Inhaltsübersicht für heute:

SQP-Verfahren (Sequentielle quadratische Opt.-Verfahren)

Anwendungsbeispiel: Optimalsteuerung

Inhaltsübersicht für heute:

SQP-Verfahren (Sequentielle quadratische Opt.-Verfahren)

Anwendungsbeispiel: Optimalsteuerung

SQP-Verfahren (Sequential Quadratic Programming)

Derzeit der erfolgreichste Ansatz für nichtlineare restringierte Optimierung.

Idee: Finde den nächsten Punkt durch Lösung eines QP-Modells, das aus den linearisierten Nebenbedingungen und einem quadratischen Modell der Lagrange-Funktion als Zielfunktion aufgebaut wird.

SQP-Verfahren (Sequential Quadratic Programming)

Derzeit der erfolgreichste Ansatz für nichtlineare restringierte Optimierung.

Idee: Finde den nächsten Punkt durch Lösung eines QP-Modells, das aus den linearisierten Nebenbedingungen und einem quadratischen Modell der Lagrange-Funktion als Zielfunktion aufgebaut wird.

In 2 Schritten:

1. Lokale Konvergenz:
Das QP-Modell leitet sich aus dem Newton-Verfahren zur Bestimmung eines stationären Punktes der Lagrange-Funktion (= KKT-Bed.) her
$\rightarrow$ Newton führt zu lokal quadratischer Konvergenz von SQP-Verf.

SQP-Verfahren (Sequential Quadratic Programming)

Derzeit der erfolgreichste Ansatz für nichtlineare restringierte Optimierung.

Idee: Finde den nächsten Punkt durch Lösung eines QP-Modells, das aus den linearisierten Nebenbedingungen und einem quadratischen Modell der Lagrange-Funktion als Zielfunktion aufgebaut wird.

In 2 Schritten:

1. Lokale Konvergenz:
Das QP-Modell leitet sich aus dem Newton-Verfahren zur Bestimmung eines stationären Punktes der Lagrange-Funktion (= KKT-Bed.) her
$\rightarrow$ Newton führt zu lokal quadratischer Konvergenz von SQP-Verf.

2. Globalisierungs-Ansätze:
Fortschritt bezüglich Erfüllung der Nebenbedingungen und Verbesserung der Zielfunktion kann durch Merit-Funktion, Trust-Region-Ansatz oder Filter-Verfahren kontrolliert werden.

Herleitung des QPs aus KKT und Newton

Mit Gleichungsbedingungen: $\min f(x)$ s.t. $h(x) = 0$. $[h : \mathbb{R}^n \to \mathbb{R}^{\mathcal{E}}]$
Löse die KKT-Bed. für $\mathcal{L}(x, \mu) = f(x) + \mu^T h(x)$ mit Newton:

Herleitung des QPs aus KKT und Newton

Mit Gleichungsbedingungen: $\min f(x)$ s.t. $h(x) = 0$. $[h : \mathbb{R}^n \to \mathbb{R}^{\mathcal{E}}]$
Löse die KKT-Bed. für $\mathcal{L}(x, \mu) = f(x) + \mu^T h(x)$ mit Newton:

KKT: $F(x, \mu) := \begin{bmatrix} \nabla_x \mathcal{L} \\ \nabla_\mu \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla f(x) + J_h(x)^T \mu \\ h(x) \end{bmatrix} = 0$

Herleitung des QPs aus KKT und Newton

Mit Gleichungsbedingungen: $\min f(x)$ s.t. $h(x) = 0.$ $[h : \mathbb{R}^n \to \mathbb{R}^{\mathcal{E}}]$

Löse die KKT-Bed. für $\mathcal{L}(x, \mu) = f(x) + \mu^T h(x)$ mit Newton:

KKT: $F(x, \mu) := \begin{bmatrix} \nabla_x \mathcal{L} \\ \nabla_\mu \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla f(x) + J_h(x)^T \mu \\ h(x) \end{bmatrix} = 0$

Newton: $\begin{bmatrix} x^{(k+1)} \\ \mu^{(k+1)} \end{bmatrix} := \begin{bmatrix} x^{(k)} \\ \mu^{(k)} \end{bmatrix} + \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix}$ mit $F(x^{(k)}, \mu^{(k)}) + J_F(x^{(k)}, \mu^{(k)}) \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix} = 0$

Herleitung des QPs aus KKT und Newton

Mit Gleichungsbedingungen: $\min f(x)$ s.t. $h(x) = 0$. $[h : \mathbb{R}^n \to \mathbb{R}^{\mathcal{E}}]$

Löse die KKT-Bed. für $\mathcal{L}(x,\mu) = f(x) + \mu^T h(x)$ mit Newton:

KKT: $$F(x,\mu) := \begin{bmatrix} \nabla_x \mathcal{L} \\ \nabla_\mu \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla f(x) + J_h(x)^T \mu \\ h(x) \end{bmatrix} = 0$$

Newton: $$\begin{bmatrix} x^{(k+1)} \\ \mu^{(k+1)} \end{bmatrix} := \begin{bmatrix} x^{(k)} \\ \mu^{(k)} \end{bmatrix} + \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix} \text{ mit } F(x^{(k)}, \mu^{(k)}) + J_F(x^{(k)}, \mu^{(k)}) \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix} = 0$$

$$J_F(x,\mu) = \begin{bmatrix} \nabla_{xx}\mathcal{L} & \nabla_{x\mu}\mathcal{L} \\ \nabla_{\mu x}\mathcal{L} & \nabla_{\mu\mu}\mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla_{xx}\mathcal{L}(x,\mu) & J_h(x)^T \\ J_h(x) & 0 \end{bmatrix} =: \begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix}$$

Herleitung des QPs aus KKT und Newton

Mit Gleichungsbedingungen: $\min f(x)$ s.t. $h(x) = 0$. $[h : \mathbb{R}^n \to \mathbb{R}^{\mathcal{E}}]$
Löse die KKT-Bed. für $\mathcal{L}(x,\mu) = f(x) + \mu^T h(x)$ mit Newton:

KKT: $$F(x,\mu) := \begin{bmatrix} \nabla_x \mathcal{L} \\ \nabla_\mu \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla f(x) + J_h(x)^T \mu \\ h(x) \end{bmatrix} = 0$$

Newton: $\begin{bmatrix} x^{(k+1)} \\ \mu^{(k+1)} \end{bmatrix} := \begin{bmatrix} x^{(k)} \\ \mu^{(k)} \end{bmatrix} + \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix}$ mit $F(x^{(k)}, \mu^{(k)}) + J_F(x^{(k)}, \mu^{(k)}) \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix} = 0$

$$J_F(x,\mu) = \begin{bmatrix} \nabla_{xx}\mathcal{L} & \nabla_{x\mu}\mathcal{L} \\ \nabla_{\mu x}\mathcal{L} & \nabla_{\mu\mu}\mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla_{xx}\mathcal{L}(x,\mu) & J_h(x)^T \\ J_h(x) & 0 \end{bmatrix} =: \begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix}$$

Iteration k: Löse $$\begin{bmatrix} Q_k & A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix} = \begin{bmatrix} -\nabla f_k - A_k^T \mu^{(k)} \\ -h^{(k)} \end{bmatrix}$$

mit $Q_k := \nabla_{xx}\mathcal{L}(x^{(k)}, \mu^{(k)})$, $A_k := J_h(x^{(k)})$, $h^{(k)} := [h_1(x^{(k)}), \dots, h_{|\mathcal{E}|}(x^{(k)})]^T$.

Herleitung des QPs aus KKT und Newton

Mit Gleichungsbedingungen: $\min f(x)$ s.t. $h(x) = 0.$ $[h : \mathbb{R}^n \to \mathbb{R}^{\mathcal{E}}]$

Löse die KKT-Bed. für $\mathcal{L}(x,\mu) = f(x) + \mu^T h(x)$ mit Newton:

KKT: $$F(x,\mu) := \begin{bmatrix} \nabla_x \mathcal{L} \\ \nabla_\mu \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla f(x) + J_h(x)^T \mu \\ h(x) \end{bmatrix} = 0$$

Newton: $$\begin{bmatrix} x^{(k+1)} \\ \mu^{(k+1)} \end{bmatrix} := \begin{bmatrix} x^{(k)} \\ \mu^{(k)} \end{bmatrix} + \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix} \text{ mit } F(x^{(k)}, \mu^{(k)}) + J_F(x^{(k)}, \mu^{(k)}) \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix} = 0$$

$$J_F(x,\mu) = \begin{bmatrix} \nabla_{xx}\mathcal{L} & \nabla_{x\mu}\mathcal{L} \\ \nabla_{\mu x}\mathcal{L} & \nabla_{\mu\mu}\mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla_{xx}\mathcal{L}(x,\mu) & J_h(x)^T \\ J_h(x) & 0 \end{bmatrix} =: \begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix}$$

Iteration k: Löse $$\begin{bmatrix} Q_k & A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix} = \begin{bmatrix} -\nabla f_k - A_k^T \mu^{(k)} \\ -h^{(k)} \end{bmatrix}$$

mit $Q_k := \nabla_{xx}\mathcal{L}(x^{(k)}, \mu^{(k)})$, $A_k := J_h(x^{(k)})$, $h^{(k)} := [h_1(x^{(k)}), \ldots, h_{|\mathcal{E}|}(x^{(k)})]^T$.

Vergleiche dazu das folgende gleichungsbeschränkte QP:

$$\begin{aligned} \min \quad & \tfrac{1}{2} d_x Q_k d_x + \nabla f_k^T d_x \\ \text{s.t.} \quad & A_k d_x = -h^{(k)} \end{aligned}$$

$[\text{Zeile } i\text{: } h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0]$

Herleitung des QPs aus KKT und Newton

Mit Gleichungsbedingungen: $\min f(x)$ s.t. $h(x) = 0$. $[h : \mathbb{R}^n \to \mathbb{R}^{\mathcal{E}}]$

Löse die KKT-Bed. für $\mathcal{L}(x,\mu) = f(x) + \mu^T h(x)$ mit Newton:

KKT: $$F(x,\mu) := \begin{bmatrix} \nabla_x \mathcal{L} \\ \nabla_\mu \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla f(x) + J_h(x)^T \mu \\ h(x) \end{bmatrix} = 0$$

Newton: $\begin{bmatrix} x^{(k+1)} \\ \mu^{(k+1)} \end{bmatrix} := \begin{bmatrix} x^{(k)} \\ \mu^{(k)} \end{bmatrix} + \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix}$ mit $F(x^{(k)},\mu^{(k)}) + J_F(x^{(k)},\mu^{(k)}) \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix} = 0$

$$J_F(x,\mu) = \begin{bmatrix} \nabla_{xx}\mathcal{L} & \nabla_{x\mu}\mathcal{L} \\ \nabla_{\mu x}\mathcal{L} & \nabla_{\mu\mu}\mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla_{xx}\mathcal{L}(x,\mu) & J_h(x)^T \\ J_h(x) & 0 \end{bmatrix} =: \begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix}$$

Iteration k: Löse $$\begin{bmatrix} Q_k & A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix} = \begin{bmatrix} -\nabla f_k - A_k^T \mu^{(k)} \\ -h^{(k)} \end{bmatrix}$$

mit $Q_k := \nabla_{xx}\mathcal{L}(x^{(k)},\mu^{(k)})$, $A_k := J_h(x^{(k)})$, $h^{(k)} := [h_1(x^{(k)}),\dots,h_{|\mathcal{E}|}(x^{(k)})]^T$.

Vergleiche dazu das folgende gleichungsbeschränkte QP:

$$\begin{aligned} \min \quad & \tfrac{1}{2} d_x Q_k d_x + \nabla f_k^T d_x \\ \text{s.t.} \quad & A_k d_x = -h^{(k)} \end{aligned}$$ $[\text{Zeile } i\text{: } h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0]$

Dafür bestimmt sich eine Optimallösung mit Multiplikatoren y durch

$\begin{bmatrix} Q_k & -A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_x \\ y \end{bmatrix} = \begin{bmatrix} -\nabla f_k \\ -h^{(k)} \end{bmatrix}$, setze $y = -(\mu^{(k)} + d_\mu) \Leftrightarrow$ Newton-System

Das QP des SQP-Verfahrens

$$(P)\quad \begin{array}{ll} \min & f(x) \\ \text{s.t.} & h_i(x) = 0 \quad i \in \mathcal{E} \\ & g_i(x) \leq 0 \quad i \in \mathcal{I} \end{array}, \qquad \mathcal{L}(x,\mu,\lambda) := f(x) + \sum_{i\in\mathcal{E}} \mu_i h_i(x) + \sum_{i\in\mathcal{I}} \lambda_i g_i(x),$$

Das QP des SQP-Verfahrens

$$(P)\quad \begin{array}{ll} \min & f(x) \\ \text{s.t.} & h_i(x) = 0 \quad i \in \mathcal{E} \\ & g_i(x) \le 0 \quad i \in \mathcal{I} \end{array}, \quad \mathcal{L}(x,\mu,\lambda) := f(x) + \sum_{i\in\mathcal{E}} \mu_i h_i(x) + \sum_{i\in\mathcal{I}} \lambda_i g_i(x),$$

bestimme $x^{(k+1)} := x^{(k)} + d_x^{(k)}$ für geg. $\mu^{(k)}, \lambda^{(k)}$, $Q_k := \nabla_{xx}\mathcal{L}(x^{(k)}, \mu^{(k)}, \lambda^{(k)})$ aus

$$(QP_k)\quad \begin{array}{ll} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x \\ \text{s.t.} & h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 \quad i \in \mathcal{E} \\ & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \le 0 \quad i \in \mathcal{I} \end{array}$$

Das QP des SQP-Verfahrens

$$(P)\quad \begin{array}{lll} \min & f(x) & \\ \text{s.t.} & h_i(x) = 0 & i \in \mathcal{E} \\ & g_i(x) \leq 0 & i \in \mathcal{I} \end{array}, \qquad \mathcal{L}(x,\mu,\lambda) := f(x) + \sum_{i\in\mathcal{E}} \mu_i h_i(x) + \sum_{i\in\mathcal{I}} \lambda_i g_i(x),$$

bestimme $x^{(k+1)} := x^{(k)} + d_x^{(k)}$ für geg. $\mu^{(k)}, \lambda^{(k)}$, $Q_k := \nabla_{xx}\mathcal{L}(x^{(k)}, \mu^{(k)}, \lambda^{(k)})$ aus

$$(QP_k)\quad \begin{array}{lll} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x & \\ \text{s.t.} & h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 & i \in \mathcal{E} \\ & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq 0 & i \in \mathcal{I} \end{array} \quad \rightarrow \quad \begin{array}{lll} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x & \\ \text{s.t.} & A_k^h d_x = -h^{(k)} & [y_h \in \mathbb{R}^{\mathcal{E}}] \\ & A_k^g d_x \leq -g^{(k)} & [y_g \in \mathbb{R}_-^{\mathcal{I}}] \end{array}$$

wobei $A_k^h = J_h(x^{(k)})$, $A_k^g = J_g(x^{(k)})$, setze $\mu^{(k+1)} := -y_h^{(k)}$, $\lambda^{(k+1)} := -y_g^{(k)}$.

Das QP des SQP-Verfahrens

$$(P)\quad \begin{array}{ll} \min & f(x) \\ \text{s.t.} & h_i(x) = 0 \quad i \in \mathcal{E} \\ & g_i(x) \le 0 \quad i \in \mathcal{I} \end{array}, \qquad \mathcal{L}(x,\mu,\lambda) := f(x) + \sum_{i\in\mathcal{E}} \mu_i h_i(x) + \sum_{i\in\mathcal{I}} \lambda_i g_i(x),$$

bestimme $x^{(k+1)} := x^{(k)} + d_x^{(k)}$ für geg. $\mu^{(k)}, \lambda^{(k)}$, $Q_k := \nabla_{xx}\mathcal{L}(x^{(k)}, \mu^{(k)}, \lambda^{(k)})$ aus

$$(QP_k)\quad \begin{array}{ll} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x \\ \text{s.t.} & h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 \quad i \in \mathcal{E} \\ & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \le 0 \quad i \in \mathcal{I} \end{array} \;\rightarrow\; \begin{array}{ll} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x \\ \text{s.t.} & A_k^h d_x = -h^{(k)} \quad [y_h \in \mathbb{R}^{\mathcal{E}}] \\ & A_k^g d_x \le -g^{(k)} \quad [y_g \in \mathbb{R}_-^{\mathcal{I}}] \end{array}$$

wobei $A_k^h = J_h(x^{(k)})$, $A_k^g = J_g(x^{(k)})$, setze $\mu^{(k+1)} := -y_h^{(k)}$, $\lambda^{(k+1)} := -y_g^{(k)}$.

Bemerkungen:

- Ist $(x^{(k)}, \mu^{(k)}, \lambda^{(k)})$ nahe genug an (x^*, μ^*, λ^*), das die hinr. Opt.-Bed. und strenge Kompl. erfüllt, sind die aktiven Mengen gleich und Newton garantiert lokal quadratische Konvergenz.

Das QP des SQP-Verfahrens

$$(P)\quad \begin{array}{ll} \min & f(x) \\ \text{s.t.} & h_i(x) = 0 \quad i \in \mathcal{E} \\ & g_i(x) \le 0 \quad i \in \mathcal{I} \end{array}, \qquad \mathcal{L}(x,\mu,\lambda) := f(x) + \sum_{i\in\mathcal{E}} \mu_i h_i(x) + \sum_{i\in\mathcal{I}} \lambda_i g_i(x),$$

bestimme $x^{(k+1)} := x^{(k)} + d_x^{(k)}$ für geg. $\mu^{(k)}, \lambda^{(k)}$, $Q_k := \nabla_{xx}\mathcal{L}(x^{(k)}, \mu^{(k)}, \lambda^{(k)})$ aus

$$(QP_k)\quad \begin{array}{lll} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x & \\ \text{s.t.} & h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 & i \in \mathcal{E} \\ & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \le 0 & i \in \mathcal{I} \end{array} \;\rightarrow\; \begin{array}{lll} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x & \\ \text{s.t.} & A_k^h d_x = -h^{(k)} & [y_h \in \mathbb{R}^{\mathcal{E}}] \\ & A_k^g d_x \le -g^{(k)} & [y_g \in \mathbb{R}_-^{\mathcal{I}}] \end{array}$$

wobei $A_k^h = J_h(x^{(k)})$, $A_k^g = J_g(x^{(k)})$, setze $\mu^{(k+1)} := -y_h^{(k)}$, $\lambda^{(k+1)} := -y_g^{(k)}$.

Bemerkungen:

- Ist $(x^{(k)}, \mu^{(k)}, \lambda^{(k)})$ nahe genug an (x^*, μ^*, λ^*), das die hinr. Opt.-Bed. und strenge Kompl. erfüllt, sind die aktiven Mengen gleich und Newton garantiert lokal quadratische Konvergenz.
- Q_k enthält Krümmung von f und den aktiven Nebenbedingungen,

  $Q_k = \nabla^2 f(x^{(k)}) + \sum_{i\in\mathcal{E}} \mu_i^{(k)} \nabla^2 h_i(x^{(k)}) + \sum_{i\in\mathcal{I}} \lambda_i^{(k)} \nabla^2 g_i(x^{(k)})$,

  $\rightarrow$ kurze Schritte in Richtungen mit starker Änderung wichtiger Funktionen.

Das QP des SQP-Verfahrens

$$(P)\quad \begin{array}{ll} \min & f(x) \\ \text{s.t.} & h_i(x) = 0 \quad i \in \mathcal{E} \\ & g_i(x) \le 0 \quad i \in \mathcal{I} \end{array}, \quad \mathcal{L}(x,\mu,\lambda) := f(x) + \sum_{i\in\mathcal{E}} \mu_i h_i(x) + \sum_{i\in\mathcal{I}} \lambda_i g_i(x),$$

bestimme $x^{(k+1)} := x^{(k)} + d_x^{(k)}$ für geg. $\mu^{(k)}, \lambda^{(k)}$, $Q_k := \nabla_{xx}\mathcal{L}(x^{(k)}, \mu^{(k)}, \lambda^{(k)})$ aus

$$(QP_k)\quad \begin{array}{lll} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x & \\ \text{s.t.} & h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 & i \in \mathcal{E} \\ & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \le 0 & i \in \mathcal{I} \end{array} \rightarrow \begin{array}{lll} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x & \\ \text{s.t.} & A_k^h d_x = -h^{(k)} & [y_h \in \mathbb{R}^{\mathcal{E}}] \\ & A_k^g d_x \le -g^{(k)} & [y_g \in \mathbb{R}_-^{\mathcal{I}}] \end{array}$$

wobei $A_k^h = J_h(x^{(k)})$, $A_k^g = J_g(x^{(k)})$, setze $\mu^{(k+1)} := -y_h^{(k)}$, $\lambda^{(k+1)} := -y_g^{(k)}$.

Bemerkungen:

- Ist $(x^{(k)}, \mu^{(k)}, \lambda^{(k)})$ nahe genug an (x^*, μ^*, λ^*), das die hinr. Opt.-Bed. und strenge Kompl. erfüllt, sind die aktiven Mengen gleich und Newton garantiert lokal quadratische Konvergenz.
- Q_k enthält Krümmung von f und den aktiven Nebenbedingungen,
  $Q_k = \nabla^2 f(x^{(k)}) + \sum_{i\in\mathcal{E}} \mu_i^{(k)} \nabla^2 h_i(x^{(k)}) + \sum_{i\in\mathcal{I}} \lambda_i^{(k)} \nabla^2 g_i(x^{(k)})$,
  $\rightarrow$ kurze Schritte in Richtungen mit starker Änderung wichtiger Funktionen.
- Die Hessematrizen in Q_k sind durch BFGS, etc. approximierbar.

Globalisierung mit Merit-Funktionen

Verwende die Lösung $d_x^{(k)}$ von (QP_k) als Suchrichtung und bestimme $x^{(k+1)} = x^{(k)} + \alpha_k d_x^{(k)}$ durch Line-Search bezüglich einer Merit-Funktion, die Verbesserung in Zielfunktion und Zulässigkeit gemeinsam bewertet.

Globalisierung mit Merit-Funktionen

Verwende die Lösung $d_x^{(k)}$ von (QP_k) als Suchrichtung und bestimme $x^{(k+1)} = x^{(k)} + \alpha_k d_x^{(k)}$ durch Line-Search bezüglich einer Merit-Funktion, die Verbesserung in Zielfunktion und Zulässigkeit gemeinsam bewertet.

---

l_1-**Merit-Funktion:** Für geg. Strafparameter $\gamma > 0$ verwende

$$f_\gamma(x) := f(x) + \gamma\Big[\sum_{i\in\mathcal{E}} |h_i(x)| + \sum_{i\in\mathcal{I}} \max\{0, g_i(x)\}\Big]$$

- f_γ ist nicht diffbar, aber Richtungsabl. existiert und genügt für Line-Search.
- $d_x^{(k)}$ ist Abstiegsrichtung für f_γ für $\gamma > \max(\{|\mu_i| : i \in \mathcal{E}\} \cup \{\lambda_i : i \in \mathcal{I}\})$.

Globalisierung mit Merit-Funktionen

Verwende die Lösung $d_x^{(k)}$ von (QP_k) als Suchrichtung und bestimme $x^{(k+1)} = x^{(k)} + \alpha_k d_x^{(k)}$ durch Line-Search bezüglich einer Merit-Funktion, die Verbesserung in Zielfunktion und Zulässigkeit gemeinsam bewertet.

---

l_1-**Merit-Funktion:** Für geg. Strafparameter $\gamma > 0$ verwende

$$f_\gamma(x) := f(x) + \gamma\Big[\sum_{i\in\mathcal{E}} |h_i(x)| + \sum_{i\in\mathcal{I}} \max\{0, g_i(x)\}\Big]$$

- f_γ ist nicht diffbar, aber Richtungsabl. existiert und genügt für Line-Search.
- $d_x^{(k)}$ ist Abstiegsrichtung für f_γ für $\gamma > \max(\{|\mu_i| : i \in \mathcal{E}\} \cup \{\lambda_i : i \in \mathcal{I}\})$.

Augmentierte-Lagrange-Merit-Funktion von Fletcher: Für geg. $\gamma > 0$,

$$\mathcal{L}_\gamma(x,\mu,\lambda) := f(x) + \sum_{i\in\mathcal{E}} \mu_i h_i(x) + \sum_{i\in\mathcal{I}} \lambda_i g_i(x) + \gamma\Big[\sum_{i\in\mathcal{E}} h_i(x)^2 + \sum_{i\in\mathcal{I}} \max\{0, g_i(x)\}^2\Big]$$

- differenzierbar und für γ groß genug ist $d_x^{(k)}$ Abstiegsrichtung
- Die Wahl von γ hat starken Einfluss auf das Konvergenzverhalten …

Globalisierung mit Merit-Funktionen

Verwende die Lösung $d_x^{(k)}$ von (QP_k) als Suchrichtung und bestimme $x^{(k+1)} = x^{(k)} + \alpha_k d_x^{(k)}$ durch Line-Search bezüglich einer Merit-Funktion, die Verbesserung in Zielfunktion und Zulässigkeit gemeinsam bewertet.

---

l_1-**Merit-Funktion:** Für geg. Strafparameter $\gamma > 0$ verwende

$$f_\gamma(x) := f(x) + \gamma\Big[\sum_{i\in\mathcal{E}} |h_i(x)| + \sum_{i\in\mathcal{I}} \max\{0, g_i(x)\}\Big]$$

- f_γ ist nicht diffbar, aber Richtungsabl. existiert und genügt für Line-Search.
- $d_x^{(k)}$ ist Abstiegsrichtung für f_γ für $\gamma > \max(\{|\mu_i| : i \in \mathcal{E}\} \cup \{\lambda_i : i \in \mathcal{I}\})$.

Augmentierte-Lagrange-Merit-Funktion von Fletcher: Für geg. $\gamma > 0$,

$$\mathcal{L}_\gamma(x,\mu,\lambda) := f(x) + \sum_{i\in\mathcal{E}} \mu_i h_i(x) + \sum_{i\in\mathcal{I}} \lambda_i g_i(x) + \gamma\Big[\sum_{i\in\mathcal{E}} h_i(x)^2 + \sum_{i\in\mathcal{I}} \max\{0, g_i(x)\}^2\Big]$$

- differenzierbar und für γ groß genug ist $d_x^{(k)}$ Abstiegsrichtung
- Die Wahl von γ hat starken Einfluss auf das Konvergenzverhalten ...

Generelles Problem: Maratos-Effekt

Line-Search bzgl. Merit-Funktion verbietet oft Schrittweite 1 im Gebiet der lokalen quadratischen Konvergenz $\to$ schlechte lokale Konvergenz.

Es gibt heuristische Gegenstrategien ...

Globalisierung mit Trust-Region-Ansätzen

Klassischer Ansatz:

- Fordere zusätzlich $\|d_x\| \leq \Delta$ oder, damit über lineare Nebenbed. darstellbar, $\|d_x\|_\infty := \max\{|[d_x]_i| : i = 1, \ldots, n\} \leq \Delta$.
- Schwierigkeit: Ist $x^{(k)}$ unzulässig, kann durch $\|d_x\|_\infty \leq \Delta$ auch (QP_k) unzulässig werden. Lösungsvorschläge sind z.B. erst ein Zulässigkeitsproblem zu lösen und dann Δ zu wählen, etc.

Globalisierung mit Trust-Region-Ansätzen

Klassischer Ansatz:

- Fordere zusätzlich $\|d_x\| \le \Delta$ oder, damit über lineare Nebenbed. darstellbar, $\|d_x\|_\infty := \max\{|[d_x]_i| : i = 1, \dots, n\} \le \Delta$.
- Schwierigkeit: Ist $x^{(k)}$ unzulässig, kann durch $\|d_x\|_\infty \le \Delta$ auch (QP_k) unzulässig werden. Lösungsvorschläge sind z.B. erst ein Zulässigkeitsproblem zu lösen und dann Δ zu wählen, etc.

Kombinierter Strafansatz: (im open-source-Paket IPOPT implementiert)
Idee: x ist zulässig, wenn $\max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\}) = 0$.

Globalisierung mit Trust-Region-Ansätzen

Klassischer Ansatz:

- Fordere zusätzlich $\|d_x\| \leq \Delta$ oder, damit über lineare Nebenbed. darstellbar, $\|d_x\|_\infty := \max\{|[d_x]_i| : i = 1, \ldots, n\} \leq \Delta$.
- Schwierigkeit: Ist $x^{(k)}$ unzulässig, kann durch $\|d_x\|_\infty \leq \Delta$ auch (QP_k) unzulässig werden. Lösungsvorschläge sind z.B. erst ein Zulässigkeitsproblem zu lösen und dann Δ zu wählen, etc.

Kombinierter Strafansatz: (im open-source-Paket IPOPT implementiert)
Idee: x ist zulässig, wenn $\max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\}) = 0$.
$\rightarrow$ Löse als Trust-Region-Unterproblem für Strafparameter $\gamma > 0$

$$
\begin{array}{lll}
\min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x + \gamma s & \\
\text{s.t.} & -s \leq h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x \leq s & i \in \mathcal{E} \\
& g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq s & i \in \mathcal{I} \\
& -\Delta \leq [d_x]_i \leq \Delta \ (i = 1, \ldots, n), s \geq 0 &
\end{array}
$$

Globalisierung mit Trust-Region-Ansätzen

Klassischer Ansatz:

- Fordere zusätzlich $\|d_x\| \leq \Delta$ oder, damit über lineare Nebenbed. darstellbar, $\|d_x\|_\infty := \max\{|[d_x]_i| : i = 1, \dots, n\} \leq \Delta$.
- Schwierigkeit: Ist $x^{(k)}$ unzulässig, kann durch $\|d_x\|_\infty \leq \Delta$ auch (QP_k) unzulässig werden. Lösungsvorschläge sind z.B. erst ein Zulässigkeitsproblem zu lösen und dann Δ zu wählen, etc.

Kombinierter Strafansatz: (im open-source-Paket IPOPT implementiert)
Idee: x ist zulässig, wenn $\max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\}) = 0$.
$\rightarrow$ Löse als Trust-Region-Unterproblem für Strafparameter $\gamma > 0$

$$\begin{array}{lll} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x + \gamma s & \\ \text{s.t.} & -s \leq h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x \leq s & i \in \mathcal{E} \\ & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq s & i \in \mathcal{I} \\ & -\Delta \leq [d_x]_i \leq \Delta \ (i = 1, \dots, n), s \geq 0 & \end{array}$$

Der Trust-Region Algorithmus vergleicht Fortschritt der Funktion

$$f_\gamma(d) := f(x^{(k)}+d) + \gamma \max(\{|h_i(x^{(k)}+d)| : i \in \mathcal{E}\} \cup \{g_i(x^{(k)}+d) : i \in \mathcal{I}\} \cup \{0\})$$

zum Fortschritt im Modell und passt Δ entsprechend an. Strafparameter γ wird nur bei Verletzung der Nebenbed. vergrößert, wenn der Fortschritt Richtung Zulässigkeit im Verhältnis zur Schrittlänge zu klein ist.

Globalisierung mit einem Filter-Ansatz

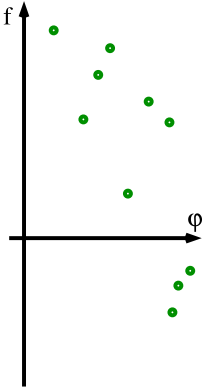
Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

Globalisierung mit einem Filter-Ansatz

Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)$ ...Verletzungsmaß, z.B. $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$.

Betrachte $(f_k, \varphi_k) := \left(f(x^{(k)}), \varphi(x^{(k)})\right)$ als Punkt in $\mathbb{R}^2$.

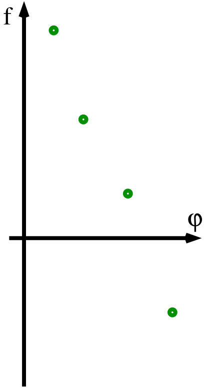


Globalisierung mit einem Filter-Ansatz

Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)\ldots$Verletzungsmaß, z.B. $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$.
Betrachte $(f_k, \varphi_k) := \big(f(x^{(k)}), \varphi(x^{(k)})\big)$ als Punkt in $\mathbb{R}^2$.
Ein Filter enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \big\{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \ldots, k\}\big\}.$$



Globalisierung mit einem Filter-Ansatz

Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)\ldots$Verletzungsmaß, z.B. $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$.

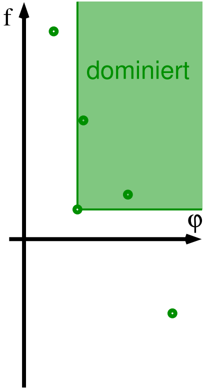
Betrachte $(f_k, \varphi_k) := \left(f(x^{(k)}), \varphi(x^{(k)})\right)$ als Punkt in $\mathbb{R}^2$.

Ein Filter enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \left\{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \ldots, k\}\right\}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ dominiert $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$, falls

$$\hat{f} \leq \bar{f} \text{ und } \hat{\varphi} \leq \bar{\varphi}.$$



Globalisierung mit einem Filter-Ansatz

Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)\ldots$Verletzungsmaß, z.B. $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$.

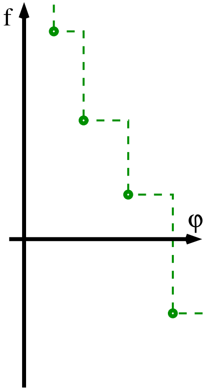
Betrachte $(f_k, \varphi_k) := \big(f(x^{(k)}), \varphi(x^{(k)})\big)$ als Punkt in $\mathbb{R}^2$.

Ein Filter enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \big\{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \ldots, k\}\big\}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ dominiert $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$, falls

$$\hat{f} \leq \bar{f} \text{ und } \hat{\varphi} \leq \bar{\varphi}.$$



Globalisierung mit einem Filter-Ansatz

Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)\ldots$Verletzungsmaß, z.B. $\varphi(x):=\max(\{|h_i(x)|: i\in\mathcal{E}\}\cup\{g_i(x): i\in\mathcal{I}\}\cup\{0\})$.

Betrachte $(f_k,\varphi_k) := \left(f(x^{(k)}), \varphi(x^{(k)})\right)$ als Punkt in $\mathbb{R}^2$.

Ein Filter enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \left\{(f_j,\varphi_j) : j \in J_k \subseteq \{1,\ldots,k\}\right\}.$$

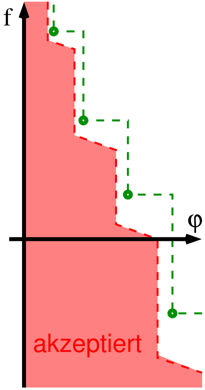
Ein $(\hat f,\hat\varphi) \in \mathbb{R}^2$ dominiert $(\bar f,\bar\varphi) \in \mathbb{R}^2$, falls

$$\hat f \le \bar f \text{ und } \hat\varphi \le \bar\varphi.$$

Ein $(\hat f,\hat\varphi) \in \mathbb{R}^2$ wird von $\mathcal{F}_k$ für gegebene $0 < \gamma_1 < \gamma_2 < 1$ akzeptiert, falls

$$\forall j \in J_k : (\hat\varphi \le \gamma_2\varphi_j) \text{ oder } (\hat f + \gamma_1\hat\varphi \le f_j)$$

[Hinreichende Verbesserung in Verletzung oder Wert]



Globalisierung mit einem Filter-Ansatz

Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)\dots$Verletzungsmaß, z.B. $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$.

Betrachte $(f_k, \varphi_k) := \big(f(x^{(k)}), \varphi(x^{(k)})\big)$ als Punkt in $\mathbb{R}^2$.

Ein Filter enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \big\{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \dots, k\}\big\}.$$

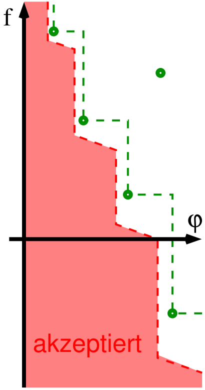
Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ dominiert $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$, falls

$$\hat{f} \le \bar{f} \text{ und } \hat{\varphi} \le \bar{\varphi}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ wird von $\mathcal{F}_k$ für gegebene $0 < \gamma_1 < \gamma_2 < 1$ akzeptiert, falls

$$\forall j \in J_k : (\hat{\varphi} \le \gamma_2 \varphi_j) \text{ oder } (\hat{f} + \gamma_1 \hat{\varphi} \le f_j)$$

[Hinreichende Verbesserung in Verletzung oder Wert]



Globalisierung mit einem Filter-Ansatz

Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)\ldots$Verletzungsmaß, z.B. $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$.

Betrachte $(f_k, \varphi_k) := \left(f(x^{(k)}), \varphi(x^{(k)})\right)$ als Punkt in $\mathbb{R}^2$.

Ein Filter enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \left\{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \ldots, k\}\right\}.$$

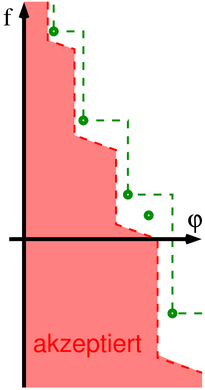
Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ dominiert $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$, falls

$$\hat{f} \le \bar{f} \text{ und } \hat{\varphi} \le \bar{\varphi}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ wird von $\mathcal{F}_k$ für gegebene $0 < \gamma_1 < \gamma_2 < 1$ akzeptiert, falls

$$\forall j \in J_k : (\hat{\varphi} \le \gamma_2 \varphi_j) \text{ oder } (\hat{f} + \gamma_1 \hat{\varphi} \le f_j)$$

[Hinreichende Verbesserung in Verletzung oder Wert]



Globalisierung mit einem Filter-Ansatz

Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)\ldots$Verletzungsmaß, z.B. $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$.

Betrachte $(f_k, \varphi_k) := \left(f(x^{(k)}), \varphi(x^{(k)})\right)$ als Punkt in $\mathbb{R}^2$.

Ein Filter enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \left\{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \ldots, k\}\right\}.$$

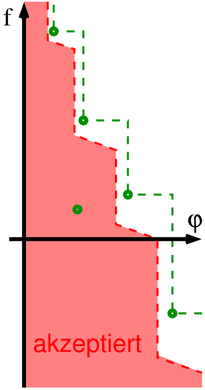
Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ dominiert $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$, falls

$$\hat{f} \leq \bar{f} \text{ und } \hat{\varphi} \leq \bar{\varphi}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ wird von $\mathcal{F}_k$ für gegebene $0 < \gamma_1 < \gamma_2 < 1$ akzeptiert, falls

$$\forall j \in J_k : (\hat{\varphi} \leq \gamma_2 \varphi_j) \text{ oder } (\hat{f} + \gamma_1 \hat{\varphi} \leq f_j)$$

[Hinreichende Verbesserung in Verletzung oder Wert]



Globalisierung mit einem Filter-Ansatz

Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)$ ... Verletzungsmaß, z.B. $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$.

Betrachte $(f_k, \varphi_k) := \left(f(x^{(k)}), \varphi(x^{(k)})\right)$ als Punkt in $\mathbb{R}^2$.

Ein Filter enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \dots, k\}\}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ dominiert $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$, falls

$$\hat{f} \le \bar{f} \text{ und } \hat{\varphi} \le \bar{\varphi}.$$

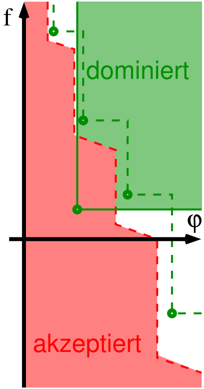
Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ wird von $\mathcal{F}_k$ für gegebene $0 < \gamma_1 < \gamma_2 < 1$ akzeptiert, falls

$$\forall j \in J_k : (\hat{\varphi} \le \gamma_2 \varphi_j) \text{ oder } (\hat{f} + \gamma_1 \hat{\varphi} \le f_j)$$

[Hinreichende Verbesserung in Verletzung oder Wert]

Wird (f_{k+1}, φ_{k+1}) von $\mathcal{F}_k$ akzeptiert, setze

$$\mathcal{F}_{k+1} := \{(f_{k+1}, \varphi_{k+1})\} \cup \{(\bar{f}, \bar{\varphi}) \in \mathcal{F}_k \text{ nicht von } (f_{k+1}, \varphi_{k+1}) \text{ dominiert}\}$$



Globalisierung mit einem Filter-Ansatz

Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)$ ... Verletzungsmaß, z.B. $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$.

Betrachte $(f_k, \varphi_k) := \big(f(x^{(k)}), \varphi(x^{(k)})\big)$ als Punkt in $\mathbb{R}^2$.

Ein Filter enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \big\{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \dots, k\}\big\}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ dominiert $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$, falls

$$\hat{f} \le \bar{f} \text{ und } \hat{\varphi} \le \bar{\varphi}.$$

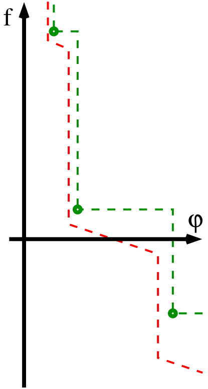
Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ wird von $\mathcal{F}_k$ für gegebene $0 < \gamma_1 < \gamma_2 < 1$ akzeptiert, falls

$$\forall j \in J_k : (\hat{\varphi} \le \gamma_2 \varphi_j) \text{ oder } (\hat{f} + \gamma_1 \hat{\varphi} \le f_j)$$

[Hinreichende Verbesserung in Verletzung oder Wert]

Wird (f_{k+1}, φ_{k+1}) von $\mathcal{F}_k$ akzeptiert, setze

$$\mathcal{F}_{k+1} := \{(f_{k+1}, \varphi_{k+1})\} \cup \{(\bar{f}, \bar{\varphi}) \in \mathcal{F}_k \text{ nicht von } (f_{k+1}, \varphi_{k+1}) \text{ dominiert}\}$$

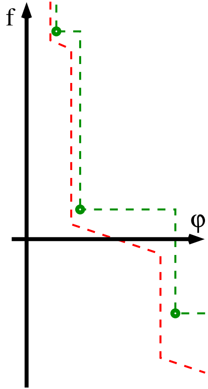


Globalisierung mit einem Filter-Ansatz

Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)\ldots$Verletzungsmaß, z.B. $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$.

Betrachte $(f_k, \varphi_k) := \big(f(x^{(k)}), \varphi(x^{(k)})\big)$ als Punkt in $\mathbb{R}^2$.



Ein Filter enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \big\{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \ldots, k\}\big\}.$$

Ein $(\hat f, \hat\varphi) \in \mathbb{R}^2$ dominiert $(\bar f, \bar\varphi) \in \mathbb{R}^2$, falls

$$\hat f \le \bar f \text{ und } \hat\varphi \le \bar\varphi.$$

Ein $(\hat f, \hat\varphi) \in \mathbb{R}^2$ wird von $\mathcal{F}_k$ für gegebene $0 < \gamma_1 < \gamma_2 < 1$ akzeptiert, falls

$$\forall j \in J_k : (\hat\varphi \le \gamma_2 \varphi_j) \text{ oder } (\hat f + \gamma_1 \hat\varphi \le f_j)$$

[Hinreichende Verbesserung in Verletzung oder Wert]

Wird (f_{k+1}, φ_{k+1}) von $\mathcal{F}_k$ akzeptiert, setze

$$\mathcal{F}_{k+1} := \{(f_{k+1}, \varphi_{k+1})\} \cup \{(\bar f, \bar\varphi) \in \mathcal{F}_k \text{ nicht von } (f_{k+1}, \varphi_{k+1}) \text{ dominiert}\}$$

[Der Filter sichert monotone Verbesserung und sammelt Erfahrung über gute Werte.]

Trust-Region-Variante eines Filter-Ansatzes

Schritt d_x löst approximativ ein polyedrisches Trust-Region-Unterproblem:

$$(QP_k^\Delta) \qquad \begin{array}{lll} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x & \\ \text{s.t.} & h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 & i \in \mathcal{E} \\ & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq 0 & i \in \mathcal{I} \\ & -\Delta \leq [d_x]_i \leq \Delta & i \in \{1, \ldots, n\} \end{array}$$

Trust-Region-Variante eines Filter-Ansatzes

Schritt d_x löst approximativ ein polyedrisches Trust-Region-Unterproblem:

$$(QP_k^\Delta) \qquad \begin{array}{lll} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x & \\ \text{s.t.} & h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 & i \in \mathcal{E} \\ & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq 0 & i \in \mathcal{I} \\ & -\Delta \leq [d_x]_i \leq \Delta & i \in \{1, \dots, n\} \end{array}$$

Falls (QP_k^Δ) unzulässig ist (in $\|d_x\|_\infty \leq \Delta$ ist kein zul. Punkt), führe Wiederherstellungs-Phase (restauration phase) durch:

Minimiere nur Verletzung $\varphi(x)$, bis für Filter akzeptabel $\to$ neues Δ, aber

$\Rightarrow$ der Filter darf nur unzulässige Punkte $(\hat{f}, \hat{\varphi})$ mit $\hat{\varphi} > 0$ enthalten!

Trust-Region-Variante eines Filter-Ansatzes

Schritt d_x löst approximativ ein polyedrisches Trust-Region-Unterproblem:

$$(QP_k^\Delta) \qquad \begin{array}{lll} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x & \\ \text{s.t.} & h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 & i \in \mathcal{E} \\ & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq 0 & i \in \mathcal{I} \\ & -\Delta \leq [d_x]_i \leq \Delta & i \in \{1, \dots, n\} \end{array}$$

Falls (QP_k^Δ) unzulässig ist (in $\|d_x\|_\infty \leq \Delta$ ist kein zul. Punkt), führe Wiederherstellungs-Phase (restauration phase) durch:

Minimiere nur Verletzung $\varphi(x)$, bis für Filter akzeptabel $\rightarrow$ neues Δ, aber
$\Rightarrow$ der Filter darf nur unzulässige Punkte $(\hat{f}, \hat{\varphi})$ mit $\hat{\varphi} > 0$ enthalten!

Verwende dazu folgende Sonderregel:
Vom Filter akzeptierte Punkte werden nur dann zum Filter hinzugefügt, wenn der nächste Schritt zu einer Wiederherstellung führt $\Rightarrow \varphi(x^{(k)}) > 0$.

Trust-Region-Variante eines Filter-Ansatzes

Schritt d_x löst approximativ ein polyedrisches Trust-Region-Unterproblem:

$$(QP_k^\Delta) \qquad \begin{array}{lll} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x & \\ \text{s.t.} & h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 & i \in \mathcal{E} \\ & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq 0 & i \in \mathcal{I} \\ & -\Delta \leq [d_x]_i \leq \Delta & i \in \{1, \ldots, n\} \end{array}$$

Falls (QP_k^Δ) unzulässig ist (in $\|d_x\|_\infty \leq \Delta$ ist kein zul. Punkt), führe Wiederherstellungs-Phase (restauration phase) durch:

Minimiere nur Verletzung $\varphi(x)$, bis für Filter akzeptabel $\rightarrow$ neues Δ, aber
$\Rightarrow$ der Filter darf nur unzulässige Punkte $(\hat{f}, \hat{\varphi})$ mit $\hat{\varphi} > 0$ enthalten!

Verwende dazu folgende Sonderregel:
Vom Filter akzeptierte Punkte werden nur dann zum Filter hinzugefügt, wenn der nächste Schritt zu einer Wiederherstellung führt $\Rightarrow \varphi(x^{(k)}) > 0$.

Der Algorithmus dazu nutzt weitere Parameter:

$\underline{\Delta}$	... Mindestradius nach Wiederherstellung
u	... maximale Unzulässigkeit für φ im Filter
$\sigma \in (0, 1)$	... Modellqualität für Reduktion von Δ

Filter-SQP-Algorithmus

0. Wähle $x^{(0)}$, $u \geq \varphi(x^{(0)})$, $\underline{\Delta} > 0$, $\sigma \in (0, 1)$, setze $\mathcal{F}_0 := \{(-\infty, u)\}$, $k := 1$

Filter-SQP-Algorithmus

0. Wähle $x^{(0)}$, $u \geq \varphi(x^{(0)})$, $\underline{\Delta} > 0$, $\sigma \in (0,1)$, setze $\mathcal{F}_0 := \{(-\infty, u)\}$, $k := 1$
1. Wiederherstellung: Ausgehend von $x^{(k-1)}$ finde $x^{(k)}$ und $\Delta \geq \underline{\Delta}$ mit (f_k, φ_k) wird von $\mathcal{F}_{k-1}$ akzeptiert und (QP_k^{Δ}) ist zulässig.

Filter-SQP-Algorithmus

0. Wähle $x^{(0)}$, $u \geq \varphi(x^{(0)})$, $\underline{\Delta} > 0$, $\sigma \in (0,1)$, setze $\mathcal{F}_0 := \{(-\infty, u)\}$, $k := 1$
1. Wiederherstellung: Ausgehend von $x^{(k-1)}$ finde $x^{(k)}$ und $\Delta \geq \underline{\Delta}$ mit (f_k, φ_k) wird von $\mathcal{F}_{k-1}$ akzeptiert und (QP_k^Δ) ist zulässig.
2. SQP-Schritt: Löse $(QP_k^\Delta) \to d_x$.
   Falls (QP_k^Δ) unzul., setze $\mathcal{F}_k := \{(f_k, \varphi_k)\} \cup \mathcal{F}_{k-1}$, $k \leftarrow k+1$, GOTO 1.

Filter-SQP-Algorithmus

0. Wähle $x^{(0)}$, $u \geq \varphi(x^{(0)})$, $\underline{\Delta} > 0$, $\sigma \in (0,1)$, setze $\mathcal{F}_0 := \{(-\infty, u)\}$, $k := 1$
1. Wiederherstellung: Ausgehend von $x^{(k-1)}$ finde $x^{(k)}$ und $\Delta \geq \underline{\Delta}$ mit (f_k, φ_k) wird von $\mathcal{F}_{k-1}$ akzeptiert und (QP_k^Δ) ist zulässig.
2. SQP-Schritt: Löse $(QP_k^\Delta) \to d_x$.
   Falls (QP_k^Δ) unzul., setze $\mathcal{F}_k := \{(f_k, \varphi_k)\} \cup \mathcal{F}_{k-1}$, $k \leftarrow k+1$, GOTO 1.
3. Ist $d_x = 0$, STOP, $x^{(k)}$ ist stationärer Punkt.

Filter-SQP-Algorithmus

0. Wähle $x^{(0)}$, $u \geq \varphi(x^{(0)})$, $\underline{\Delta} > 0$, $\sigma \in (0,1)$, setze $\mathcal{F}_0 := \{(-\infty, u)\}$, $k := 1$
1. Wiederherstellung: Ausgehend von $x^{(k-1)}$ finde $x^{(k)}$ und $\Delta \geq \underline{\Delta}$ mit (f_k, φ_k) wird von $\mathcal{F}_{k-1}$ akzeptiert und (QP_k^Δ) ist zulässig.
2. SQP-Schritt: Löse $(QP_k^\Delta) \rightarrow d_x$.
   Falls (QP_k^Δ) unzul., setze $\mathcal{F}_k := \{(f_k, \varphi_k)\} \cup \mathcal{F}_{k-1}$, $k \leftarrow k+1$, GOTO 1.
3. Ist $d_x = 0$, STOP, $x^{(k)}$ ist stationärer Punkt.
4. Wird $\left(f(x^{(k)} + d_x), \varphi(x^{(k)} + d_x)\right)$ nicht vom erweiterten Filter $\mathcal{F}_{k-1} \cup \{(f_k, \varphi_k)\}$ akzeptiert, setze $\Delta \leftarrow \Delta/2$, GOTO 2.

Filter-SQP-Algorithmus

0. Wähle $x^{(0)}$, $u \geq \varphi(x^{(0)})$, $\underline{\Delta} > 0$, $\sigma \in (0,1)$, setze $\mathcal{F}_0 := \{(-\infty, u)\}$, $k := 1$
1. Wiederherstellung: Ausgehend von $x^{(k-1)}$ finde $x^{(k)}$ und $\Delta \geq \underline{\Delta}$ mit (f_k, φ_k) wird von $\mathcal{F}_{k-1}$ akzeptiert und (QP_k^Δ) ist zulässig.
2. SQP-Schritt: Löse $(QP_k^\Delta) \to d_x$.
   Falls (QP_k^Δ) unzul., setze $\mathcal{F}_k := \{(f_k, \varphi_k)\} \cup \mathcal{F}_{k-1}$, $k \leftarrow k+1$, GOTO 1.
3. Ist $d_x = 0$, STOP, $x^{(k)}$ ist stationärer Punkt.
4. Wird $\left(f(x^{(k)} + d_x), \varphi(x^{(k)} + d_x)\right)$ nicht vom erweiterten Filter $\mathcal{F}_{k-1} \cup \{(f_k, \varphi_k)\}$ akzeptiert, setze $\Delta \leftarrow \Delta/2$, GOTO 2.
5. Setze $\Delta q := -\frac{1}{2} d_x^T Q_k d_x - \nabla f_k^T d_x$ und $\Delta f := f_k - f(x^{(k)} + d_x)$.
   Ist $\Delta q > 0$ und $\Delta f < \sigma \Delta q$, setze $\Delta \leftarrow \Delta/2$ und GOTO 2.
   [Der Fortschritt in f ist im Vergleich zum Modell zu gering.]

Filter-SQP-Algorithmus

0. Wähle $x^{(0)}$, $u \geq \varphi(x^{(0)})$, $\underline{\Delta} > 0$, $\sigma \in (0,1)$, setze $\mathcal{F}_0 := \{(-\infty, u)\}$, $k := 1$
1. Wiederherstellung: Ausgehend von $x^{(k-1)}$ finde $x^{(k)}$ und $\Delta \geq \underline{\Delta}$ mit (f_k, φ_k) wird von $\mathcal{F}_{k-1}$ akzeptiert und (QP_k^{Δ}) ist zulässig.
2. SQP-Schritt: Löse $(QP_k^{\Delta}) \to d_x$.
   Falls (QP_k^{Δ}) unzul., setze $\mathcal{F}_k := \{(f_k, \varphi_k)\} \cup \mathcal{F}_{k-1}$, $k \leftarrow k+1$, GOTO 1.
3. Ist $d_x = 0$, STOP, $x^{(k)}$ ist stationärer Punkt.
4. Wird $\big(f(x^{(k)} + d_x), \varphi(x^{(k)} + d_x)\big)$ nicht vom erweiterten Filter $\mathcal{F}_{k-1} \cup \{(f_k, \varphi_k)\}$ akzeptiert, setze $\Delta \leftarrow \Delta/2$, GOTO 2.
5. Setze $\Delta q := -\frac{1}{2} d_x^T Q_k d_x - \nabla f_k^T d_x$ und $\Delta f := f_k - f(x^{(k)} + d_x)$.
   Ist $\Delta q > 0$ und $\Delta f < \sigma \Delta q$, setze $\Delta \leftarrow \Delta/2$ und GOTO 2.
   [Der Fortschritt in f ist im Vergleich zum Modell zu gering.]
6. Ist $\Delta q < 0$, [$\Rightarrow x^{(k)}$ ist unzulässig, sonst wäre $d_x = 0$ besser]
   setze $\mathcal{F}_k := \{(f_k, \varphi_k)\} \cup \mathcal{F}_{k-1}$, sonst $\mathcal{F}_k := \mathcal{F}_{k-1}$

Filter-SQP-Algorithmus

0. Wähle $x^{(0)}$, $u \geq \varphi(x^{(0)})$, $\underline{\Delta} > 0$, $\sigma \in (0,1)$, setze $\mathcal{F}_0 := \{(-\infty, u)\}$, $k := 1$
1. Wiederherstellung: Ausgehend von $x^{(k-1)}$ finde $x^{(k)}$ und $\Delta \geq \underline{\Delta}$ mit (f_k, φ_k) wird von $\mathcal{F}_{k-1}$ akzeptiert und (QP_k^Δ) ist zulässig.
2. SQP-Schritt: Löse $(QP_k^\Delta) \to d_x$.
   Falls (QP_k^Δ) unzul., setze $\mathcal{F}_k := \{(f_k, \varphi_k)\} \cup \mathcal{F}_{k-1}$, $k \leftarrow k+1$, GOTO 1.
3. Ist $d_x = 0$, STOP, $x^{(k)}$ ist stationärer Punkt.
4. Wird $\big(f(x^{(k)} + d_x), \varphi(x^{(k)} + d_x)\big)$ nicht vom erweiterten Filter $\mathcal{F}_{k-1} \cup \{(f_k, \varphi_k)\}$ akzeptiert, setze $\Delta \leftarrow \Delta/2$, GOTO 2.
5. Setze $\Delta q := -\frac{1}{2} d_x^T Q_k d_x - \nabla f_k^T d_x$ und $\Delta f := f_k - f(x^{(k)} + d_x)$.
   Ist $\Delta q > 0$ und $\Delta f < \sigma \Delta q$, setze $\Delta \leftarrow \Delta/2$ und GOTO 2.
   [Der Fortschritt in f ist im Vergleich zum Modell zu gering.]
6. Ist $\Delta q < 0$, [$\Rightarrow x^{(k)}$ ist unzulässig, sonst wäre $d_x = 0$ besser]
   setze $\mathcal{F}_k := \{(f_k, \varphi_k)\} \cup \mathcal{F}_{k-1}$, sonst $\mathcal{F}_k := \mathcal{F}_{k-1}$
7. Setze $x^{(k+1)} := x^{(k)} + d_x$, $k \leftarrow k+1$, wähle $\Delta \geq \underline{\Delta}$, GOTO 2.

Filter-SQP-Algorithmus

0. Wähle $x^{(0)}$, $u \geq \varphi(x^{(0)})$, $\underline{\Delta} > 0$, $\sigma \in (0,1)$, setze $\mathcal{F}_0 := \{(-\infty, u)\}$, $k := 1$
1. Wiederherstellung: Ausgehend von $x^{(k-1)}$ finde $x^{(k)}$ und $\Delta \geq \underline{\Delta}$ mit (f_k, φ_k) wird von $\mathcal{F}_{k-1}$ akzeptiert und (QP_k^Δ) ist zulässig.
2. SQP-Schritt: Löse $(QP_k^\Delta) \to d_x$.
   Falls (QP_k^Δ) unzul., setze $\mathcal{F}_k := \{(f_k, \varphi_k)\} \cup \mathcal{F}_{k-1}$, $k \leftarrow k+1$, GOTO 1.
3. Ist $d_x = 0$, STOP, $x^{(k)}$ ist stationärer Punkt.
4. Wird $\left(f(x^{(k)} + d_x), \varphi(x^{(k)} + d_x)\right)$ nicht vom erweiterten Filter $\mathcal{F}_{k-1} \cup \{(f_k, \varphi_k)\}$ akzeptiert, setze $\Delta \leftarrow \Delta/2$, GOTO 2.
5. Setze $\Delta q := -\frac{1}{2} d_x^T Q_k d_x - \nabla f_k^T d_x$ und $\Delta f := f_k - f(x^{(k)} + d_x)$.
   Ist $\Delta q > 0$ und $\Delta f < \sigma \Delta q$, setze $\Delta \leftarrow \Delta/2$ und GOTO 2.
   [Der Fortschritt in f ist im Vergleich zum Modell zu gering.]
6. Ist $\Delta q < 0$, [$\Rightarrow x^{(k)}$ ist unzulässig, sonst wäre $d_x = 0$ besser]
   setze $\mathcal{F}_k := \{(f_k, \varphi_k)\} \cup \mathcal{F}_{k-1}$, sonst $\mathcal{F}_k := \mathcal{F}_{k-1}$
7. Setze $x^{(k+1)} := x^{(k)} + d_x$, $k \leftarrow k+1$, wähle $\Delta \geq \underline{\Delta}$, GOTO 2.

Unter geeigneten Voraussetzungen kann Konvergenz gegen einen stationären Punkt gezeigt werden.

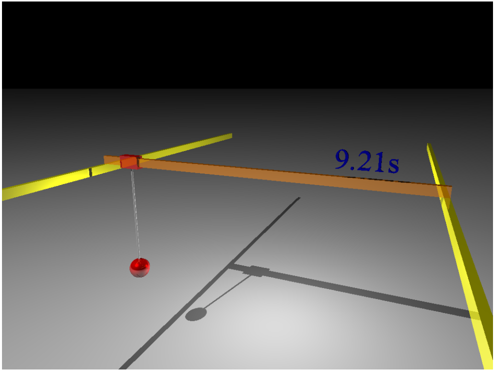
Inhaltsübersicht für heute:

SQP-Verfahren (Sequentielle quadratische Opt.-Verfahren)

Anwendungsbeispiel: Optimalsteuerung

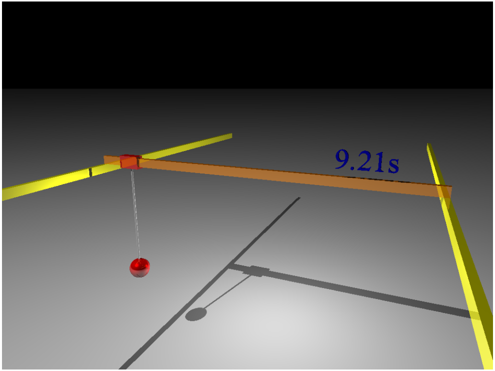
Anwendungsbeispiel: Optimalsteuerung

Wie muss die Laufkatze eines Industriekrans gesteuert werden, damit eine Last aus dem Ruhezustand in Punkt P möglichst schnell zu Punkt Q transportiert wird und dort wieder ruhig hängt?



Anwendungsbeispiel: Optimalsteuerung

Wie muss die Laufkatze eines Industriekrans gesteuert werden, damit eine Last aus dem Ruhezustand in Punkt P möglichst schnell zu Punkt Q transportiert wird und dort wieder ruhig hängt?

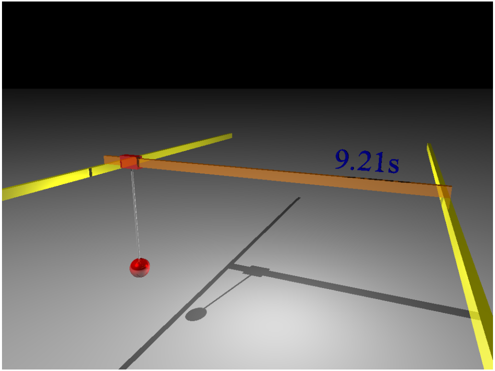


Die erste Idee – bis zur Hälfte voll beschleunigen, dann voll bremsen – ist nicht richtig.

Illustration

Anwendungsbeispiel: Optimalsteuerung

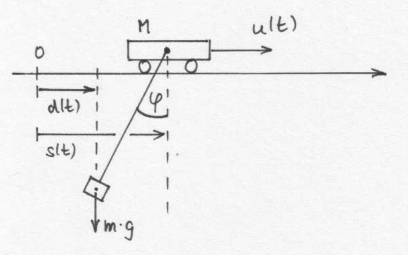
Wie muss die Laufkatze eines Industriekrans gesteuert werden, damit eine Last aus dem Ruhezustand in Punkt P möglichst schnell zu Punkt Q transportiert wird und dort wieder ruhig hängt?



Wie lässt sich das mathematisch modellieren?

(Material und Illustrationen von Prof. Dr. Roland Herzog)

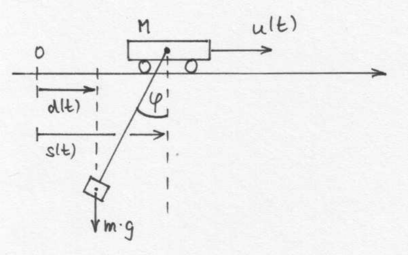
(eindimensionale) Laufkatze: Mathematisches Modell



$s(t)$: horizontale Position der Laufkatze
$d(t)$: horizontale Position der Last
M: die Masse der Laufkatze
m: die Masse der Last
$u(t)$: Steuerung, hier die auf die Laufkatze einwirkende Kraft (E-Motor)
$\varphi(t)$: Winkel der Last zur Vertikalen
g: die Erdbeschleunigung
L: die Länge des Seiles

(eindimensionale) Laufkatze: Mathematisches Modell

$s(t)$: horizontale Position der Laufkatze
$d(t)$: horizontale Position der Last
M: die Masse der Laufkatze
m: die Masse der Last
$u(t)$: Steuerung, hier die auf die Laufkatze einwirkende Kraft (E-Motor)
$\varphi(t)$: Winkel der Last zur Vertikalen
g: die Erdbeschleunigung
L: die Länge des Seiles

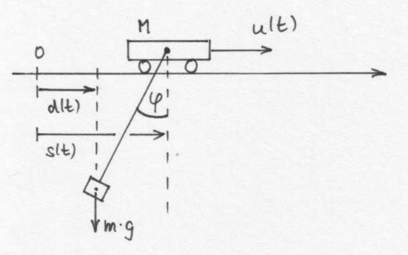


Der Zustand $y(t) \in \mathbb{R}^4$ wird durch Position $s(t)$ und Geschwindigkeit $\dot{s}(t) := \frac{\partial s}{\partial t}$ der Laufkatze sowie Relativposition $z(t) := s(t) - d(t)$ und Relativgeschwindigkeit $\dot{z}(t)$ der Last zur Laufkatze beschrieben,

$$y = (s, \dot{s}, z, \dot{z})^T. \qquad [t \text{ wird meist weggelassen}]$$

(eindimensionale) Laufkatze: Mathematisches Modell

$s(t)$: horizontale Position der Laufkatze
$d(t)$: horizontale Position der Last
M: die Masse der Laufkatze
m: die Masse der Last
$u(t)$: Steuerung, hier die auf die Laufkatze einwirkende Kraft (E-Motor)
$\varphi(t)$: Winkel der Last zur Vertikalen
g: die Erdbeschleunigung
L: die Länge des Seiles



Der Zustand $y(t) \in \mathbb{R}^4$ wird durch Position $s(t)$ und Geschwindigkeit $\dot{s}(t) := \frac{\partial s}{\partial t}$ der Laufkatze sowie Relativposition $z(t) := s(t) - d(t)$ und Relativgeschwindigkeit $\dot{z}(t)$ der Last zur Laufkatze beschrieben,

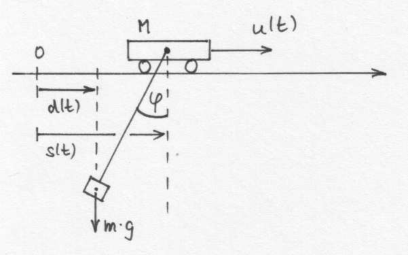
$$y = (s, \dot{s}, z, \dot{z})^T. \qquad \text{[}t \text{ wird meist weggelassen]}$$

Bewegungsgleichungen beschreiben die Beschleunigung und damit die Veränderung des Zustands über die Zeit. Für kleine Winkel φ ist

$$\begin{aligned} \ddot{s}(t) &= -\frac{m}{M}\frac{g}{L} z + \frac{u(t)}{M} \\ \ddot{z}(t) &= -\frac{(m+M)}{M}\frac{g}{L} z + \frac{u(t)}{M} \end{aligned}$$

(eindimensionale) Laufkatze: Mathematisches Modell

$s(t)$: horizontale Position der Laufkatze
$d(t)$: horizontale Position der Last
M: die Masse der Laufkatze
m: die Masse der Last
$u(t)$: Steuerung, hier die auf die Laufkatze einwirkende Kraft (E-Motor)
$\varphi(t)$: Winkel der Last zur Vertikalen
g: die Erdbeschleunigung
L: die Länge des Seiles



Der Zustand $y(t) \in \mathbb{R}^4$ wird durch Position $s(t)$ und Geschwindigkeit $\dot{s}(t) := \frac{\partial s}{\partial t}$ der Laufkatze sowie Relativposition $z(t) := s(t) - d(t)$ und Relativgeschwindigkeit $\dot{z}(t)$ der Last zur Laufkatze beschrieben,

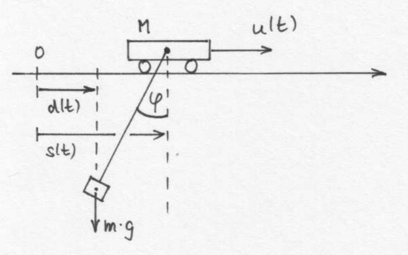
$$y = (s, \dot{s}, z, \dot{z})^T. \qquad [t \text{ wird meist weggelassen}]$$

Bewegungsgleichungen beschreiben die Beschleunigung und damit die Veränderung des Zustands über die Zeit. Für kleine Winkel φ ist

$$\begin{aligned} \ddot{s}(t) &= -\frac{m}{M}\frac{g}{L} z + \frac{u(t)}{M} \\ \ddot{z}(t) &= -\frac{(m+M)}{M}\frac{g}{L} z + \frac{u(t)}{M} \end{aligned} \quad \leftrightarrow \quad \dot{y} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{mg}{ML} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{(m+M)g}{ML} & 0 \end{bmatrix} y + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{M} \end{bmatrix} u$$

(eindimensionale) Laufkatze: Mathematisches Modell

$s(t)$: horizontale Position der Laufkatze
$d(t)$: horizontale Position der Last
M: die Masse der Laufkatze
m: die Masse der Last
$u(t)$: Steuerung, hier die auf die Laufkatze einwirkende Kraft (E-Motor)
$\varphi(t)$: Winkel der Last zur Vertikalen
g: die Erdbeschleunigung
L: die Länge des Seiles



Der Zustand $y(t) \in \mathbb{R}^4$ wird durch Position $s(t)$ und Geschwindigkeit $\dot{s}(t) := \frac{\partial s}{\partial t}$ der Laufkatze sowie Relativposition $z(t) := s(t) - d(t)$ und Relativgeschwindigkeit $\dot{z}(t)$ der Last zur Laufkatze beschrieben,

$$y = (s, \dot{s}, z, \dot{z})^T. \qquad [t \text{ wird meist weggelassen}]$$

Bewegungsgleichungen beschreiben die Beschleunigung und damit die Veränderung des Zustands über die Zeit. Für kleine Winkel φ ist

lineare gewöhnliche Differentialgleichung

$$\dot{y} = Ay + Bu \quad \leftrightarrow \quad \dot{y} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{mg}{ML} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{(m+M)g}{ML} & 0 \end{bmatrix} y + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{M} \end{bmatrix} u$$

Optimalsteueraufgaben für gew. Diffgl.

Ein System gewöhnlicher Differentialgleichungen

$$\dot{y}(t) = f(t, y(t), u(t))$$

beschreibe die Entwicklung des Zustands $y(t)$ in Abhängigkeit von Zeit t und Steuerung $u(t)$.

Optimalsteueraufgaben für gew. Diffgl.

Ein System gewöhnlicher Differentialgleichungen

$$\dot{y}(t) = f(t, y(t), u(t))$$

beschreibe die Entwicklung des Zustands $y(t)$ in Abhängigkeit von Zeit t und Steuerung $u(t)$.

Für $t \in [0, T]$ ist eine Steuerfunktion $u(t)$ gesucht, die den Zustand $y(t)$ von einem Anfangszustand $y(0) = y_0$ in einen Endzustand $y(T) = y_T$ überführt, dabei Zulässigkeitsbedingungen $h(t, y(t), u(t)) \leq 0$ erfüllt und zugleich eine Zielfunktion/Gütekriterium/Performance-Maß optimiert.

Optimalsteueraufgaben für gew. Diffgl.

Ein System gewöhnlicher Differentialgleichungen

$$\dot{y}(t) = f(t, y(t), u(t))$$

beschreibe die Entwicklung des Zustands $y(t)$ in Abhängigkeit von Zeit t und Steuerung $u(t)$.

Für $t \in [0, T]$ ist eine Steuerfunktion $u(t)$ gesucht, die den Zustand $y(t)$ von einem Anfangszustand $y(0) = y_0$ in einen Endzustand $y(T) = y_T$ überführt, dabei Zulässigkeitsbedingungen $h(t, y(t), u(t)) \leq 0$ erfüllt und zugleich eine Zielfunktion/Gütekriterium/Performance-Maß optimiert.

$$\begin{array}{lll}
\min\limits_{u} & g(y(T)) + \int_0^T f_0(t, y(t), u(t))\, dt & \text{Zielfunktion(al)} \\
\text{s.t.} & \dot{y}(t) = f(t, y(t), u(t)) & \text{Differentialgleichung} \\
& y(0) = y_0 & \text{Anfangsbedingungen} \\
& y(T) = y_T & \text{Endbedingungen} \\
& h(t, y(t), u(t)) \leq 0 & \text{Beschränkungen.}
\end{array}$$

Optimalsteueraufgaben für gew. Diffgl.

Ein System gewöhnlicher Differentialgleichungen

$$\dot{y}(t) = f(t, y(t), u(t))$$

beschreibe die Entwicklung des Zustands $y(t)$ in Abhängigkeit von Zeit t und Steuerung $u(t)$.

Für $t \in [0, T]$ ist eine Steuerfunktion $u(t)$ gesucht, die den Zustand $y(t)$ von einem Anfangszustand $y(0) = y_0$ in einen Endzustand $y(T) = y_T$ überführt, dabei Zulässigkeitsbedingungen $h(t, y(t), u(t)) \leq 0$ erfüllt und zugleich eine Zielfunktion/Gütekriterium/Performance-Maß optimiert.

$$\begin{array}{lll}
\min\limits_u & g(y(T)) + \int_0^T f_0(t, y(t), u(t))\,dt & \text{Zielfunktion(al)} \\
\text{s.t.} & \dot{y}(t) = f(t, y(t), u(t)) & \text{Differentialgleichung} \\
& y(0) = y_0 & \text{Anfangsbedingungen} \\
& y(T) = y_T & \text{Endbedingungen} \\
& h(t, y(t), u(t)) \leq 0 & \text{Beschränkungen.}
\end{array}$$

Die **Endzeit** T kann dabei fest gegeben oder auch frei (d.h. Teil der zu optimierenden Größen) sein, ebenso könnten y_0 oder y_T teilweise frei sein.

Zeitoptimale Steuerung der Laufkatze

Die Laufkatze soll mit Last in kürzest möglicher Zeit T vom Stillstand am Punkt P zum Stillstand an einem Punkt Q gebracht werden.
$\rightarrow$ Zielfunktion:

Zeitoptimale Steuerung der Laufkatze

Die Laufkatze soll mit Last in kürzest möglicher Zeit T vom Stillstand am Punkt P zum Stillstand an einem Punkt Q gebracht werden.
$\rightarrow$ Zielfunktion: $\min \int_0^T 1 dt [= T]$

$\rightarrow y(0) =$

Zeitoptimale Steuerung der Laufkatze

Die Laufkatze soll mit Last in kürzest möglicher Zeit T vom Stillstand am Punkt P zum Stillstand an einem Punkt Q gebracht werden.

$\rightarrow$ Zielfunktion: $\min \int_0^T 1 dt [= T]$

$$\rightarrow y(0) = \begin{pmatrix} P \\ 0 \\ 0 \\ 0 \end{pmatrix} \begin{array}{l} \text{Position der Laufkatze} \\ \text{Relativposition der Last zur Laufkatze} \\ \text{Geschwindigkeit der Laufkatze} \\ \text{Relativgeschwindigkeit der Last zur Laufkatze} \end{array}$$

$\rightarrow y(T) =$

Zeitoptimale Steuerung der Laufkatze

Die Laufkatze soll mit Last in kürzest möglicher Zeit T vom Stillstand am Punkt P zum Stillstand an einem Punkt Q gebracht werden.

$\rightarrow$ Zielfunktion: $\min \int_0^T 1 dt [= T]$

$$\rightarrow y(0) = \begin{pmatrix} P \\ 0 \\ 0 \\ 0 \end{pmatrix} \begin{array}{l} \text{Position der Laufkatze} \\ \text{Relativposition der Last zur Laufkatze} \\ \text{Geschwindigkeit der Laufkatze} \\ \text{Relativgeschwindigkeit der Last zur Laufkatze} \end{array}$$

$\rightarrow y(T) = (Q, 0, 0, 0)^T$

Steuerbeschränkungen: $-1 \leq u(t) \leq 1$ [keine unendliche Beschleunigung]

Es könnte auch Zustandsbeschränkungen für $y(t)$ (z.B. Hindernisse) geben.

Zeitoptimale Steuerung der Laufkatze

Die Laufkatze soll mit Last in kürzest möglicher Zeit T vom Stillstand am Punkt P zum Stillstand an einem Punkt Q gebracht werden.

$\rightarrow$ Zielfunktion: $\min \int_0^T 1 dt [= T]$

$$\rightarrow y(0) = \begin{pmatrix} P \\ 0 \\ 0 \\ 0 \end{pmatrix} \begin{array}{l} \text{Position der Laufkatze} \\ \text{Relativposition der Last zur Laufkatze} \\ \text{Geschwindigkeit der Laufkatze} \\ \text{Relativgeschwindigkeit der Last zur Laufkatze} \end{array}$$

$\rightarrow y(T) = (Q, 0, 0, 0)^T$

Steuerbeschränkungen: $-1 \leq u(t) \leq 1$ [keine unendliche Beschleunigung]

Es könnte auch Zustandsbeschränkungen für $y(t)$ (z.B. Hindernisse) geben.

Hauptproblem noch:

T ist eine Variable und gibt zugleich den Definitionsbereich von u an!

Transformation von freier Endzeit auf feste Endzeit

$\min_{u,T}$	$g(y(T)) + \int_0^T f_0(t, y(t), u(t))\, dt$	Zielfunktion(al)
s.t.	$\dot{y}(t) = f(t, y(t), u(t))$	Differentialgleichung
	$y(0) = y_0$	Anfangsbedingungen
	$y(T) = y_T$	Endbedingungen
	$h(t, y(t), u(t)) \leq 0$	Beschränkungen.

Transformation von freier Endzeit auf feste Endzeit

$\min_{u,T}$	$g(y(T)) + \int_0^T f_0(t, y(t), u(t))\, dt$	Zielfunktion(al)
s.t.	$\dot{y}(t) = f(t, y(t), u(t))$	Differentialgleichung
	$y(0) = y_0$	Anfangsbedingungen
	$y(T) = y_T$	Endbedingungen
	$h(t, y(t), u(t)) \leq 0$	Beschränkungen.

Führe eine neue Zeitvariable $\tau \in [0,1]$ ein: $t = \tau T$, $\tau \in [0,1]$

Definiere neue Zustände/Steuerung für τ: $\widetilde{y}(\tau) := y(\tau T)$, $\widetilde{u}(\tau) := u(\tau T)$

Transformation von freier Endzeit auf feste Endzeit

$\min_{u,T}$	$g(y(T)) + \int_0^T f_0(t, y(t), u(t))\, dt$	Zielfunktion(al)
s.t.	$\dot{y}(t) = f(t, y(t), u(t))$	Differentialgleichung
	$y(0) = y_0$	Anfangsbedingungen
	$y(T) = y_T$	Endbedingungen
	$h(t, y(t), u(t)) \leq 0$	Beschränkungen.

Führe eine neue Zeitvariable $\tau \in [0, 1]$ ein: $t = \tau T$, $\tau \in [0, 1]$

Definiere neue Zustände/Steuerung für τ: $\widetilde{y}(\tau) := y(\tau T)$, $\widetilde{u}(\tau) := u(\tau T)$

Die Differentialgleichung bezüglich τ ergibt sich aus der Kettenregel,

$$\frac{d\widetilde{y}}{d\tau}(\tau) = T\, \frac{dy}{dt}(\tau T) = Tf(\tau T, \widetilde{y}(\tau), \widetilde{u}(\tau)).$$

Transformation von freier Endzeit auf feste Endzeit

$$
\begin{array}{lll}
\min_{u,T} & g(y(T)) + \int_0^T f_0(t, y(t), u(t))\, dt & \text{Zielfunktion(al)} \\
\text{s.t.} & \dot{y}(t) = f(t, y(t), u(t)) & \text{Differentialgleichung} \\
 & y(0) = y_0 & \text{Anfangsbedingungen} \\
 & y(T) = y_T & \text{Endbedingungen} \\
 & h(t, y(t), u(t)) \le 0 & \text{Beschränkungen.}
\end{array}
$$

Führe eine neue Zeitvariable $\tau \in [0,1]$ ein: $t = \tau T$, $\tau \in [0,1]$

Definiere neue Zustände/Steuerung für τ: $\widetilde{y}(\tau) := y(\tau T)$, $\widetilde{u}(\tau) := u(\tau T)$

Die Differentialgleichung bezüglich τ ergibt sich aus der Kettenregel,

$$\frac{d\widetilde{y}}{d\tau}(\tau) = T\,\frac{dy}{dt}(\tau T) = Tf(\tau T, \widetilde{y}(\tau), \widetilde{u}(\tau)).$$

Die neue Zielfunktion lässt sich über die Substitutionsregel berechnen,

$$g(y(T)) + \int_0^T f_0(t, y(t), u(t))\, dt \quad \rightarrow \quad g(\widetilde{y}(1)) + T\int_0^1 f_0(\tau T, \widetilde{y}(\tau), \widetilde{u}(\tau))\, d\tau.$$

Transformation von freier Endzeit auf feste Endzeit

$\min_{u,T}$	$g(y(T)) + \int_0^T f_0(t, y(t), u(t))\, dt$	Zielfunktion(al)
s.t.	$\dot{y}(t) = f(t, y(t), u(t))$	Differentialgleichung
	$y(0) = y_0$	Anfangsbedingungen
	$y(T) = y_T$	Endbedingungen
	$h(t, y(t), u(t)) \le 0$	Beschränkungen.

Führe eine neue Zeitvariable $\tau \in [0,1]$ ein: $t = \tau T$, $\tau \in [0,1]$

Definiere neue Zustände/Steuerung für τ: $\tilde{y}(\tau) := y(\tau T)$, $\tilde{u}(\tau) := u(\tau T)$

Die Differentialgleichung bezüglich τ ergibt sich aus der Kettenregel,

$$\frac{d\tilde{y}}{d\tau}(\tau) = T\,\frac{dy}{dt}(\tau T) = Tf(\tau T, \tilde{y}(\tau), \tilde{u}(\tau)).$$

Die neue Zielfunktion lässt sich über die Substitutionsregel berechnen,

$$g(y(T)) + \int_0^T f_0(t, y(t), u(t))\, dt \quad \to \quad g(\tilde{y}(1)) + T\int_0^1 f_0(\tau T, \tilde{y}(\tau), \tilde{u}(\tau))\, d\tau.$$

$\min_{\tilde{u},T}$	$g(\tilde{y}(1)) + T\int_0^1 f_0(\tau T, \tilde{y}(\tau), \tilde{u}(\tau))\, d\tau$	
s.t.	$\dot{\tilde{y}}(\tau) = Tf(\tau T, \tilde{y}(\tau), \tilde{u}(\tau))$	[Def-Bereich von $\tilde{u}$ unabhängig von T, T ist nun „normale“ Variable!]
	$\tilde{y}(0) = y_0$	
	$\tilde{y}(1) = y_T$	
	$h(\tau T, \tilde{y}(\tau), \tilde{u}(\tau)) \le 0.$	

Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne $\sim$ aber mit neuer Zeit $\tau \in [0,1]$)

$$\begin{array}{ll} \min\limits_{u:[0,1]\to\mathbb{R},\ T\in\mathbb{R}} & T \\ \text{s.t.} & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \qquad \text{[nichtlineare Nebenbed.!]} \\ & y(0) = (P,0,0,0)^T \\ & y(1) = (Q,0,0,0)^T \\ & -1 \le u(\tau) \le 1 \quad \text{für alle } \tau \in [0,1]. \end{array}$$

Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne $\sim$ aber mit neuer Zeit $\tau \in [0,1]$)

$$\begin{array}{ll} \min\limits_{u:[0,1]\to\mathbb{R},\ T\in\mathbb{R}} & T \\ \text{s.t.} & \dot{y}(\tau) = T A y(\tau) + T B u(\tau) \qquad \text{[nichtlineare Nebenbed.!]} \\ & y(0) = (P,0,0,0)^T \\ & y(1) = (Q,0,0,0)^T \\ & -1 \le u(\tau) \le 1 \quad \text{für alle } \tau \in [0,1]. \end{array}$$

Variablen:

Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne $\sim$ aber mit neuer Zeit $\tau \in [0,1]$)

$$\begin{array}{ll} \min\limits_{u:[0,1]\to\mathbb{R},\ T\in\mathbb{R}} & T \\ \text{s.t.} & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \qquad \text{[nichtlineare Nebenbed.!]} \\ & y(0) = (P,0,0,0)^T \\ & y(1) = (Q,0,0,0)^T \\ & -1 \le u(\tau) \le 1 \quad \text{für alle } \tau \in [0,1]. \end{array}$$

Variablen: T, $u(\tau)$ für $\tau \in [0,1]$ $\quad\rightarrow\quad$ ∞-dimensional

Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne $\sim$ aber mit neuer Zeit $\tau \in [0,1]$)

$$\begin{array}{lll} \min\limits_{u:[0,1]\to\mathbb{R},\ T\in\mathbb{R}} & T & \\ \text{s.t.} & \dot{y}(\tau) = T A y(\tau) + T B u(\tau) & \text{[nichtlineare Nebenbed.!]} \\ & y(0) = (P,0,0,0)^T & \\ & y(1) = (Q,0,0,0)^T & \\ & -1 \leq u(\tau) \leq 1 \quad \text{für alle } \tau \in [0,1]. & \end{array}$$

Variablen: T, $u(\tau)$ für $\tau \in [0,1]$ $\quad\rightarrow\quad$ ∞-dimensional

Wie optimiert man über Funktionen?

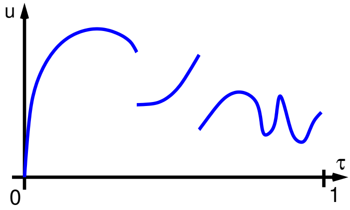
Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne $\sim$ aber mit neuer Zeit $\tau \in [0,1]$)

$$
\begin{array}{lll}
\min\limits_{u:[0,1]\to\mathbb{R},\ T\in\mathbb{R}} & T & \\
\text{s.t.} & \dot{y}(\tau) = T A y(\tau) + T B u(\tau) & \text{[nichtlineare Nebenbed.!]} \\
 & y(0) = (P,0,0,0)^T & \\
 & y(1) = (Q,0,0,0)^T & \\
 & -1 \le u(\tau) \le 1 \quad \text{für alle } \tau \in [0,1]. &
\end{array}
$$

Variablen: T, $u(\tau)$ für $\tau \in [0,1]$ $\quad\rightarrow\quad$ ∞-dimensional

Wie optimiert man über Funktionen?
Wenn die Funktion nicht zu oft springen darf (z.B. stetig oder stückweise stetig), reichen die Funktionswerte an endlich vielen Stellen aus, um sie gut zu approximieren.



Das Laufkatzenproblem nach Transformation

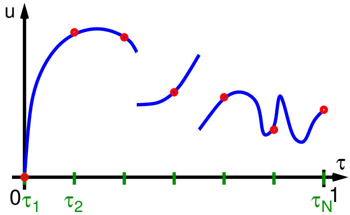
Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne $\sim$ aber mit neuer Zeit $\tau \in [0,1]$)

$$\min_{u:[0,1]\to\mathbb{R},\ T\in\mathbb{R}} \quad T$$

$$\begin{aligned} \text{s.t.} \quad & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \qquad \text{[nichtlineare Nebenbed.!]} \\ & y(0) = (P,0,0,0)^T \\ & y(1) = (Q,0,0,0)^T \\ & -1 \le u(\tau) \le 1 \quad \text{für alle } \tau \in [0,1]. \end{aligned}$$

Variablen: T, $u(\tau)$ für $\tau \in [0,1]$ $\quad\to\quad$ ∞-dimensional

Wie optimiert man über Funktionen?
Wenn die Funktion nicht zu oft springen darf (z.B. stetig oder stückweise stetig), reichen die Funktionswerte an endlich vielen Stellen aus, um sie gut zu approximieren.
$\to$ Diskretisierung



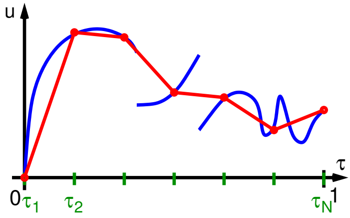
Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne $\sim$ aber mit neuer Zeit $\tau \in [0,1]$)

$$\begin{aligned} \min_{u:[0,1]\to\mathbb{R},\ T\in\mathbb{R}} \quad & T \\ \text{s.t.} \quad & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \qquad \text{[nichtlineare Nebenbed.!]} \\ & y(0) = (P,0,0,0)^T \\ & y(1) = (Q,0,0,0)^T \\ & -1 \le u(\tau) \le 1 \quad \text{für alle } \tau \in [0,1]. \end{aligned}$$

Variablen: T, $u(\tau)$ für $\tau \in [0,1]$ $\rightarrow$ ∞-dimensional

Wie optimiert man über Funktionen?
Wenn die Funktion nicht zu oft springen darf (z.B. stetig oder stückweise stetig), reichen die Funktionswerte an endlich vielen Stellen aus, um sie gut zu approximieren.
$\rightarrow$ Diskretisierung und, z.B., lineare Interpolation $\rightarrow$ Diskretisierungsfehler



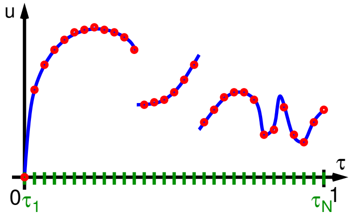
Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne $\sim$ aber mit neuer Zeit $\tau \in [0,1]$)

$$\begin{array}{ll} \min\limits_{u:[0,1]\to\mathbb{R},\ T\in\mathbb{R}} & T \\ \text{s.t.} & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \qquad \text{[nichtlineare Nebenbed.!]} \\ & y(0) = (P,0,0,0)^T \\ & y(1) = (Q,0,0,0)^T \\ & -1 \le u(\tau) \le 1 \quad \text{für alle } \tau \in [0,1]. \end{array}$$

Variablen: T, $u(\tau)$ für $\tau \in [0,1]$ $\quad\rightarrow\quad$ ∞-dimensional

Wie optimiert man über Funktionen?
Wenn die Funktion nicht zu oft springen darf (z.B. stetig oder stückweise stetig), reichen die Funktionswerte an endlich vielen Stellen aus, um sie gut zu approximieren.
$\rightarrow$ Diskretisierung und, z.B., lineare Interpolation $\rightarrow$ Diskretisierungsfehler



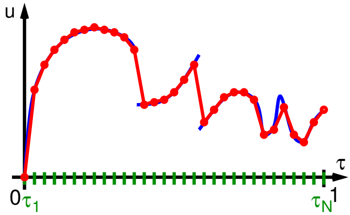
Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne $\sim$ aber mit neuer Zeit $\tau \in [0,1]$)

$$\begin{array}{lll} \min\limits_{u:[0,1]\to\mathbb{R},\ T\in\mathbb{R}} & T & \\ \text{s.t.} & \dot{y}(\tau) = T A y(\tau) + T B u(\tau) & \text{[nichtlineare Nebenbed.!]} \\ & y(0) = (P,0,0,0)^T & \\ & y(1) = (Q,0,0,0)^T & \\ & -1 \le u(\tau) \le 1 \quad \text{für alle } \tau \in [0,1]. & \end{array}$$

Variablen: T, $u(\tau)$ für $\tau \in [0,1]$ $\quad\rightarrow\quad$ ∞-dimensional

Wie optimiert man über Funktionen?
Wenn die Funktion nicht zu oft springen darf (z.B. stetig oder stückweise stetig), reichen die Funktionswerte an endlich vielen Stellen aus, um sie gut zu approximieren.
$\rightarrow$ Diskretisierung und, z.B., lineare Interpolation $\rightarrow$ Diskretisierungsfehler



Laufkatze: Diskretisierung des Optimierungsproblems

Ersetze die Funktionsvariable $u : [0,1] \to \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \cdots < \tau_N = 1$, $N \in \mathbb{N}$, durch einen Variablenvektor $u \in \mathbb{R}^N$ mit Interpretation $u_i = u(\tau_i)$ und (bei linearer Interpolation) $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i}(u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$.

$$
\begin{array}{ll}
\min & T \\
\text{s.t.} & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \quad \tau \in [0,1] \\
& y(0) = (P, 0, 0, 0)^T \\
& y(1) = (Q, 0, 0, 0)^T \\
& -1 \leq u_i \leq 1 \ (i = 1, \ldots, N),\ T \in \mathbb{R}_+.
\end{array}
$$

Laufkatze: Diskretisierung des Optimierungsproblems

Ersetze die Funktionsvariable $u : [0,1] \to \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \cdots < \tau_N = 1$, $N \in \mathbb{N}$, durch einen Variablenvektor $u \in \mathbb{R}^N$ mit Interpretation $u_i = u(\tau_i)$ und (bei linearer Interpolation) $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i}(u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$.

$$\begin{array}{ll} \min & T \\ \text{s.t.} & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \quad \tau \in [0,1] \\ & y(0) = (P,0,0,0)^T \\ & y(1) = (Q,0,0,0)^T \\ & -1 \le u_i \le 1 \ (i = 1,\dots,N),\ T \in \mathbb{R}_+. \end{array}$$

Was geschieht mit der Differentialgleichung?

Laufkatze: Diskretisierung des Optimierungsproblems

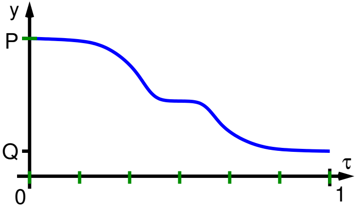
Ersetze die Funktionsvariable $u : [0,1] \to \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \cdots < \tau_N = 1$, $N \in \mathbb{N}$, durch einen Variablenvektor $u \in \mathbb{R}^N$ mit Interpretation $u_i = u(\tau_i)$ und (bei linearer Interpolation) $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i}(u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$.

$$\begin{array}{ll} \min & T \\ \text{s.t.} & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \quad \tau \in [0,1] \\ & y(0) = (P,0,0,0)^T \\ & y(1) = (Q,0,0,0)^T \\ & -1 \leq u_i \leq 1 \ (i = 1,\ldots,N),\ T \in \mathbb{R}_+. \end{array}$$

Was geschieht mit der Differentialgleichung? Berechne

$$y(\tau) = y(0) + \int_0^\tau TAy(t) + TBu(t)dt$$

über ein numerisches Integrationsverfahren, das das Integral wieder durch Diskretisierung über Summen approximiert (z.B. Euler- oder Runge-Kutta-Verfahren).



Laufkatze: Diskretisierung des Optimierungsproblems

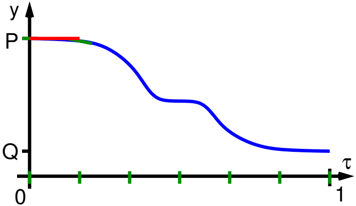
Ersetze die Funktionsvariable $u : [0,1] \to \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \cdots < \tau_N = 1$, $N \in \mathbb{N}$, durch einen Variablenvektor $u \in \mathbb{R}^N$ mit Interpretation $u_i = u(\tau_i)$ und (bei linearer Interpolation) $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i}(u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$.

$$\begin{array}{ll} \min & T \\ \text{s.t.} & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \quad \tau \in [0,1] \\ & y(0) = (P,0,0,0)^T \\ & y(1) = (Q,0,0,0)^T \\ & -1 \le u_i \le 1 \ (i = 1,\dots,N),\ T \in \mathbb{R}_+. \end{array}$$

Was geschieht mit der Differentialgleichung? Berechne

$$y(\tau) = y(0) + \int_0^\tau TAy(t) + TBu(t)dt$$

über ein numerisches Integrationsverfahren, das das Integral wieder durch Diskretisierung über Summen approximiert (z.B. Euler- oder Runge-Kutta-Verfahren).



Laufkatze: Diskretisierung des Optimierungsproblems

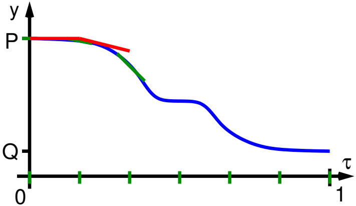
Ersetze die Funktionsvariable $u : [0, 1] \to \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \cdots < \tau_N = 1$, $N \in \mathbb{N}$, durch einen Variablenvektor $u \in \mathbb{R}^N$ mit Interpretation $u_i = u(\tau_i)$ und (bei linearer Interpolation) $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i}(u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$.

$$\begin{array}{ll} \min & T \\ \text{s.t.} & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \quad \tau \in [0, 1] \\ & y(0) = (P, 0, 0, 0)^T \\ & y(1) = (Q, 0, 0, 0)^T \\ & -1 \le u_i \le 1 \ (i = 1, \ldots, N), T \in \mathbb{R}_+. \end{array}$$

Was geschieht mit der Differentialgleichung? Berechne

$$y(\tau) = y(0) + \int_0^\tau TAy(t) + TBu(t)dt$$

über ein numerisches Integrationsverfahren, das das Integral wieder durch Diskretisierung über Summen approximiert (z.B. Euler- oder Runge-Kutta-Verfahren).



Laufkatze: Diskretisierung des Optimierungsproblems

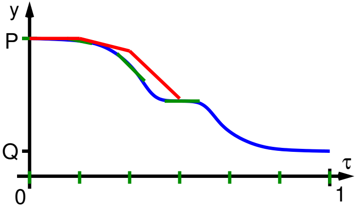
Ersetze die Funktionsvariable $u : [0,1] \to \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \cdots < \tau_N = 1$, $N \in \mathbb{N}$, durch einen Variablenvektor $u \in \mathbb{R}^N$ mit Interpretation $u_i = u(\tau_i)$ und (bei linearer Interpolation) $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i}(u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$.

$$\begin{array}{ll} \min & T \\ \text{s.t.} & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \quad \tau \in [0,1] \\ & y(0) = (P,0,0,0)^T \\ & y(1) = (Q,0,0,0)^T \\ & -1 \le u_i \le 1 \ (i = 1,\ldots,N),\ T \in \mathbb{R}_+. \end{array}$$

Was geschieht mit der Differentialgleichung? Berechne

$$y(\tau) = y(0) + \int_0^\tau TAy(t) + TBu(t)dt$$

über ein numerisches Integrationsverfahren, das das Integral wieder durch Diskretisierung über Summen approximiert (z.B. Euler- oder Runge-Kutta-Verfahren).



Laufkatze: Diskretisierung des Optimierungsproblems

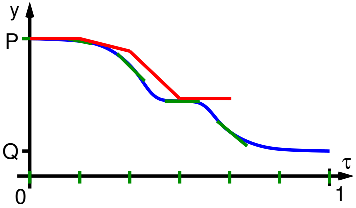
Ersetze die Funktionsvariable $u : [0,1] \to \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \cdots < \tau_N = 1$, $N \in \mathbb{N}$, durch einen Variablenvektor $u \in \mathbb{R}^N$ mit Interpretation $u_i = u(\tau_i)$ und (bei linearer Interpolation) $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i}(u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$.

$$\begin{array}{ll} \min & T \\ \text{s.t.} & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \quad \tau \in [0,1] \\ & y(0) = (P,0,0,0)^T \\ & y(1) = (Q,0,0,0)^T \\ & -1 \le u_i \le 1 \ (i = 1,\dots,N),\ T \in \mathbb{R}_+. \end{array}$$

Was geschieht mit der Differentialgleichung? Berechne

$$y(\tau) = y(0) + \int_0^\tau TAy(t) + TBu(t)dt$$

über ein numerisches Integrationsverfahren, das das Integral wieder durch Diskretisierung über Summen approximiert (z.B. Euler- oder Runge-Kutta-Verfahren).



Laufkatze: Diskretisierung des Optimierungsproblems

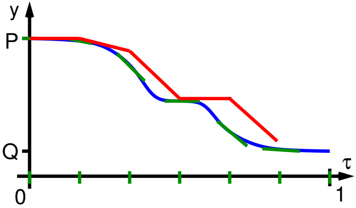
Ersetze die Funktionsvariable $u : [0,1] \to \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \cdots < \tau_N = 1$, $N \in \mathbb{N}$, durch einen Variablenvektor $u \in \mathbb{R}^N$ mit Interpretation $u_i = u(\tau_i)$ und (bei linearer Interpolation) $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i}(u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$.

$$
\begin{array}{ll}
\min & T \\
\text{s.t.} & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \quad \tau \in [0,1] \\
& y(0) = (P, 0, 0, 0)^T \\
& y(1) = (Q, 0, 0, 0)^T \\
& -1 \leq u_i \leq 1 \ (i = 1, \ldots, N),\ T \in \mathbb{R}_+.
\end{array}
$$

Was geschieht mit der Differentialgleichung? Berechne

$$y(\tau) = y(0) + \int_0^\tau TAy(t) + TBu(t)dt$$

über ein numerisches Integrationsverfahren, das das Integral wieder durch Diskretisierung über Summen approximiert (z.B. Euler- oder Runge-Kutta-Verfahren).



Laufkatze: Diskretisierung des Optimierungsproblems

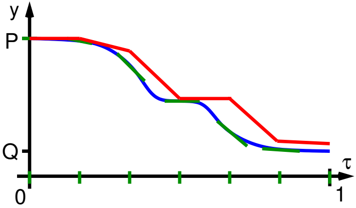
Ersetze die Funktionsvariable $u : [0,1] \to \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \cdots < \tau_N = 1$, $N \in \mathbb{N}$, durch einen Variablenvektor $u \in \mathbb{R}^N$ mit Interpretation $u_i = u(\tau_i)$ und (bei linearer Interpolation) $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i}(u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$.

$$
\begin{array}{ll}
\min & T \\
\text{s.t.} & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \quad \tau \in [0,1] \\
& y(0) = (P,0,0,0)^T \\
& y(1) = (Q,0,0,0)^T \\
& -1 \le u_i \le 1 \ (i = 1,\dots,N),\ T \in \mathbb{R}_+.
\end{array}
$$

Was geschieht mit der Differentialgleichung? Berechne

$$y(\tau) = y(0) + \int_0^\tau TAy(t) + TBu(t)dt$$

über ein numerisches Integrationsverfahren, das das Integral wieder durch Diskretisierung über Summen approximiert (z.B. Euler- oder Runge-Kutta-Verfahren).



$\to$ Funktionsorakel $y(\tau) = R(u,\tau)$, das $y(0) = (P,0,0,0)^T$ erfüllt.

$\to$ Es bleibt nur eine Glgs-Nebenbedingung: $R(u,1) - (Q,0,0,0)^T = 0$

Funktionsauswertung mit Gradienten über automatisches Differenzieren!

Laufkatze: Diskretisierte Optimierungsaufgabe

$$\begin{array}{ll} \min & T \\ \text{s.t.} & R(u,1) - (Q,0,0,0)^T = 0 \\ & -1 \leq u_i \leq 1 \ (i = 1, \ldots, N),\ T \in \mathbb{R}_+. \end{array}$$

Dieses kann z.B. mit einem SQP-Verfahren gelöst werden, wobei die Gradienten zum Runge-Kutta-Verfahren $R(\cdot)$ über automatisches Differenzieren und die Hessematrix über BFGS approximiert wird.

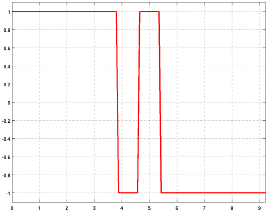
Laufkatze: Diskretisierte Optimierungsaufgabe

$$\begin{array}{ll} \min & T \\ \text{s.t.} & R(u,1) - (Q,0,0,0)^T = 0 \\ & -1 \leq u_i \leq 1 \ (i = 1,\ldots,N),\ T \in \mathbb{R}_+. \end{array}$$

Dieses kann z.B. mit einem SQP-Verfahren gelöst werden, wobei die Gradienten zum Runge-Kutta-Verfahren $R(\cdot)$ über automatisches Differenzieren und die Hessematrix über BFGS approximiert wird.

$\rightarrow$ Approximation der zeitoptimalen Steuerung
$u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i}(u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$

Illustration



Laufkatze: Diskretisierte Optimierungsaufgabe

$$\begin{array}{ll} \min & T \\ \text{s.t.} & R(u,1) - (Q,0,0,0)^T = 0 \\ & -1 \le u_i \le 1 \ (i = 1,\dots,N),\ T \in \mathbb{R}_+. \end{array}$$

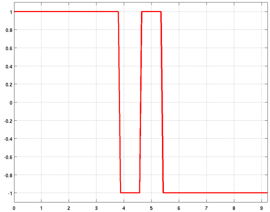
Dieses kann z.B. mit einem SQP-Verfahren gelöst werden, wobei die Gradienten zum Runge-Kutta-Verfahren $R(\cdot)$ über automatisches Differenzieren und die Hessematrix über BFGS approximiert wird.

$\to$ Approximation der zeitoptimalen Steuerung
$u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i}(u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$

Illustration

Ganz analog sind zeitoptimale 2D-Steuerungen berechenbar,

2D-Illustration



Laufkatze: Diskretisierte Optimierungsaufgabe

$$\begin{array}{ll} \min & T \\ \text{s.t.} & R(u,1) - (Q,0,0,0)^T = 0 \\ & -1 \leq u_i \leq 1 \ (i = 1,\dots,N),\ T \in \mathbb{R}_+. \end{array}$$

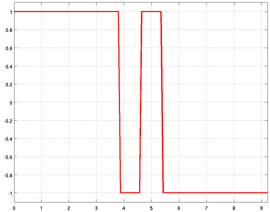
Dieses kann z.B. mit einem SQP-Verfahren gelöst werden, wobei die Gradienten zum Runge-Kutta-Verfahren $R(\cdot)$ über automatisches Differenzieren und die Hessematrix über BFGS approximiert wird.

$\rightarrow$ Approximation der zeitoptimalen Steuerung
$u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i}(u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$

Illustration

Ganz analog sind zeitoptimale 2D-Steuerungen berechenbar,

2D-Illustration



Die Qualität der Lösung und der notwendige Aufwand zur Bestimmung der Lösung sind stark von der Wahl der Diskretisierung abhängig!