

Übung 10 - LQG, Numerische Optimierung, MPC

1 LQG

Im LQR Problem wurde es angenommen, dass alle Zustände des Systems verfügbar waren. In meisten praktischen Situationen das ist trotzdem nicht der Fall, darum muss ein Beobachter implementiert werden.

1.1 Beobachter

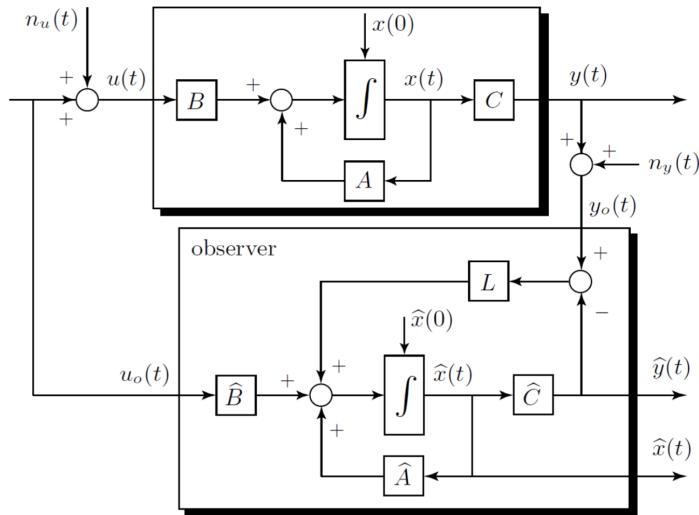


Abbildung 1: Struktur eines Zustandsbeobachter.

Definiere den Fehler als

$$\bar{x}(t) = x(t) - \hat{x}(t), \quad (1.1)$$

wobei $x(t)$ ist der Zustand des Systems und $\hat{x}(t)$ der Zustand des Beobachters. Die Dynamik des Fehlers wird beschrieben durch

$$\begin{aligned} \frac{d}{dt} \bar{x}(t) &= \frac{d}{dt} x(t) - \frac{d}{dt} \hat{x}(t) \\ &= A \cdot x(t) + B \cdot u(t) - (\hat{A} \cdot \hat{x}(t) + \hat{B} \cdot u(t) + L \cdot (C \cdot x(t) - \hat{C} \cdot \hat{x}(t))) \\ &= A \cdot \bar{x}(t) - L \cdot C \cdot \bar{x}(t) \\ &= (A - L \cdot C) \cdot \bar{x}(t), \end{aligned} \quad (1.2)$$

d.h. \bar{x} konvergiert falls $A - L \cdot C$ asymptotisch stabil ist. Das Problem kann als LQR betrachtet werden, da LQR garantiert, dass die Eigenwerte von $A - B \cdot K$ eine negativen Realteil besitzen. Darum¹

$$(A - L \cdot C)^\top = A^\top - C^\top \cdot L^\top, \quad (1.3)$$

ist äquivalent zu

$$A - B \cdot K. \quad (1.4)$$

Die Matrix L^\top kann also durch Lösen des LQR Problems mit $\{A^\top, C^\top, \bar{B}^\top\}$. Das Ergebnis ist in Tabelle 1 zusammengefasst.

¹Die Eigenwerte von M zu den Eigenwerten von M^\top .

LQR	LQG
$[A - B \cdot K]$	$[(A - L \cdot C)^\top] = [(A^\top - C^\top \cdot L^\top)]$
A	A^\top
B	C^\top
$Q = \bar{C}^\top \cdot \bar{C}$	$\bar{B} \cdot \bar{B}^\top$
$R = r \cdot \mathbb{I}$	$q \cdot \mathbb{I}$

Tabelle 1: LQG und LQR.

Die Matrix L ist gegeben durch

$$L^\top = \frac{1}{q} \cdot C \cdot \Psi, \quad (1.5)$$

wobei Φ ist die Lösung der Riccati Gleichung

$$\frac{1}{q} \cdot \Psi \cdot C^\top \cdot C \cdot \Psi - \Psi \cdot A^\top - A \cdot \Psi - \bar{B} \cdot \bar{B}^\top = 0. \quad (1.6)$$

Das kann durch ‘‘Substitution’’ der Matrizen in die Lösung des LQR

$$K = R^{-1} \cdot B^\top \cdot \Phi, \quad (1.7)$$

wobei Φ ist die Lösung der Riccati Gleichung

$$\Phi \cdot B \cdot R^{-1} \cdot B^\top \cdot \Phi - \Phi \cdot A - A^\top \cdot \Phi - Q = 0. \quad (1.8)$$

1.2 LQG Regler

Der LQG Regler ist die Kombination zwischen LQR und Beobachter. Man benutzt den Regler K (Lösung von LQR) und den Signal $\hat{x}(t)$ aus dem Beobachter, d.h.

$$u(t) = -K \cdot \hat{x}(t). \quad (1.9)$$

Somit können Systeme, von welchen den Zustand $x(t)$ nicht bekannt ist, mit LQR geregelt werden.

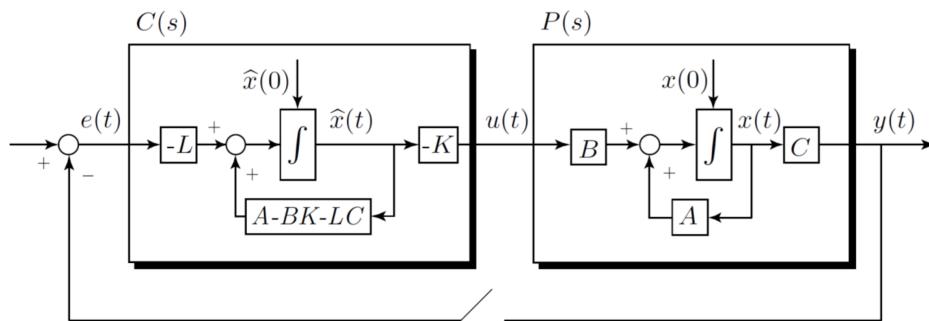


Abbildung 2: Struktur eines Zustandsbeobachter.

Führe die Zustandsvariable

$$\tilde{x} = \begin{bmatrix} x(t) \\ \hat{x}(t) \end{bmatrix} \quad (1.10)$$

bekommt man folgende Open-loop Dynamik

$$\frac{d}{dt} \tilde{x}(t) = \tilde{A}_{ol} \cdot \tilde{x}(t) + \tilde{B} \cdot e(t), \quad (1.11)$$

mit

$$\tilde{A}_{\text{ol}} = \begin{bmatrix} A - B \cdot K & 0 \\ 0 & A - B \cdot K - L \cdot C \end{bmatrix}, \quad \tilde{B} = \begin{bmatrix} 0 \\ -L \end{bmatrix}, \quad \tilde{C} = [C \ 0]. \quad (1.12)$$

Die Closed-loop Dynamik ist beschrieben durch

$$\frac{d}{dt} \tilde{x}(t) = \tilde{A}_{\text{cl}} \cdot \tilde{x}(t) + \tilde{B} \cdot e(t), \quad (1.13)$$

mit

$$\tilde{A}_{\text{cl}} = \begin{bmatrix} A & -B \cdot K \\ L \cdot C & A - B \cdot K - L \cdot C \end{bmatrix}. \quad (1.14)$$

In Frequenzbereich bekommt man

$$L_{\text{LQG}}(s) = P(s) \cdot C(s) = \underbrace{C \cdot (s \cdot I - A)^{-1} \cdot B}_{P(s)} \underbrace{K \cdot (s \cdot I - (A - B \cdot K - L \cdot C))^{-1} \cdot L}_{C(s)}, \quad (1.15)$$

$$T_{\text{LQG}}(s) = L_{\text{LQG}}(s) \cdot (\mathbb{I} + L_{\text{LQG}}(s))^{-1}. \quad (1.16)$$

Bemerkung. Stabilität und Robustheit des Systems müssen “a posteriori” überprüft werden, d.h. alle Garantie die wir für LQR hatten sind verloren.

1.2.1 Methode

1. Finde K : lös ein standard LQR-Problem mit $\{A, B, Q = \bar{C}^\top \cdot C, R\}$;
2. Finde L : lös ein standard LQR-Problem mit $\{A^\top, C^\top, Q = \bar{B} \cdot \bar{B}^\top, R = q \cdot \mathbb{I}\}$.

Matlab Befehle:

```
K = lqr(A,B,C_tilde'*C_tilde,r*eye(m,m)), L = lqr(A',C',B_bar*B_bar',q*eye(m,m)).
```

Bemerkung. Es muss gelten:

- $\{A, C\}$ vollständig beobachtbar;
- $\{A, \bar{B}\}$ vollständig steuerbar.

1.3 LQGI

Eine möglich Erweiterung des LQG Reglers ist der LQGI Regler. Hier wird ein Integralteil ins LQG System hinzugefügt. Zu bestimmen sind also K , K_I und L .

1.3.1 Methode

1. Finde K und K_I : lös ein standard LQR-Problem mit $\{\tilde{A}, \tilde{B}, \tilde{C}, \tilde{R}\}$ (siehe LQRI);
2. Finde L : lös ein standard LQR-Problem mit $\{A^\top, C^\top, Q = \bar{B} \cdot \bar{B}^\top, R = q \cdot \mathbb{I}\}$.

Das resultierende System ist gegeben durch:

- Open Loop:

$$\begin{aligned} \frac{d}{dt} \begin{bmatrix} x \\ \hat{x} \\ v \end{bmatrix} &= \begin{bmatrix} A & -B_u \cdot K & B_u \cdot K_I \\ 0 & A - B_u \cdot K - L \cdot C & B_u \cdot K_I \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ \hat{x} \\ v \end{bmatrix} + \begin{bmatrix} 0 \\ -L \\ I \end{bmatrix} \cdot e \\ y &= [C \ 0 \ 0] \cdot \begin{bmatrix} x \\ \hat{x} \\ v \end{bmatrix}. \end{aligned} \quad (1.17)$$

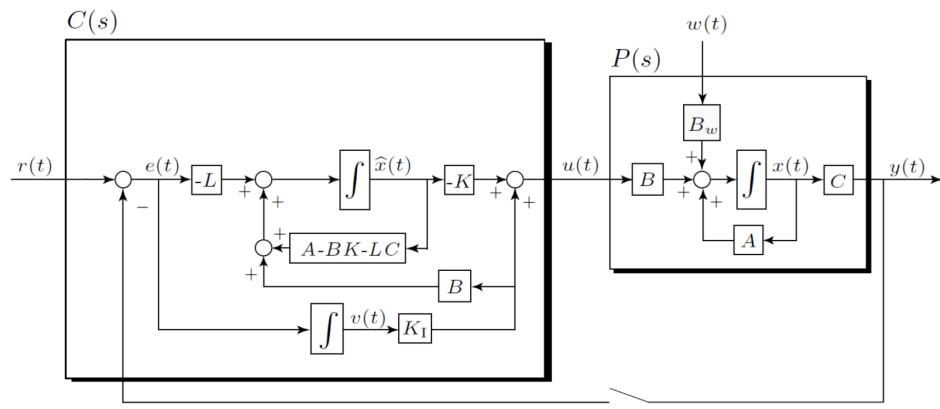


Abbildung 3: Struktur des LQGI Reglers.

- Closed Loop:

$$\frac{d}{dt} \begin{bmatrix} x \\ \hat{x} \\ v \end{bmatrix} = \begin{bmatrix} A & -B_u \cdot K & B_u \cdot K_I \\ L \cdot C & A - B_u \cdot K - L \cdot C & B_u \cdot K_I \\ -C & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x \\ \hat{x} \\ v \end{bmatrix} + \begin{bmatrix} B_u \\ B_u \\ I \end{bmatrix} \cdot r. \quad (1.18)$$

2 Numerische Optimierung

Der optimale Input $u^*(t \in [0, T])$ wird numerisch für das ganze “planning window” offline berechnet und als feedforward Signal benutzt, d.h $u_{ff} = u^*(t)$.

$$\begin{aligned} \min_{u(t)} J_T(x_0, u(t)) &= \min_{u(t)} \int_0^T \underbrace{l(x(t), u(t))}_{\text{Stage cost}} dt + \underbrace{m(x(T))}_{\text{Terminal cost}} \\ \text{s.t. } &\dot{x}(t) = f(t, x(t), u(t)), \\ &x(0) = x_0, \\ &x(t) \in \mathcal{X}, u(t) \in \mathcal{U}, \\ &x(T) \in \mathcal{X}_f, \end{aligned} \quad (2.1)$$

wobei

- $\dot{x}(t) = f(t, x(t), u(t))$: Modell des Systems;
- $x(0) = x_0$: Anfangsbedingung;
- $x(t) \in \mathcal{X}, u(t) \in \mathcal{U}$: “state and input constraint sets”;
- $x(T) \in \mathcal{X}_f$: “terminal state constraint set”.

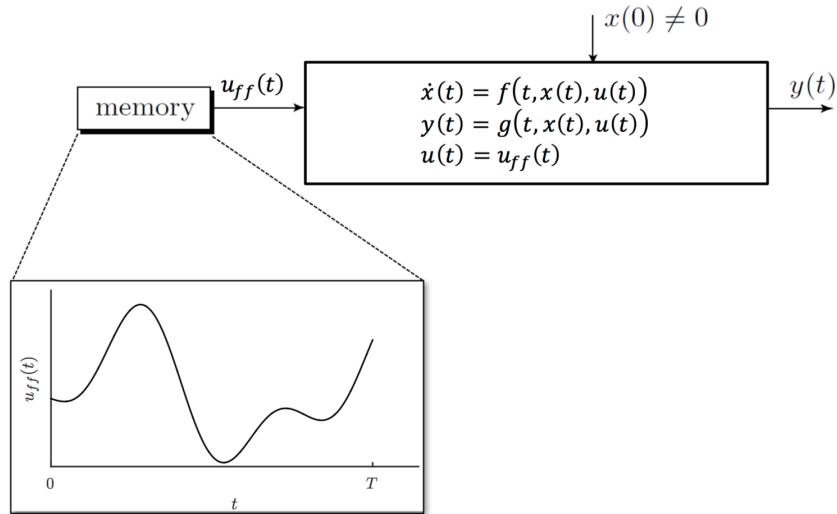


Abbildung 4: Numerische Optimierung.

3 MPC

3.1 Vorgehen

Mit Model Predictive Control löst man in jedem Zeitintervall die numerische Lösung des Optimierungsproblems, so dass ein Feedback ins System eingeführt wird. Alle Δt folgendes Vorgehen wird wiederholt:

1. Messe/Estimate aktuellen Zustand:

$$x(t) = z. \quad (3.1)$$

2. Finde den optimalen² Input für die ganze “planning window” T :

$$u^*([0, T]) = \arg \min_{u([0, T])} J_T(z, u([0, T])), \quad (3.2)$$

mit den in (2.1) eigenförmigen Modell und Constraints.

3. Implementiere nur den ersten Teil $u^*([0, \Delta t])$.

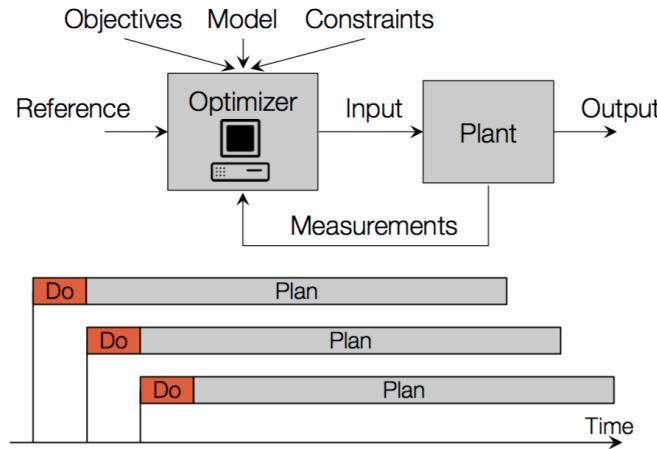


Abbildung 5: MPC.

3.2 Bemerkung zum MPC

Als Vorteil von MPC kann man mit allen Modellen arbeiten (nichtlinear, Totzeiten,...) und das Gütekriterium (objective function) kann auch frei gewählt werden (quadratischer Fehler, absoluter Fehler, maximaler Fehler, economic cost,...). Die Nachteile sind die nichtgarantierte Stabilität (die muss überprüft werden), die benötigte Genauigkeit des Modells, “computationally demanding” und “feasibility”.

3.3 Tipps zu der Serie

- Die Lösung des LQR könnte hilfreich sein:

$$\dot{x} = (A - B \cdot K) \cdot x, \quad (3.3)$$

mit K aus Übung 8.

- Kann man (3.3) analytisch lösen? Wie lautet der Input dann?

² $\arg \min f(x)$ ist der Argument, der $f(x)$ minimiert. Zum Beispiel $\arg \min x^2$ ist $x = 0$, da für $x = 0$ ist $f(x) = x^2$ minimal.