Kapitel 7

Implementierung

In diesem Kapitel werden verschiedene Punkte einer effizienten Implementierung diskutiert. Sie betreffen

- Problemformulierung,
- Berechnung von Gradienten,
- Lösung der nichtlinearen Optimierungsprobleme und
- Durchführung einer Gitteranpassung.

Sie sind bei der Implementierung des im Rahmen dieser Arbeit erstellten Algorithmus DIRCOL berücksichtigt worden.

7.1 Angepaßte Problemformulierung

In diesem Abschnitt werden einige, die Formulierung des Optimalsteuerungsproblems (AP) betreffende Techniken beschrieben, mit denen die *Effizienz* der Implementierung direkter Kollokationsverfahren deutlich verbessert werden kann.

7.1.1 Nichtautonome Probleme

Werden im Entwurf eines Algorithmus nur autonome Probleme berücksichtigt, so ist man gezwungen, ein nichtautonomes Problem um den Preis der Einführung einer zusätzlichen Differentialgleichung auf ein autonomes Problem zu transformieren (vgl. Gleichung (4.13)). Bei direkten Kollokationsverfahren erhöht sich dadurch die Anzahl der Unbekannten und der nichtlinearen Gleichungsbeschränkungen im diskretisierten Problem i. w. um die Anzahl der Gitterpunkte. Um diese Erhöhung der Anzahl der Freiheitsgrade und Beschränkungen zu vermeiden, ist es daher effizienter, nichtautonome Probleme bereits im Algorithmus zu berücksichtigen.

7.1.2 Unterscheidung expliziter Rand- und Schaltbedingungen

Anstatt spezielle Rand-, Schalt- oder Stufungsbedingungen, die direkt Werte für die Zustandsvariablen vorschreiben, in der allgemeinen nichtlinearen impliziten Formulierung von Gleichung (1.8) bzw. Gleichung (1.9) zu berücksichtigen, kann z. B. die folgende Unterscheidung *expliziter* Randbedingungen

$$\begin{aligned} x_i(t_0) &= \text{const.}, & & i \in \{1,\ldots,n\}, \\ x_j(t_f) &= R_j^1(x(t_0), t_0, t_f), & & j \in \{1,\ldots,n\},\ R_j^1 : \mathbb{R}^{n+2} \to \mathbb{R}^1, \end{aligned} \tag{7.1}$$

sowie von Schalt- bzw. Stufungsbedingungen

$$\begin{array}{rcll} x_i(E_k - 0) & = & \text{const.}, & i \in \{1, \dots, n\}, \\ x_j(E_k + 0) & = & R_j^k(x(E_k - 0), E_k), & j \in \{1, \dots, n\},\ R_j^k : \mathbb{R}^{n+1} \to \mathbb{R}^1, \\ k & = & 2, \dots, M-1, & \end{array} \tag{7.2}$$

benutzt werden. Das hat den Vorteil, daß einige Freiheitsgrade des diskretisierten Optimierungsproblems (DP1) bzw. (DP2) durch vorgeschriebene Funktionswerte eliminiert werden können. Somit wird die Anzahl der Freiheitsgrade und der Gleichungsbeschränkungen reduziert. Ein „robusteres“ Konvergenzverhalten bei geringerem Rechenzeitbedarf ist dadurch zu erwarten.

7.1.3 Unterscheidung zwischen einfachen Schranken und nichtlinearen Beschränkungen

Bei nichtlinearen Ungleichungsbeschränkungen (1.10) können einfache untere und obere Schranken für die Zustands- und Steuervariablen

$$\begin{array}{rcccl} x_{i,\min}^k & \le & x_i(t) & \le & x_{i,\max}^k, \quad i = 1, \dots, n, \\ u_{j,\min}^k & \le & u_j(t) & \le & u_{j,\max}^k, \quad j = 1, \dots, l, \\ & & & & \quad k = 1, \dots, M-1, \end{array} \tag{7.3}$$

in jeder Phase $E_k \le t \le E_{k+1}$ unterschieden werden, ebenso für die Steuerparameter und die Ereignisse (der Anfangszeitpunkt $E_1 = t_0$ bleibt fest)

$$\begin{array}{rcccl} p_{j,\min} & \le & p_j & \le & p_{j,\max}, \quad j = 1, \dots, l_P, \\ E_{k,\min} & \le & E_k & \le & E_{k,\max}, \quad k = 2, \dots, M. \end{array} \tag{7.4}$$

Die separate Berücksichtigung solcher oberer und unterer Schranken in einem direkten Kollokationsverfahren hat zwei Vorteile: Zum einen werden bei Verwendung numerischer Differenzenquotienten für die Gradienten der Ungleichungsbeschränkungen zahlreiche Funktionsauswertungen eingespart und zum anderen erlauben nichtlineare Optimierungsverfahren (wie das verwendete SQP-Verfahren NPSOL) meist die separate Berücksichtigung solcher Schranken, die dann, im Gegensatz zu nichtlinearen Beschränkungen, *in jedem Iterationsschritt* eingehalten werden können, was ein robusteres und effizienteres Optimierungsverfahren zur Folge hat.

7.1.4 Hinzunahme unterer und oberer Schranken

Um zu verhindern, daß das Parameteroptimierungsverfahren während einzelner Iterationsschritte in physikalisch unsinnige Bereiche gerät (wie z. B. eine negative Flughöhe oder eine negative Gesamtmasse bei flugmechanischen Problemen), was zur Divergenz des direkten Kollokationsverfahrens oder auch zum Abbruch des Programmlaufes (z. B. wegen versuchter Auswertung von $\sqrt{x}$ für $x < 0$) führen kann, kann es notwendig sein, untere und obere Grenzen für Zustands- und Steuervariablen, sowie für die Steuerparameter und die Ereignisse zu spezifizieren. Diese sind dabei so eng als möglich zu wählen, ohne daß dadurch jedoch die optimale Bahn des ursprünglichen Problems (fälschlicherweise) beschränkt wird. D. h. es ist für eine berechnete Lösung zu überprüfen, ob diese zusätzlichen Beschränkungen für die berechnete Näherungslösung aktiv werden.

7.1.5 Skalierungen

Eine geeignete Skalierung aller auftretenden abhängigen und unabhängigen Variablen kann sich in einem Optimierungsverfahren einerseits auf die Kondition von auftretenden Matrizen höherer (partieller) Ableitungen und andererseits auch auf die Abbruch- und Konvergenzkriterien auswirken, die implizite Definitionen von „groß“ und „klein“ verwenden (siehe Kapitel 7.5, 8.4.2.2 und 8.7 bei Gill, Murray und Wright [49]). Eine „schlechte“ Skalierung kann selbst bei „gutartigen“ Problemen zur Divergenz numerischer Verfahren führen, während eine „gute“ Skalierung die Konvergenzeigenschaften hinsichtlich Robustheit gegenüber schlechten Startwerten verbessern kann. Auch werden weniger Iterationen benötigt. Die Skalierungsinvarianz (Affininvarianz) ist daher eine erstrebenswerte Eigenschaft für viele Algorithmen. Die *theoretische* Skalierungsinvarianz kann jedoch in der Praxis aufgrund der endlichen Arithmetik nicht im gesamten Wertebereich gesichert werden. Eine „gute“ Skalierung ist daher selbst bei affininvarianten Algorithmen von Vorteil.

Die Grundgedanken bei Skalierung durch lineare Transformationen und deren Auswirkungen seien im folgenden kurz skizziert.

Skalierung von $\boldsymbol{x}$, $\boldsymbol{u}$, $\boldsymbol{p}$ und $\boldsymbol{t_f}$

Alle (skalierten) $x_i(t)$ und $u_j(t)$ sollten ungefähr im selben Bereich von etwa $[-1,+1]$ liegen und auch *laufen*. Wir betrachten zunächst einstufige Systeme (d. h. $M = 2$, keine Schaltpunkte). Skalierungen lassen sich dann durch lineare Transformationen der Art

$$
\begin{array}{rcllcll}
\bar{x}_i(t) &=& (x_i(t) - X_{i,\mathrm{m}})\,/X_{i,\mathrm{s}}, & X_{i,\mathrm{m}},\ X_{i,\mathrm{s}} &=& \text{const.} \neq 0, & i = 1,\ldots,n,\\
\bar{u}_j(t) &=& (u_j(t) - U_{j,\mathrm{m}})\,/U_{j,\mathrm{s}}, & U_{j,\mathrm{m}},\ U_{j,\mathrm{s}} &=& \text{const.} \neq 0, & j = 1,\ldots,l,\\
\bar{p}_j &=& (p_j - P_{j,\mathrm{m}})\,/P_{j,\mathrm{s}}, & P_{j,\mathrm{m}},\ P_{j,\mathrm{s}} &=& \text{const.} \neq 0, & j = 1,\ldots,l_P,\\
\bar{t}_f &=& (t_f - t_0)/T, & T &=& \text{const.} > 0 &
\end{array}
\tag{7.5}
$$

beschreiben. Die neue unabhängige Veränderliche ist also

$$\bar{t} \;=\; (t - t_0)/T \;\in\; [0, (t_f - t_0)/T].$$

Damit transformiert sich die i-te Differentialgleichung wegen

$$\dot{\bar{x}}(\bar{t}) \;=\; \frac{\mathrm{d}\bar{x}_i(\bar{t})}{\mathrm{d}\bar{t}} \;=\; \frac{1}{X_{i,\mathrm{s}}}\frac{\mathrm{d}x(\bar{t})}{\mathrm{d}\bar{t}} \;=\; \frac{1}{X_{i,\mathrm{s}}}\frac{\mathrm{d}x(t)}{\mathrm{d}t}\frac{\mathrm{d}t}{\mathrm{d}\bar{t}} \;=\; \frac{T}{X_{i,\mathrm{s}}}\dot{x}(t)$$

zu

$$\dot{\bar{x}}(\bar{t}) \;=\; \frac{T}{X_{i,\mathrm{s}}}\, f\,(\,X_{\mathrm{s}}\bar{x}(\bar{t}) + X_{\mathrm{m}},\ U_{\mathrm{s}}\bar{u}(\bar{t}) + U_{\mathrm{m}},\ P_{\mathrm{s}}\bar{p} + P_{\mathrm{m}},\ t_0 + T\bar{t}\,)\,, \tag{7.6}$$

mit $X_{\mathrm{s}} = \mathrm{diag}(X_{1,\mathrm{s}},\ldots,X_{n,\mathrm{s}})$, $X_{\mathrm{m}} = (X_{1,\mathrm{m}},\ldots,X_{n,\mathrm{m}})^T$ und U_{s}, U_{m}, P_{s}, P_{m} entsprechend.

Sind x_i oder u_k zeitabhängige Winkel, so werden sie in [rad] gemessen und brauchen wegen $x_i(t)$, $u_k(t) \in [\,-\pi, +\pi]$ nicht skaliert zu werden.

Werden bei Problemen mit Schaltpunkten (mehrstufige Probleme) auch die Ereignisgrößen transformiert

$$\bar{E}_k = (E_k - E_{k,\mathrm{m}})\,E_{k,\mathrm{s}}/T, \quad E_{k,\mathrm{m}} \neq 0,\ E_{k,\mathrm{s}} > 0, \quad k = 2,\ldots,M, \tag{7.7}$$

so gilt in der k-ten Stufe

$$\begin{aligned}\hat{t} &= (t - E_{k,\mathrm{m}})E_{k,\mathrm{s}}/T, \\ \dot{\hat{x}}(\hat{t}) &= \frac{T}{X_{i,\mathrm{s}}E_{k,\mathrm{s}}} f(.,.,., E_{k,\mathrm{m}} + T\hat{t}/E_{k,\mathrm{s}}).\end{aligned} \tag{7.8}$$

In der Optimierung wird mit den skalierten Größen gerechnet.

Skalierung der nichtlinearen Beschränkungen $\boldsymbol{g} >= 0$ und $\boldsymbol{h}^e = 0$

„Gut“ skalierte Beschränkungen sind erstens gut konditioniert hinsichtlich Änderungen in den Veränderlichen $x(t_i)$, $u(t_i)$, p und t_f, und zweitens haben verschiedene Beschränkungen „gleiches Gewicht“ in der Lösung. Gleichungs- und Ungleichungsbeschränkungen des optimalen Steuerungsproblems (AP) können mit

$$\begin{aligned}\bar{g}_i^k(x,u,p,t) &= g_i^k(x,u,p,t)\cdot G_{i,\mathrm{s}}^k, & G_{i,\mathrm{s}}^k &= \text{const.} > 0, \quad i = 1,\ldots,n_{g,\mathrm{nln},k}, \\ \bar{h}_j^{e\,k}(x,u,p,t) &= h_j^{e\,k}(x,u,p,t)\cdot H_{j,\mathrm{s}}^k, & H_{j,\mathrm{s}}^k &= \text{const.} > 0, \quad i = 1,\ldots,n_{h,\mathrm{nln},k},\end{aligned} \tag{7.9}$$

in jeder Phase $k = 1,\ldots,M-1$, skaliert werden. Entsprechend werden auch die nichtlinearen Rand- und Schaltbedingungen (1.8) und (1.9) skaliert.

Skalierung der Zielfunktion $\boldsymbol{\Phi}$

Auch eine Skalierung der Zielfunktion kann wirkungsvoll sein. Konstanten, die zu einem nichtkonstanten Anteil von Φ hinzuaddiert werden, sollten weggelassen werden. Bei vielen Problemen hat es sich in der Praxis gut bewährt, Φ so zu skalieren, daß der optimale Wert etwa bei 100 liegt, und daß der Bereich von 0 bis 100 auch der tatsächliche Schwankungsbereich ist, den $\Phi(\tilde{x}(\tilde{t}_f), \tilde{t}_f)$ für verschiedene Werte von $\tilde{x}(\tilde{t}_f)$ und $\tilde{t}_f$ in der Iteration annehmen kann

$$\bar{\Phi} = (\Phi - \Phi_\mathrm{m})/\Phi_\mathrm{s}, \quad \Phi_\mathrm{m} = \text{const.} \neq 0,\ \Phi_\mathrm{s} = \text{const.} > 0. \tag{7.10}$$

Es ist zum Beispiel meist effektiver, statt

$$\Phi = 1000 + x_1(t_f)$$

die Skalierung

$$\bar{\Phi} = 100 \cdot (x_1(t_f) - X_{1,\mathrm{m}})/X_{1,\mathrm{s}}$$

(mit $X_{1,\mathrm{s}} > 0$) zu verwenden.

Realisierung der Skalierung

Die Ein- und Ausgabe sowie die Formulierung der Differentialgleichungen etc. erfolgt in den ursprünglichen Einheiten. Die Skalierungen und Reskalierungen werden im Programm *intern* durchgeführt. Der Benutzer kann also mit den vertrauten Maßeinheiten operieren und ist nicht gezwungen, die linearen Transformationen selbst zu programmieren.

Im Programm DIRCOL sind mehrere Optionen zur Durchführung der Skalierungen implementiert. Damit ist es u. a. möglich, in jeder Phase $[E_i, E_{i+1}]$ andere Skalierungen für die Zustands- und Steuervariablen sowie für die nichtlinearen Gleichungs- und Ungleichungsbeschränkungen zu verwenden:

1) Keine Skalierung (d. h. $X_{i,\mathrm{s}} = 1$, $X_{i,\mathrm{m}} = 0$, $\Phi_\mathrm{s} = 1$, $\Phi_\mathrm{m} = 0$, $G_{i,\mathrm{s}} = 1$, etc.),

2) Verwendung von Skalierungen durch lineare Transformationen, die alle vom Benutzer in einer speziellen Eingabedatei spezifiziert werden müssen, oder

3) Verwendung von Skalierungen durch lineare Transformationen, die vom Programm selbst nach den folgenden Kriterien berechnet werden:

   - Durch Untersuchung der Werte der Starttrajektorie von $x_i(t)$ an den Punkten t_j, $t_{j+1/2}$, t_{j+1}, $j = 1, \ldots, N-1$, werden Näherungen für $\max\{x_i(t) : \ t \in [t_0, t_f]\}$ und $\min\{x_i(t) : \ t \in [t_0, t_f]\}$ bestimmt, mit denen dann $X_{i,\mathrm{s}}$ und $X_{i,\mathrm{m}}$ berechnet werden können ($i = 1, \ldots, n$). Analog wird für die Steuerungen vorgegangen.
   - Durch Untersuchung der Werte der Beschränkungen g und h^e für die Starttrajektorien von $x(t)$, $u(t)$ an den Punkten t_j, $t_{j+1/2}, t_{j+1}$ und für die Startwerte von p und E werden Näherungen $g_{i,\max}^k$ für $\max\{g_i^k(x(t), u(t), p, t) : \ t \in [E_k, E_{k+1}]\}$ und $h_{j,\max}^{e\,k}$ für $\max\{|h_j^{e\,k}(x(t), u(t), p, t)| : \ t \in [E_k, E_{k+1}]\}$ bestimmt. Die Skalierungsfaktoren werden dann nach

$$G_{i,\mathrm{s}}^k = \begin{cases} 1, & \text{falls } g_{i,\max}^k = 0, \\ \max(\epsilon_H, 1) / g_{i,\max}^k, & \text{sonst}, \end{cases} \quad , \ i = 1, \ldots, n_{g,\mathrm{nln},k},$$

$$\text{mit } \epsilon_H = \begin{cases} 1, & \text{falls } \epsilon_{\mathrm{NFT}} \le \sqrt{\epsilon_{\mathrm{MACH}}}, \\ 10^{0.5 \cdot \log_{10}(\epsilon_{\mathrm{NFT}} / \sqrt{\epsilon_{\mathrm{MACH}}})}, & \text{sonst}, \end{cases}$$

$$\text{und } H_{j,\mathrm{s}}^k = \begin{cases} 1, & \text{falls } h_{j,\max}^k = 0, \\ 100 \cdot \epsilon_{\mathrm{NFT}} / h_{j,\max}^k, & \text{sonst}, \end{cases} \quad , \ j = 1, \ldots, n_{h,\mathrm{nln},k},$$

     berechnet. Dabei ist ϵ_{MACH} die relative Maschinengenauigkeit und ϵ_{NFT} die gewünschte Genauigkeit in den nichtlinearen Beschränkungen (Kapitel 7.3.3).
   - Durch Variation der Werte von $x(t)$ an den Rand- bzw. Schaltpunkten, den Ereignissen und Steuerparametern wird versucht, eine Schwankungsbreite der Zielfunktion (Gleichung (1.12)) zu bestimmen, mit der dann Φ_s und Φ_m berechnet werden.

Diese Kriterien haben sich bei vielen optimalen Steuerungsproblemen als nützlich erwiesen. Es sind jedoch leicht Probleme konstruierbar, wo diese Kriterien schlechte Resultate liefern können. Sie sind daher lediglich als Hilfestellung für Benutzer des Programms DIRCOL zu betrachten.

7.2 Gradientenberechnung

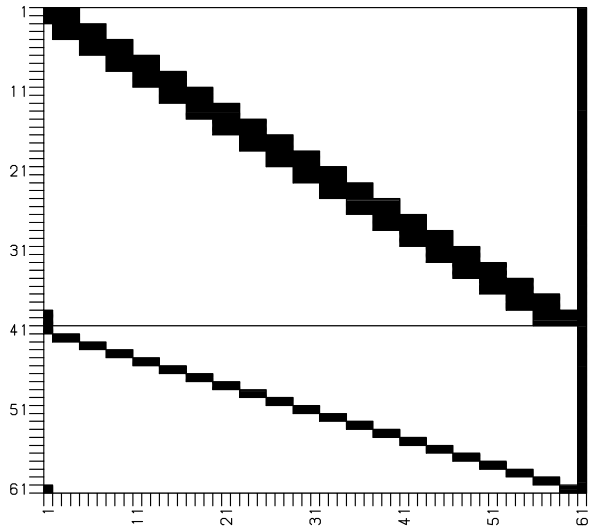

Abbildung 7.1: Jacobi-Matrix der Diskretisierung (DP2) mit 21 Gitterpunkten bei der Brachistochrone mit Wegbeschränkung (Kapitel 5.3.2): Die ersten 40 Zeilen resultieren aus den Kollokationsbedingungen, die letzten 21 aus der Ungleichungsbeschränkung, die Einträge der letzten Spalte enthalten die Ableitungen nach der freien Endzeit $\partial/\partial t_f$.

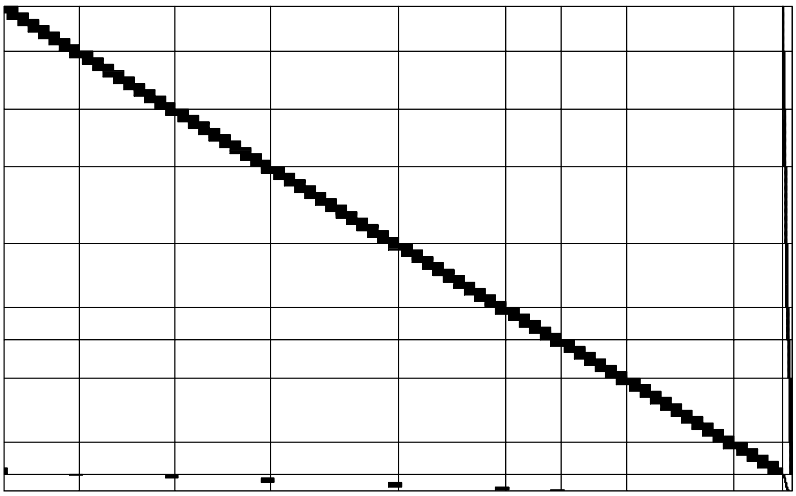

Abbildung 7.2: Jacobi-Matrix der Diskretisierung (DAP2) mit Berücksichtigung der Schaltstruktur (9 Phasen) bei der zeitoptimalen Steuerung des Roboters Manutec r3 (Kapitel 8.4.3): Horizontale und vertikale Hilfslinien kennzeichnen die den einzelnen Phasen zugeordneten Beschränkungen und Freiheitsgrade. Die letzten Zeilen enthalten die Gradienten der nichtlinearen Schaltbedingungen, die letzten Spalten die Abhängigkeiten von den Ereignissen.

Zur numerischen Lösung der endlich-dimensionalen nichtlinearen Optimierungsaufgaben werden die Gradienten von Zielfunktion und Beschränkungen der Probleme (DP1), (DP2) bzw. (DAP2) benötigt. Diese sind strukturiert und dünnbesetzt. Beispiele für Jacobi-Matrizen sind in den Abbildungen 7.1 und 7.2 dargestellt. Dort sind die Blöcke schwarz eingefärbt, in denen a priori ein von null verschiedener Eintrag erwartet werden kann.

Die Struktur der Gradienten wird in DIRCOL insoweit ausgenützt, als nur die Einträge berechnet werden, in denen ein von null verschiedener Wert erwartet werden kann. Approximationen der benötigten Gradienten werden durch Vorwärtsdifferenzenquotienten nach dem Algorithmus von Gill, Murray, Saunders und Wright [48], [49] berechnet.

7.3 SQP-Verfahren

In der letzten Dekade haben Verfahren der sequentiellen (auch sukzessiven oder rekursiven) quadratischen Programmierung (SQP) große Verbreitung (in der Praxis) gefunden. Eine ihrer herausragenden Eigenschaften ist die geringe Anzahl benötigter Funktionsauswertungen im Vergleich zu anderen Verfahren [125]. Für einen historischen Überblick über die Entstehungsgeschichte der SQP-Verfahren sei auf die Zitate bei Schittkowski [126] und Fletcher [45] verwiesen. Zur Beschreibung der elementaren Eigenschaften von SQP-Verfahren werden wir uns im wesentlichen an der Darstellung von Schittkowski [126], [127], [128] orientieren.

7.3.1 Prinzip

Wir betrachten das nichtlineare Optimierungsproblem (NP) aus Kapitel 4.2 (Gleichungen (4.31) – (4.33)): Minimiere

$$\Phi(y), \quad y \in \mathbb{R}^n,$$

unter den Restriktionen

$$\begin{aligned} a_i(y) &= 0,\ i = 1, \ldots, m_e, \\ b_j(y) &\geq 0,\ j = 1, \ldots, m, \end{aligned}$$

mit $n,\ m_e,\ m \in \mathbb{N}$ und stetig differenzierbaren reellwertigen Funktionen Φ, a_i und b_j.

Die grundlegende Idee beruht auf der Formulierung und Lösung eines quadratischen Teilproblems (QP) in jeder Iteration zur Bestimmung der *Suchrichtung*. Das QP erhält man durch Linearisierung der Beschränkungen und quadratische Approximation der Lagrangefunktion $L(y, \mu, \sigma)$ aus Gleichung (4.34)

$$L(y, \mu, \sigma) = \Phi(y) - \sum_{i=1}^{m_e} \mu_i a_i(y) - \sum_{j=1}^{m} \sigma_j b_j(y),\ \mu \in \mathbb{R}^{m_e},\ \sigma \in \mathbb{R}^m.$$

Zur Formulierung des quadratischen Teilproblems gehen wir von einer gegebenen Iterierten $y_k \in \mathbb{R}^n$ (Approximation der Lösung), $v_k = (\mu_1^{(k)}, \ldots, \mu_{m_e}^{(k)}, \sigma_1^{(k)}, \ldots, \sigma_m^{(k)})^T \in \mathbb{R}^{m_e+m}$ (Approximationen der Multiplikatoren) und $B_k \in \mathbb{R}^{n \times n}$ (Approximation der Hesse-Matrix von L) aus. Dann lautet das quadratische Teilproblem (wobei k der Iterationsindex ist)

$$\min_d \frac{1}{2} d^T B_k d + \nabla \Phi(y_k)^T d \tag{7.11}$$

unter

$$\begin{aligned} \nabla a_i(y_k)^T d + a_i(y_k) &= 0, \ i = 1, \ldots, m_e, &(7.12)\\ \nabla b_j(y_k)^T d + b_j(y_k) &\geq 0, \ j = 1, \ldots, m. &(7.13)\end{aligned}$$

Es sei d_k die Lösung und u_k der zugehörige Multiplikator. Eine neue Iterierte erhält man durch

$$\begin{pmatrix} y_{k+1} \\ v_{k+1} \end{pmatrix} = \begin{pmatrix} y_k \\ v_k \end{pmatrix} + \alpha_k \begin{pmatrix} d_k \\ u_k - v_k \end{pmatrix}. \tag{7.14}$$

Dabei ist $\alpha_k \in \mathbb{R}$ eine *Schrittweite*, die geeignet gewählt wird, um globale Konvergenzeigenschaften beim Start von beliebigen Anfangswerten, d. h. von einem $y_0 \in \mathbb{R}$, $v_0 := 0$ und $B_k := I$, zu erhalten. Die Schrittweite sollte dabei mindestens die Abnahme der „Gütefunktion"

$$\phi_r(\alpha) := \psi_r \left(\begin{pmatrix} y \\ v \end{pmatrix} + \alpha \begin{pmatrix} d \\ u - v \end{pmatrix} \right),$$

mit einer geeigneten Straffunktion $\psi_r(y, v)$, $r = (r_1, ..., r_{m_e+m})^T \in \mathbb{R}^{m_e+m}$, $r_i > 0$, garantieren. Verbreitet sind die nicht differenzierbare L_1-Straffunktion (Han [56] und Powell [111])

$$\psi_r(y, v) := \Phi(y) + \sum_{j=1}^{m_e} r_j |a_j(y)| + \sum_{j=1}^{m} r_{m_e+j} |\min(0, b_j(y))| \tag{7.15}$$

sowie die differenzierbare erweiterte Lagrangefunktion (Schittkowski [126], [128])

$$\psi_r(y, v) := \Phi(y) - \sum_{j=1}^{m_e} \left(\mu_j a_j(y) - \frac{1}{2} r_j a_j(y)^2 \right) - \sum_{j \in J} \left(\sigma_j b_j(y) - \frac{1}{2} r_{m_e+j} b_j(y)^2 \right) - \frac{1}{2} \sum_{j \in K} \frac{\sigma_j^2}{r_{m_e+j}} \tag{7.16}$$

mit den Indexmengen

$$J = \{j | 1 \leq j \leq m, \ b_j(y) \leq \sigma_j / r_{m_e+j}\}, \quad K = \{1, \ldots, m\} \setminus J.$$

Die erweiterte Lagrangefunktion (7.16) besitzt dabei einige theoretische und numerische Vorteile gegenüber der exakten L_1-Straffunktion (s. Schittkowski [126]).
Bei den hier zu lösenden nichtlinearen Optimierungsproblemen besitzt diese Gütefunktion einen weiteren wichtigen Vorteil: Die Lagrangefunktionen von (DP1) und (DP2) lassen sich als *Diskretisierungen erweiterter Funktionale* interpretieren (Kapitel 4.3 und 4.4). Bei ihrer Minimierung wird auch das *Minimumprinzip* (in einer speziellen Form) erfüllt.

Zur Vermeidung der Berechnung der zweiten Ableitungen und zur Erhaltung einer dennoch lokal superlinearen Konvergenzrate wird die Approximation B_k der Hesse-Matrix von L meist nach der BFGS-quasi-Newton Formel erneuert. Dabei hängt die Berechnung von B_{k+1} nur von B_k und zwei Vektoren ab

$$\begin{aligned} q_k &:= \nabla_y L(y_{k+1}, u_k) - \nabla_y L(y_k, u_k), \\ p_k &:= y_{k+1} - y_k, \\ B_{k+1} &:= B_k + \frac{1}{q_k^T p_k} q_k q_k^T - \frac{1}{p_k^T B_k p_k} B_k p_k p_k^T B_k. &(7.17)\end{aligned}$$

Falls B_k positiv definit ist und falls $q_k^T p_k > 0$, dann ist nach der obigen Formel auch B_{k+1} positiv definit. Die Verwendung der BFGS-Formel setzt dabei geeignete Schätzungen u_k der Lagrangeschen Multiplikatoren voraus.
Zwei weitere Strategien zur Wahl von B_k im quadratischen Teilproblem werden von Gill, Murray und Wright in [50] zitiert und kurz beschrieben.

Theoretische Eigenschaften von SQP-Verfahren werden z. B. von Schittkowski in [126] und [128] untersucht.

7.3.2 Einige praktische Details

Quadratische Teilprobleme ohne zulässige Lösung

Das quadratische Teilproblem (7.11) – (7.13) hat möglicherweise, wegen inkonsistenter (linearisierter) Beschränkungen, keine Lösung. Durch Einführung einer zusätzlichen Variablen $\delta \in \mathbb{R}$ kann man jedoch ein $(n+1)$-dimensionales quadratisches Teilproblem mit konsistenten Beschränkungen erhalten (Powell [111]):

$$\min_d \frac{1}{2} d^T B_k d + \nabla\Phi(y_k)^T d + \rho_k \delta^2 \tag{7.18}$$

unter

$$\begin{aligned}
\nabla a_i(y_k)^T d + (1-\delta) a_i(y_k) &= 0,\ i = 1, \ldots, m_e, && (7.19)\\
\nabla b_j(y_k)^T d + (1-\delta) b_j(y_k) &\geq 0,\ j \in J_k, && (7.20)\\
\nabla b_j(y_k)^T d + b_j(y_k) &\geq 0,\ j \in K_k, && (7.21)
\end{aligned}$$

mit Indexmenge J_k der aktiven Ungleichungsbeschränkungen

$$J_k = \{j | 1 \leq j \leq m,\ b_j(y_k) \leq \epsilon \text{ oder } \sigma_j^{(k)} > 0\}, \quad K_k = \{1, \ldots, m\} \setminus J_k.$$

Dabei ist ϵ eine kleine Toleranz zur Definition der aktiven Ungleichungsbeschränkungen. Offensichtlich erfüllt der Punkt $d_0 := 0$, $\delta_0 := 1$ die linearen Beschränkungen. Somit ist das erweiterte quadratische Teilproblem immer lösbar. Der Strafparameter ρ_k ist in Abhängigkeit von der Gütefunktion passend zu wählen (Schittkowski [128]).

Eine andere Möglichkeit besteht darin, einen Vektor d zu bestimmen, der eine gewichtete Summe der verletzten (linearisierten) Beschränkungen und der quadratischen Zielfunktion minimiert (Gill, Murray, Wright [50] und die dort zitierten Arbeiten). Diese Variante ist im Algorithmus NPSOL verwendet worden (Gill et al. [51]).

Modifikation des BFGS-Updates

Für den Fall $q_k^T p_k <= 0$ hat Powell [111] eine Modifikation der BFGS-Formel angegeben, die sicherstellt, daß B_{k+1} positiv definit ist. In der Formel (7.17) wird dazu die Definition von q_k ersetzt durch

$$\begin{aligned}
q_k &:= \Theta \hat{q}_k + (1-\Theta) B_k p_k,\\
\hat{q}_k &:= \nabla_y L(y_{k+1}, u_k) - \nabla_y L(y_k, u_k),\\
\Theta &:= \begin{cases} 1, & \text{falls } \hat{q}_k^T p_k \geq 0.2 p_k^T B_k p_k,\\ \frac{0.8 p_k^T B_k p_k}{p_k^T B_k p_k - p_k^T \hat{q}_k}, & \text{sonst.} \end{cases}
\end{aligned}$$

Diese Wahl von Θ garantiert, daß

$$q_k^T p_k \geq 0.2 p_k^T B_k p_k$$

und daß alle Matrizen B_k positiv definit bleiben, sofern B_0 positiv definit ist.

Gill et al. beschreiben eine weitere Möglichkeit zur Wahl von q_k, die im Algorithmus NPSOL verwendet worden ist [51].

In der Praxis ist es meist effizienter, von Iteration zu Iteration eine geeignete Faktorisierung der Matrix B_k, die in der Lösung des quadratischen Teilproblems verwendet wird, zu erneuern als diese selbst (vgl. z. B. Schittkowski [127], Gill et al. [51]).

Untere und obere Schranken für die Variablen

Da in den praktischen Anwendungen von Optimierungsverfahren die Problemstellungen häufig obere und untere Schranken

$$y_{\min} \leq y \leq y_{\max} \tag{7.22}$$

enthalten, und weil deren Einhaltung in jedem Iterationsschritt zudem die Robustheit des Verfahrens erhöht, werden in allen gängigen Implementierungen von SQP-Verfahren diese Schranken separat berücksichtigt. Somit werden die linearen Beschränkungen im quadratischen Teilproblem um

$$y_{\min} - y_k \leq d \leq y_{\max} - y_k \tag{7.23}$$

erweitert (z. B. Schittkowski [127]).

Lösung des quadratischen Teilproblems

Schittkowski [127] hat die Formulierung des quadratischen Teilproblems als lineares Ausgleichsproblem vorgeschlagen: Das quadratische Teilproblem kann als lineares (beschränktes) Ausgleichsproblem angesehen werden, welches wiederum auf ein nicht negatives lineares Ausgleichsproblem transformiert werden kann.

Da im SQP-Verfahren eine Folge verwandter QPe gelöst werden muß, ist eine „Warm Start" Option für das Unterprogramm zur Lösung der quadratischen Teilprobleme, die es beispielsweise erlaubt, Faktorisierungen und Arbeitsmengen der vorherigen Iteration wieder zu verwenden, sehr effizienzsteigernd [50].

7.3.3 (Weitere) Besonderheiten von NPSOL

Zur Lösung der durch die Diskretisierungen auftretenden nichtlinearen Optimierungsprobleme (DP1) und (DP2) wird das SQP-Verfahren NPSOL von Gill, Murray, Saunders und Wright [51] verwendet. Neben den teilweise bereits erwähnten Eigenschaften werden die folgenden in DIRCOL ausgenützt:

- **Problemformulierung:**
  Der Algorithmus behandelt einfache untere und obere Schranken, lineare und nichtlineare Beschränkungen separat. Schranken sowie lineare Beschränkungen werden dabei in jedem Iterationsschritt eingehalten. Dies ermöglicht bei variablen Gitterpunkten die Einhaltung der Gleichungen (6.45), die die aufsteigende Ordnung der Gitterpunkte garantieren.

- **Abbruchkriterien:**
  NPSOL terminiert erfolgreich, wenn die Folge der Iterierten (y_k) einen Punkt y enthält, der die KKT Bedingungen erster Ordnung (s. Kapitel 4.2) erfüllt. Dazu werden drei Kriterien überprüft:

$$\begin{array}{lrcl} (1) & \alpha||d|| & \leq & \sqrt{\epsilon_{\text{OPT}}}(1+\|y\|), \\ (2) & ||Z^T g|| & \leq & \sqrt{\epsilon_{\text{OPT}}}(1+\max\{1+|\Phi(y)|,\|g\|\}), \\ (3) & |a_j(y)| & \leq & \epsilon_{\text{NFT}}, \quad b_k(y) \;\geq\; -\epsilon_{\text{NFT}}, \end{array} \tag{7.24}$$

  dabei sind a_j, b_k nichtlineare Beschränkungen, g ist der Gradient von Φ nach den freien, nicht gebundenen Variablen und $Z^T g$ ist der projizierte Gradient. Es sind ϵ_{OPT} die Optimalitätsgenauigkeit (optimality tolerance) und ϵ_{NFT} die nichtlineare Beschränkungsgenauigkeit (nonlinear feasibility tolerance), die beide vom Benutzer spezifiziert werden können.

7.4 Ablaufschema

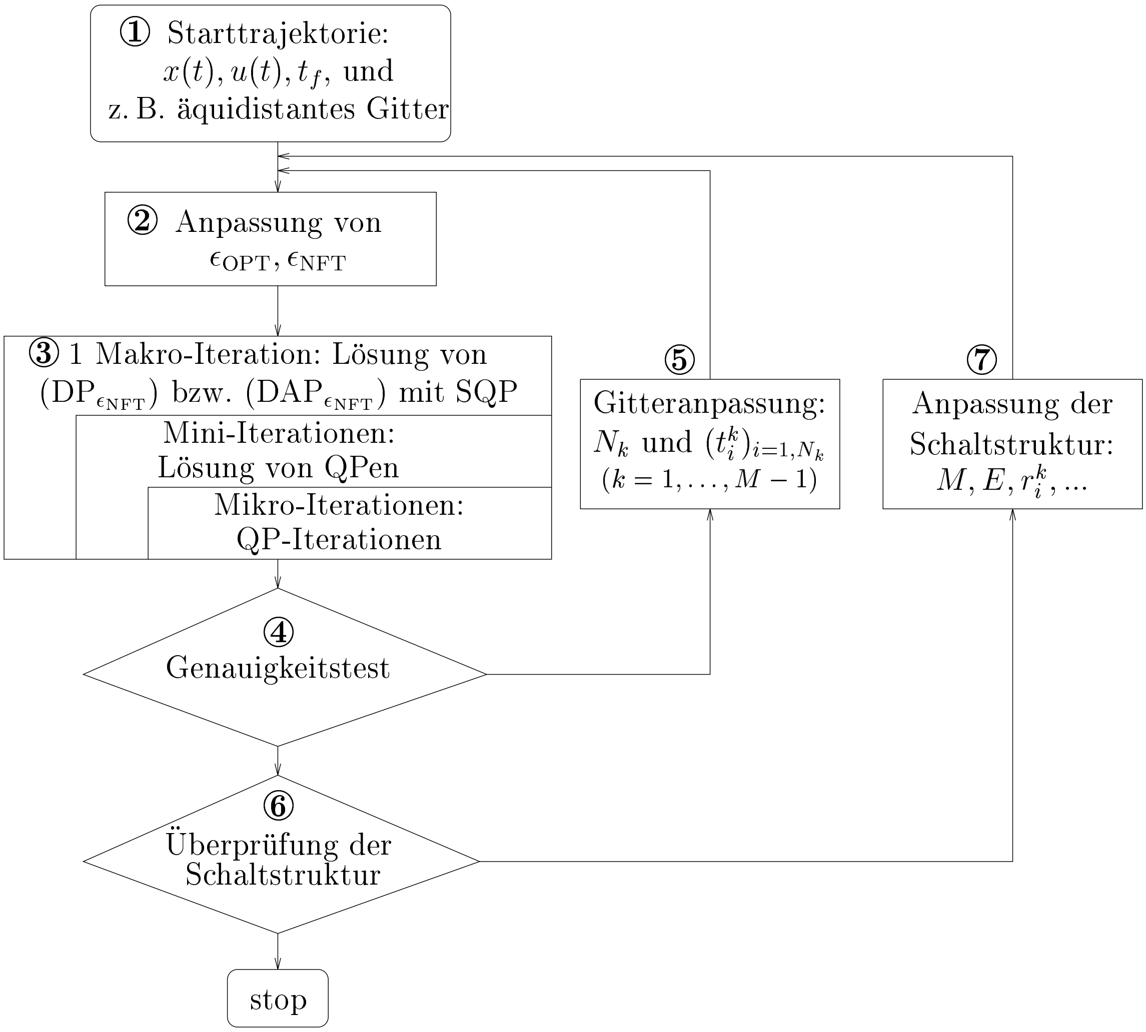

Abbildung 7.3: Schematischer Ablauf beim direkten Kollokationsverfahren DIRCOL.

Die Durchführung des direkten Kollokationsverfahrens kann gemäß den in Abbildung 7.3 dargestellten Schritten erfolgen. Sie beruht auf der Lösung einer Folge verwandter, endlich-dimensionaler nichtlinearer Optimierungsprobleme, deren Dimension mit der Verfeinerung der Lösung durch Anpassung der Lage der Gitterpunkte mit wachsendem N zunimmt:

(1) *Anfangsschätzwerte* für die Zustandsvariablen, die Steuervariablen und die ggf. freie Endzeit sind anzugeben. Meist ist eine lineare Interpolation vorgeschriebener Anfangs- und Endwerte der Zustandsvariablen und eine auf null geschätzte Steuerung ausreichend, um bereits bei einem groben äquidistanten Anfangsgitter (z. B. $N = 10$) hinreichendes Konvergenzverhalten des SQP-Verfahrens zu einer ersten Approximation zu erzeugen.

(2) Die *Toleranzen* ϵ_{OPT} und ϵ_{NFT} werden im Laufe der Verfeinerung der Lösung durch die Anpassung des Gitters sukzessive erniedrigt (von 10^{-3} auf maximal ϵ_{MACH}).

(3) In jeder *Makro-Iteration* wird ein nichtlineares Optimierungsproblem durch Aufruf des SQP-Verfahrens NPSOL [51] gelöst. Dazu muß in jeder Iteration des SQP-Verfahrens ein quadratisches Problem gelöst werden (*Mini-Iteration*). Die Lösung des QPs geschieht wiederum iterativ (*Mikro-Iterationen*).

(4) Ein *Genauigkeitstest*, bestehend aus der Überprüfung

- des maximalen Defektes $\max |f_j(x_{\text{app}}(t), u_{\text{app}}(t), t) - \dot{x}_{j,\text{app}}(t)|$ bei $t = t_{i+k/4}$, $k = 1, 2, 3$, $j = 1, \dots, n$, $i = 1, \dots, N-1$,
- des maximalen Fehlers in den Beschränkungen an den Testpunkten $t = t_{i+k/4}$, $k = 0, 1, 2, 3, 4$ und
- des geschätzten Optimalitätsfehlers im Wert des Zielfunktionals (Kapitel 5.4),

dient zur Beurteilung der Güte der Approximation der Lösung des Optimalsteuerungsproblems aus der berechneten Näherung.

(5) Die *Gitteranpassung* wird anhand der lokalen, intervallweisen Fehlerschätzer

- für die Differentialgleichungen: des lokalen Defekts $\max_{j=1,\dots,n}\{|f_j(x_{\text{app}}(t), u_{\text{app}}(t), t) - \dot{x}_{j,\text{app}}(t)|,\ t = t_{i+k/4},\ k = 1, 2, 3,$ und des lokalen Kollokationsfehlers (Kapitel 6.2.2),
- für die Beschränkungen: $\max_j |g_j(x, u, t)_-|,\quad t = t_{i+k/4},\ k = 0, 1, 2, 3, 4$ ($g_- = g$, falls $g < 0$, und 0 sonst)
- und des lokalen Optimalitätsfehlers (Kapitel 5.4)

durchgeführt.

(6) Die *Überprüfung der Schaltstruktur* und

(7) die *Anpassung der Schaltstruktur* ist nur dann erforderlich, falls die Berücksichtigung der Schaltstruktur in der Diskretisierung explizit gewünscht wird. Dies erfordert einen zusätzlichen Aufwand vom Benutzer, ermöglicht aber dafür eine verbesserte Approximation der Lösung. Dazu ist die Neuformulierung eines Optimalsteuerungsproblems, das z. B. ursprünglich vom Typ (P) war, als mehrstufiges Problem vom Typ (AP) (Kapitel 1) mit Schaltpunkten notwendig. Beispiele für dieses Vorgehen werden in den Kapiteln 8.3 (reichweitenmaximaler Flug eines Hängegleiters) und 8.4.3 (zeitoptimale Robotersteuerung) angegeben.

Bemerkung 7.1 Bei Anwendung einer *beweglichen Gitteranpassung* nach Kapitel 6.2 bzw. 6.3 verändert sich durch die Einführung weiterer Beschränkungen mit zugehörigen Multiplikatoren die Eigenschaft der Lagrangefunktion (Diskretisierung des erweiterten Funktionals) sowie die Interpretation der Multiplikatoren als Diskretisierungen der adjungierten Variablen. Eine bewegliche Gitteranpassung erfolgt daher nur optional in den ersten ein oder zwei Makroiterationen zur Bestimmung eines geeigneten, nichtäquidistanten Anfangsgitters. Die letzten Makroiterationen zur Verfeinerung der Lösung werden jeweils auf einem festen Gitter durchgeführt.