N+1 vektorwertige Nebenbedingungen $\Rightarrow$ N+1 vektorwertige Lagrange-Multipl. $\underline{\lambda}_0, \ldots, \underline{\lambda}_N$

$\hookrightarrow$ Lagrange-Funktion:

$$\mathcal{L}(\underline{x}_{0:N}, a_{0:N-1}, \underline{\lambda}_{0:N}) = J + (\underline{c} - x_0)^T \cdot \underline{\lambda}_0 + \sum_{k=0}^{N-1} \underbrace{(\underline{h}_k(\underline{x}_k, \underline{a}_k) - \underline{x}_{k+1}) \cdot \underline{\lambda}_{k+1}}_{\hat{=}\, n_x \text{ endim. NB}}$$

Def.: Hamilton-Funktion

$$H_k(\underline{x}_k, a_k, \lambda_{k+1}) = g_k(x_k, a_k) + \underline{h}_k(\underline{x}_k, \underline{a}_k)^T \cdot \lambda \ldots\ldots$$

$\hookrightarrow$ $\mathcal{L}(\underline{x}_{0:N}, a_{0:N-1}, \underline{\lambda}_{0:N}) = g_N(\underline{x}_N) + (\underline{c} - \underline{x}_0)\underline{\lambda}_0 + \sum_{k=0}^{N-1} (H_k - x_{k+1}^T \cdot \lambda_{k+1})$

Lösung des Optimierungsproblems

- Ableitung bzgl. $\underline{x}_n$:

$$\frac{\partial \mathcal{L}}{\partial \underline{x}_N} = \frac{\partial}{\partial x_N} g_N(x_N) - \underline{\lambda}_N$$

$$\frac{\partial \mathcal{L}}{\partial \underline{x}_n} = \frac{\partial}{\partial \underline{x}_n} H_k - \lambda_k = \underbrace{\nabla_{x_k} g_k(\underline{x}_k, \underline{a}_k)}_{=: g_k^x} + \big(\underbrace{\nabla_{\underline{x}_k} \underline{h}_k(\underline{x}_k, \underline{a}_k)}_{=: h_k^x}\big)^T \cdot \underline{\lambda}_{k+1} - \underline{\lambda}_k$$

mit $\frac{\partial \mathcal{L}}{\partial \underline{x}_k} = 0$ folgt:

$$\boxed{\begin{aligned} \underline{\lambda}_N &= g_N^x(\underline{x}_N) \\ \underline{\lambda}_k &= g_k^x(\underline{x}_k, \underline{a}_k) + \underline{h}_k^x(\underline{x}_k, \underline{a}_k)^T \cdot \underline{\lambda}_{k+1} \end{aligned}}$$

Rückwärtsrekursion, $k = 0, \ldots, N-1$

$\hookrightarrow$ Rückwärtsrekursion für $\underline{\lambda}_k$

- Ableitung bzgl. a_k:

$$\frac{\partial \mathcal{L}}{\partial \underline{a}_k} = \nabla_{a_k} H_k(x_k, a_k, \lambda_{k+1}) = \underbrace{\nabla_{a_k} g_k(x_k, \underline{a}_k)}_{= g_k^a} + \big(\underbrace{\nabla_{\underline{a}_k} h_k(x_k, a_k)}_{=: h_k^a}\big)^T \cdot \lambda_{k+1}$$

$$\stackrel{!}{=} 0$$

$\hookrightarrow$ Notwendige Bedingung für eine Minimierung von H_k bzgl. a_k

Ableitung bzgl. $\underline{\lambda}_k$:

$$\frac{\partial \mathcal{L}}{\partial \underline{\lambda}_0} = (\underline{c} - \underline{x}_0) \stackrel{!}{=} \underline{0}$$

$$\frac{\partial \mathcal{L}}{\partial \underline{\lambda}_k} = \frac{\partial}{\partial \underline{\lambda}_k} H_k - \underline{x}_k = \underline{h}_{k-1}(x_{k-1}, \ldots$$

(Ls) Nebenbed. der Optimierung

Insgesamt: Notwendige Bedingungen an die Optimierung sind

Pontryagin'sches Minimums-Prinzip

$$x_{k+1} = h_k(x_k, a_k)$$

$$\underline{\lambda}_k = g_k^x(\underline{x}_k, \underline{a}_k) + \underline{h}_k^x(\underline{x}_k, a_k)^T \cdot \underline{\lambda}_{k+1}$$

$$0 = \nabla_{a_k} H_k(\underline{x}_k, \underline{a}_k, \underline{\lambda}_{k+1}) \quad (a_k = \arg\min H_k(x_k, a_k, \lambda_{k+1}))$$

für $k = 0, \dots, N-1$ mit $\underline{\lambda}_N = g_N^x(\underline{x}_N)$ und $\underline{x}_0$ fest.

Vorteil des Pontryagin'schen Minimums-Prinzip:

⊕ i.A. einfacher zu lösen als DP, da nur für einen Startzustand $\underline{x}_0$

⊖ wie DP i.A. nicht geschlossen lösbar, aber liefert effiziente Vorschrift zur Berechnung des Gradienten von $J(x_{0:N}, \underline{a}_{0:N-1})$ bzgl. $a_{0:N-1}$

Satz: Gegeben sei der Plan $a_{0:N-1}$. Weiter seien x_k, λ_k durch

$$x_{k+1} = h_k(x_k, a_k)$$

$$\underline{\lambda}_k = g_k^x(\underline{x}_k, \underline{a}_k) + h_k^x(x_k, a_k)^T \cdot \underline{\lambda}_{k+1}$$

definiert, mit $\underline{x}_0$ gegeben und $\underline{\lambda}_N = g_N^x(x_N)$. ~~$x_0$ x_3 fest~~:
Dann gilt für den Gradient

$$\nabla_{a_k} J(x_{0:N-1}, a_{0:N-1}) = \nabla_{a_k} H_k(\underline{x}_k, a_k, \lambda_{k+1})$$

$g_0 + g_1 + \dots g_{N-1} + g_N$

Beweis

- Definiere Kostenfunktion für $k = 0, \dots, N-1$

$$J_k'(x_{k,N}, a_{k,N-1}) = g_k(x_k, a_k) + J_{k+1}'(x_{k+1,N}, a_{k+1,N-1})$$

- Ableitung bzgl. a_k:

$$\frac{\partial J_k'}{\partial a_k} = g_k^a(\underline{x}_k, \underline{a}_k) + \frac{\partial J_{k+1}'}{\partial a_k} \qquad J_{k+1}'(h_k(x_k, a_k), x_{k+2:N}, a_{k+1:N-1})$$

$= \{a_k$ beeinflusst J_{k+1}' lediglich durch $\underline{x}_{k+1} = \underline{h}_k(x_k, a_k)\}$

$$= g_k^a(\underline{x}_k, \underline{a}_k) + h_k^a(x_k, a_k)^T \cdot \nabla x_{k+1} J_{k+1}'$$

Behauptung: $\lambda_k = \nabla x_k J_k'$:

- für $k = N$ gilt $\nabla x_N J_N' = g_N^x(x_N) = \lambda_N$
- für $k < N$ gilt $\nabla_{x_k} J_k' = g_k^x(\underline{x}_k, a_k) + h_k^x(x_k, a_k)^T \cdot \underbrace{\nabla_{k+1} J_{k+1}'}_{\text{IV } \lambda_{k+1}}$

$= \lambda_k$

↳ Gradient $\nabla_{a_k} J_k' = g_k^a + \begin{pmatrix} a \\ h_k \end{pmatrix}^T \cdot x_{k+1} = \nabla_{a_k} H_k$ (*)

Nur notw. Bedingung!

<u>Bezug zur dynamischen Programmierung (DP):</u>

- DP: $J_k(\underline{x}_k) = \min_{a_k} \{ g_k(\underline{x}_k, a_k) + J_{k+1}(h_k(\underline{x}_k, a_k)) \}$
  ↳ $\frac{\partial}{\partial a_k}(g_k + J_{k+1}) = g_k^a(x_k, \underline{a}_k) + (\underline{h}_k^a(x_k, a_k))^T \cdot \underline{\nabla_{x_{k+1}} J_{k+1}} \overset{!}{=} \underline{0}$

- Vergleich mit (*)

$$\frac{\partial}{\partial a_k} H_k = g_k^a(\underline{x}_k, a_k) + (h_k^a(x_k, a_k))^T \cdot \underline{x_{k+1}} \overset{!}{=} \underline{0}$$

↳ Lagrange-Mult. entsprechen dem Gradienten der Wertefunktion bzgl. $\underline{x}_k$

A11 „Trajektorienverfolgungsproblem"

a) Ableitung bzgl. x_k

$$g_k^x = \frac{\partial}{\partial x_k}\left(Q_k(x_k - \hat{x}_k)^2 + R_k a_k^2\right) = 2 \cdot Q_k (x_k - \hat{x}_k)$$

$$g_N^x = \frac{\partial}{\partial x_N} Q_N (x_N - \hat{x}_N)^2 = 2 \cdot Q_N \cdot (x_N - \hat{x}_N)$$

$$h_k^x = \frac{\partial}{\partial x_k}(A_k \cdot x_k + B_k \cdot a_k) = A_k$$

Ableitung bzgl. a_k:

$$g_k^a = \frac{\partial}{\partial a_k}\left(Q_k(x_k - \hat{x}_k)^2 + R_k a_k^2\right) = 2R_k \cdot a_k$$

$$h_k^a = B_k$$

Beobachtung: Lineares Zustandsraummodell, quadratische Kosten

↳ H_k ist konvex bzgl. a_k (für festes x_k, λ_{k+1})

↳ Notw. Bedingung ist auch hinreichend

↳ Gradientenabstieg liefert globales Minimum (optimale Lösung)

b) Algorithmus:

1. wähle $a_{0:N-1}$ beliebig (bel. Plan)
2. Bestimme $x_{1:N-1}$ (Zustandstrajektorie) durch Auswerten des Zustandsraummodells für gegebenes x_0 und $a_{0:N-1}$

3. Berechne Lagrange-Multiplikatoren $\lambda_{1:N}$ und Gradient $\nabla_{a_k} H_k$
4. $a_{0:N-1}$ ← Abstieg in neg. Richtung des Gradienten.
5. Zurück zu 2. solange bis $a_{0:N-1}$ nicht konvergiert

Spezialfall: Lineare Planungsprobleme

Gegeben: Lineares Zustandsraummodell

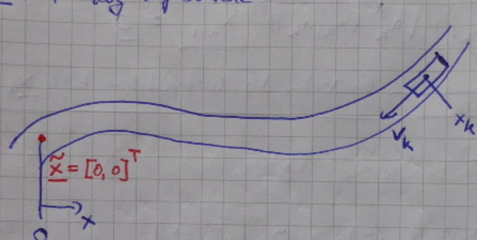
Rauschen

$$x_{k+1} = A_k \cdot \underline{x}_{k+1} + B_k \cdot a_k + \underline{w}_k \quad \text{für } k=0,\dots,N-1$$

Eingangsmatrix

Systemmatrix

Beispiel: Fahrzeug auf Schiene



Bahn ist parametrische Funktion, x ist Parameter

$$\underline{x}_k = \begin{bmatrix} x_k \\ v_k \end{bmatrix} \begin{matrix} \dots \text{Pos} \\ \dots \text{Geschw.} \end{matrix}$$

a_k ... Geschw.-Änderung (Beschleunigung)

Mit Abtastintervall T:

$$A_k = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \qquad B_k = T \begin{bmatrix} \frac{1}{2} \cdot T \\ 1 \end{bmatrix}$$

$$x_{k+1} = x_k + T \cdot v_k \qquad a_k \cdot B_k = \begin{bmatrix} \frac{1}{2} a_k \cdot T^2 \\ a_k \cdot T \end{bmatrix}$$

Zustands- und Aktionsmenge: $X = \mathbb{R}^{n_x}$, $A = \mathbb{R}^{n_a}$

Rauschen w_k:

- mittelwertfrei, d.h. $E\{w_k\} = 0$
- unabh. von x_k, a_k und w_i mit $i \neq k$
- charakterisiert durch Dichtefunktion $f_k^w(\underline{w}_k)$

Ziel: Überführung von Initialzustand $x_0 \rightarrow \tilde{x} = [0, \dots 0]^T$

durch Minimierung der quadratischen Kostenfunktion

$$E\left\{ \underline{x}_N^T \, Q_N \, x_N + \sum_{k=0}^{N-1} \left(x_k^T \cdot Q_k \cdot x_k + a_k^T \cdot R_k \cdot a_k \right) \right\}$$

symmetrisch, pos. definit

Bestimmung der opt. Strategie

- DP: $J_N(x_N) = \underline{x}_N^T \cdot Q_N \underline{x}_N$

terminale Kosten

$$J_k(x_k) = \min_{\underline{a}_k} \{ x_k^T Q_k x_k + \underline{a}_k^T \cdot R_k \underline{a}_k + \mathbb{E}\{ J_{k+1}(x_{k+1}) | \underline{x}_{k+1}, a_k \} \}$$
$$\text{for } k = 0, \dots, N-1$$

- Betrachte Erwartungswert:

$$\mathbb{E}\{ J_{k+1}(x_{k+1}) | \underline{x}_k, \underline{a}_k \}$$
$$= \int_{\mathcal{X}} J_{k+1}(x_{k+1}) \cdot \underbrace{f(x_{k+1} | \underline{x}_k, \underline{a}_k)}_{} \, d\underline{x}_{k+1}$$
$$= f_k^w(\underline{x}_{k+1} - A_k x_k - B_k a_k)$$
$$= \int_\omega \delta(\underline{x}_{k+1} - (A_k x_k + B_k a_k + \underline{\omega}_k)) \cdot f_k^\omega(\underline{\omega}_k) \, d\underline{\omega}_k$$
$$= \int_\omega J_{k+1}(A_k \underline{x}_k + B_k a_k + \underline{\omega}_k) \cdot f_k^\omega(\underline{\omega}_k) \, d\underline{\omega}_k$$

Rauschen

$$= \mathbb{E}_w \{ J_{k+1}(A_k \cdot \underline{x}_k + B_k \cdot \underline{a}_k + \underline{\omega}_k) / \underline{x}_k, \underline{a}_k \}$$

Induktionsbeweis: Behauptung $J_{k+1}(\underline{x}_{k+1}) = \underline{x}_{k+1}^T \cdot P_{k+1} \cdot \underline{x}_{k+1} + c_{k+1}$

mit P_{k+1} ... symmetrisch, pos.-semidefinit Matrix

c_{k+1} ... reele Konstante

- Induktionsanfang: $k = N-1$

$\Rightarrow P_N = Q_N$ und $c_N = 0$

- Induktionshypothese: Behauptung gilt für $k+1$
- Induktionsschluss:

$$J_k(x_k) = \min_{a_k} \{ x_k^T Q_k \underline{x}_k + a_k^T R_k a_k + \mathbb{E}_w \{ (\underbrace{A_k \cdot x_k + B_k a_k}_{} + \omega_k)^T P_{k+1} \cdot$$
$$(\underbrace{A_k x_k + B_k a_k}_{} + \omega_k) + c_{k+1} \} \}$$

$$= x_k^T \cdot Q_k x_k + \min_{a_k} \underbrace{\{ a_k^T R_k a_k + a_k^T B_k^T P_{k+1} B_k a_k + 2 \cdot x_k^T A_k^T P_{k+1} B_k a_k \}}_{(*)}$$
$$+ \underline{x}_k^T A_k^T P_{k+1} A_k \underline{x}_k + \mathbb{E}\{ w_k^T P_{k+1} \underline{\omega}_k \} + c_{k+1} \qquad (**)$$

- Minimierung von (*) → Ableiten bzgl. a_k und zu Null setzen

- $\frac{\partial x^T A x}{\partial x} = 2 \cdot A x$ sofern A symmetrisch

- $\frac{\partial \underline{a}^T \cdot x}{\partial x} = \frac{\partial x^T \cdot \underline{a}}{\partial x} = \underline{a}$

Symmetrische Matrix per Definition

↳ $(*)' = \cancel{2} R_k \underline{a}_k + \cancel{2} B_k^T \cdot P_{k+1} \cdot B_k \cdot a_k + \cancel{2} \cdot B_k^T P_{k+1} \cdot a_k \overset{!}{=} 0$

↳ $(R_k + B_k^T P_{k+1} B_k) \cdot \underline{a}_k = - B_k^T \cdot P_{k+1} \cdot A_k \underline{x}_k$ (Lineares GLS bzgl. a_k)

↳ $a_k^* = \underbrace{-(R_k + B_k^T P_{k+1} \cdot B_k)^{-1} \cdot B_k^T \cdot P_{k+1} \cdot A_k}_{=: L_k} \cdot \underline{x}_k$ „Verstärkungsmatrix"

↳ $\boxed{\text{Opt. Strategie } \pi_k^*(\underline{x}_k) = L_k \cdot \underline{x}_k}$ ⇝ Linear in x_k

Linearer quadratischer Regulator (LQR)

Zustandsraummodell — Kosten — wurde zuerst in der Regelungstechnik entdeckt

- Abschluss der Induktion: Einsetzen von a_k^* in $(**)$ liefert

$$J_k(x_k) = \underline{x}_k^T P_k \cdot \underline{x}_k + c_k$$

mit
$$P_k = A_k^T (P_{k+1} - P_{k+1} B_k (B_k^T P_{k+1} B_k + R_k)^{-1} B_k^T P_{k+1}) A_k + Q_k$$
$$c_k = \mathbb{E}\{ \underline{w}_k^T \cdot P_{k+1} \cdot \underline{w}_k \} + c_{k+1}$$

↳ Rückwärtsrekursion für P_k mit $P_N = Q_N$

Zeitdiskrete Riccati-Gleichung

Interpretation der Kostenfunktion

$$\underbrace{\underline{x}_N^T \cdot Q_N \cdot \underline{x}_N + \sum_{k=0}^{N-1} x_k^T Q_k x_k}_{\text{Zustandsabhängige Kosten}} \quad + \quad \underbrace{\sum_{k=0}^{N-1} a_k^T R_k a_k}_{\text{aktionsabhängige Kosten}}$$

Bewerte Abweichung vom Ziel
$$\tilde{\underline{x}} = [0 \ldots 0]^T$$

Ist Q_k diagonal mit $q_{ii} \geq 0$
$$\underline{x}_k^T Q_k \underline{x}_k = \sum_{i=1}^{L_x} q_{ii} \cdot x_{k,i}^2$$

Gewichtung — quadr. Abweichung

- Ist dabei $q_{ii} = 0$, so wird die Abweichung in $x_{k,i}$ nicht berücksichtigt
- Große q_{ii} bestrafen Abw. stärker

Bewerte aufgewendete „Energie" um $\tilde{x}$ zu erreichen

Ist R_k Diagonalmatrix mit $r_{ii} \geq 0$
$$\underline{a}_k^T R_k a_k = \sum_{i=1}^{L_a} r_{ii} a_{k,i}^2$$

- Ist $r_{ii} = 0$, so kann $a_{k,i}$ sehr groß sein bzw. evtl. gar nicht realisierbar
- Große r_{ii} bestrafen die „Energie"

↳ Glättung von des Verlauf von $\underline{x}_k$

Beispiel-Fall: $A_k = B_k = 1$ und $R_k = r \cdot I$ $Q_k = q \cdot I$ für alle k mit $q, r \in \mathbb{R}^+$

$-q \gg r$: $P_{N-1} = (Q_N - Q_N \underbrace{(Q_N + R_{N-1})^{-1}}_{\approx Q_N} Q_N) + Q_{N-1} \approx Q_{N-1}$

$\hookrightarrow P_k \approx Q_k$ für alle k

$\hookrightarrow L_k \approx -(Q_{k+1} + R_k)^{-1} Q_{k+1} \approx -I$

$\hookrightarrow \underline{a}_k^* \approx -\underline{x}_k$

$-r \gg q$: $P_{N-1} = (Q_N - \underbrace{Q_N (Q_N + R_{N-1})^{-1} Q_N}_{\approx \underline{0}\text{-Matrix}}) + Q_{N-1} \approx Q_N + Q_{N-1} = 2 \cdot q \cdot I$

Also proport. stetiges Ansteigen der aufgewendeten Energie

$\hookrightarrow P_k \approx (N-k+1) \cdot q I = (N-k+1) Q_k$

$\hookrightarrow L_k \approx -\underbrace{((N-k+1) \cdot Q_{k+1} + R_k)^{-1}}_{\approx 0} \cdot Q_{k+1} \approx 0$

$\hookrightarrow a_k^* \approx \underline{0} \Rightarrow$ keine Zustandsänderung

$-q = r$: $L_k \hat{=}$ goldener Schnitt

Kritische Betrachtung:

⊕ geschlossen berechenbare der ~~gesuchten~~ opt. Strategie

⊖ viele Freiheitsgrade, z.B. Q_k besitzt $\frac{1}{2} n_x (n_x + 1)$ freie Parameter (und P-Matrix, für jeden Zeitschritt k)

⊖ physikalischer Zusammenhang schwer/gar nicht herstellbar

$\hookrightarrow$ iterativer Prozess zur Bestimmung von Q_k, P_k (vertretbar falls Simulation des Systems vorhanden)

$\boxed{A12}$

a) Wahl: $Q_k = q_k \cdot I$ mit $q_k = 10$
$Q_N = q_N I$ und $q_N = 20$
$R_k = r_k \cdot I$ mit $r_k = 1$

b) • Krümmung der Bahn nicht berücksichtigt

• Beschleunigung evtl. nicht realisierbar

• Geschwindigkeit evtl. nicht realisierbar

<u>Sicherheitsäquivalenz</u> (engl. certainty equivalence)

• Beobachtung: Verstärkungsmatrix L_k und Strategie $\underline{\pi}_k^*$ sind unabhängig vom Rauschen $\underline{w}_k$

• Grund: Lin. Modell und quadratische Kostenfunktion (und weißes (unkorreliertes) Rauschen)

$\Rightarrow$ Dieselbe optimale Strategie ergibt sich bei Betrachtung des korrespondierenden det. Zustandsraummodells $x_{k+1} = A_k \cdot x_k + B_k \cdot a_k$ bei welchem das Rauschen durch dessen EW $E\{w_k\} = 0$ ersetzt ist.