrained Optimization.* Wiley, Chichester 1980.

[4] R. Franke and E. Arnold. *Applying new numerical algorithms to the solution of discrete-time optimal control problems.* In: *2nd European Workshop on Computer-Intensive Methods in Control and Signal Processing.*, pp. 67-72, Prague, 1996

[5] R. Franke. *Integrierte dynamische Modellierung und Optimierung von Systemen mit saisonaler Wärmespeicherung.* Dissertation, TU Ilmenau, 1998.

[6] C.J. Goh, K.L. Teo. *Control parametrization: A unified approach to optimal control problems with general constraints.* Automatica, 24(1):3-18, 1988.

[7] P. Li. *Vorlesung Dynamische Prozessoptimierung.* TU Ilmenau.

[8] P. Li. *Entwicklung optimaler Führungsstraetgien für Batch-Destillationsprozesse.* Dissertation. TU Berlin. Reihe 3: Verfahrenstechnik. VDI-Verlag. 1998

[9] P. Li. *Prozessoptimierung unter Unsicherheiten.* Oldenbourg. 2007

[10] M. Papageorgiou. *Optimierung.* Oldenbourg. 1996

[11] S.J. Wright. *Interior-Point-Methods for Optimal Control of Discrete Time Systems.* JOTA, 77(1):161-187, 1983.

[12] The Mathworks Inc. *MATLAB®/Simulink®.* `https://www.mathworks.com/products.html`. 2020

[13] GNU Octave Software and Documentation. `https://www.gnu.org/software/octave/`, `https://www.gnu.org/software/octave/octave.pdf`. 2020