Weiterbildendes Studium Ingenieurmathematik – WS 98/99

Optimierung mit MATLAB

H.-J. Oberle, G. Opfer, S. Fredenhagen

Arbeitsblatt 13

1. Aufgabe optimaler Steuerung

Gegeben sei eine Optimalsteuerungsaufgabe

$$\text{Minimiere} \quad f(\boldsymbol{x}(t_f), t_f) \tag{1}$$

$$\text{unter den Nebenbedingungen:} \quad \boldsymbol{x}' = \boldsymbol{h}(t, \boldsymbol{x}, u), \boldsymbol{x}(t_0) = \boldsymbol{x}_0. \tag{2}$$

Diskretisiert man die Steuerung auf einem Gitter $t_0 < t_1 < ... < t_N = t_f$ und gibt sie auf den einzelnen Intervallen durch $u(t) = u_i$ für $t \in [u_i, u_{i+1})$, $i = 0, 1, ..., N-1$, als konstant vor, kann man mit einem Differentialgleichungslöser (vgl. Blatt 12 und `gety.m`) eine Lösung des Differentialgleichungssystems (2) berechnen. Diese Lösung ist zwar zulässig, da sie die Differentialgleichungen erfüllt, aber sie wird im allgemeinen kein Minimum von (1) sein. Mit Hilfe des SQP-Verfahren (vgl. Blatt 11) kann man nun die Wahl der diskretisierten Steuerung $(u_0, u_1, \ldots, u_{N-1})$ optimieren. Ist die Endzeit frei oder ist ein Anfangswert $x_i(t_0)$ nicht vorgegeben, sind dieses weitere zu optimierende Parameter. Sind Werte einer Zustandsfunktion am Ende $x_j(t_f) = x_{jf}$ oder an einem Zwischenknoten vorgegeben, so führt dieses zu Gleichungsnebenbedingungen. Auch Steuerbeschränkungen wie $u(t) \le 0$ kann man als Ungleichungsnebenbedingungen im SQP-Verfahren verwenden.

2. Aufgabe 13.1: Kubischer Spline

Die Lösung der Aufgabe optimaler Steuerung

$$\text{Minimiere} \quad \int_{t_0}^{t_f} u^2(t)dt$$

$$\text{mit} \quad \begin{aligned} x_1' &= x_2 \\ x_2' &= u \\ \boldsymbol{x}(t_0) &= \boldsymbol{x}^{(0)}, \boldsymbol{x}(t_f) = \boldsymbol{x}^{(f)} \end{aligned}$$

hat als Lösung einen kubischen Spline. Führt man eine weitere Zustandvariable $x_3(t) = \int_{t_0}^{t} u^2(\tau)d\tau$ ein, so gilt

$$x_3' = u^2, x_3(t_0) = 0$$

und das Integral kann durch $x_3(t_f)$ ersetzt werden. Für $t_0 = 0$, $t_f = 1$, $\boldsymbol{x}^{(0)} = (0, 2)$ und $\boldsymbol{x}^{(1)} = (0, -1)$ wurde das Problem in `startspline.m` und `spline.m` programmiert. Die Differentialgleichungen stehen in `dglspline.m`.

a) Rufen Sie das Programm mit

```
>> startspline
```

auf.

b) Rufen Sie das Programm mit

```
>> xopt=startspline
```

auf. Der Variable `xopt` enthält nun der Vektor $(u(t_0), u(t_1), ...)^{\mathrm{T}}$. Sehen Sie sich die Datei `startspline.m` an und wählen Sie eine feinere Diskretisierung, bei der zwischen zwei alten Knoten immer ein neuer Knoten liegt. Wählen Sie als Startschätzung `xoptneu=`$(u(t_0), u(t_0), u(t_1), u(t_1), ...)^{\mathrm{T}}$ und rufen Sie das Programm mit

```
>> startspline(xoptneu)
```

auf.

c) Sehen Sie sich die Dateien `spline.m` und `dglspline.m` an und vergleichen Sie sie mit der obengegebenen Aufgebe.

d) Lösen Sie das Problem mit $x^{(f)} = (0, 1)$.

3. Aufgabe 13.2: Steuerung einer Fähre

Die optimale Steuerung einer Fähre auf einem Fluß wird durch die folgende Problemstellung beschrieben:

$$\text{Minimiere} \quad t_f$$

$$\begin{array}{llll} \text{mit} & y_1' = w\cos(\alpha), & y_1(0) = -1, & y_1(t_f) = 1 \\ & y_2' = w\sin(\alpha) + s_0(1 - y_1^2), & y_2(0) = 0, & y_2(t_f) = 0 \end{array}$$

Sie finden die Aufgabe in den Dateien `startferry.m`, `ferry.m` und `dglferry.m`. Dabei wurde das Problem auf das feste Intervall $[0, 1]$ transformiert. Beachten Sie die geänderten Differentialgleichungen in `dglferry.m`.

a) Berechnen Sie eine Lösung des Problems für eine grobe Diskretisierung ($n = 2$) und die Startschätzung $u \equiv 0$ und $t_f = 1$ durch den Aufruf

```
>> xopt=startferry
```

Dabei hat `xopt` die Form $(t_f, u(t_0), u(t_1))^{\mathrm{T}}$.

b) Verfeinern Sie die Diskretisierung in der Datei `startferry.m`, benutzen Sie die in a) berechnete Lösung, um die Anfangsschätzung `xoptneu` zu berechnen und rufen Sie das Program mit

```
>> startferry(xoptneu)
```

auf.

c) Eine gute Startschätzung für das Problem mit $n = 16$ finden Sie in der Datei `faehr.dat`. Laden Sie die Startschätzung mit

```
>> load faehr.dat
```

und rufen Sie das Programm mit

```
>> startferry(faehr)
```

auf.

4. Aufgabe 13.3: Zug (Problem (13 a))

Im Skript steht das Problem (13.6 a):

$$\text{Minimiere} \quad I_1(u) = \int_0^{t_f} u(t)^2 dt$$

$$\text{mit} \quad \begin{array}{lll} y_1' = y_2, & y_1(0) = 0, & y_1(t_f) = a \\ y_2' = u, & y_2(0) = 0, & y_2(t_f) = 0 \end{array}$$

a) Sehen Sie sich die Dateien `spline.m`, `dglspline.m` und `startspline.m` an, kopieren Sie diese in die Dateien `zuga.m`, `dglzuga.m` und `startzuga.m` und programmieren Sie das Problem für $a = 1$ und $t_f) = 2$.

b) Vergleichen Sie ihre Ergebnisse mit den Zeichnungen im Skript. Zeichnen Sie auch einmal die Geschwindigkeit $y_2(t)$. Dazu muss die Datei `startzuga.m` geändert werden.

b) Berechnen Sie eine Lösung für eine Diskretisierung mit $n \geq 10$.

c) Fügen Sie die zusätzlichen Schranken für die Beschleunigung, $-1.2 \leq u(t) \leq 1.2$ für $t \in [0, t_f]$ ein. Hierzu muß die Datei `zuga.m` geändert werden.

5. Aufgabe 13.4: Zug (Problem (13 b))

Im Skript steht das Problem (13.6 b):

$$\text{Minimiere} \quad I_2(u) = t_f$$

$$\text{mit} \quad \begin{array}{lll} y_1' = y_2, & y_1(0) = 0, & y_1(t_f) = a \\ y_2' = u, & y_2(0) = 0, & y_2(t_f) = 0 \\ -1 \leq u(t) \leq 1, & t \in [0, t_f] & \end{array}$$

a) Sehen Sie sich die Datei `dglferry.m` an, und transformieren Sie dieses Problem entsprechend auf das feste Intervall $[0, 1]$. Programmieren Sie das Problem für $a = 1$. Sehen Sie sich dabei auch die Dateien `startferry.m` und `ferry.m` an.

b) Vergleichen Sie ihre Bilder mit denen aus dem Skript. Haben Sie die richtigen Skalierungen oder müssen Sie die Bildausgabe noch ändern. Sehen Sie sich dazu die Datei `startferry.m` an.

b) Berechnen Sie eine Lösung für eine Diskretisierung mit $n \geq 10$.