Zusammenfassung Control Systems II

Sommersemester 07

Inhaltsverzeichnis

1 Reglerauslegung **2**
1.1 PID-Regler . . . 2
1.1.1 Allgemeines . . . 2
1.1.2 Closed-Form Cross-Over Specification . . . 2
1.1.3 Sollwertgewichtung . . . 2
1.1.4 Anti-Windup . . . 2
1.1.5 Aström-Hägglund-Regel . . . 3
1.1.6 Prädiktive PI-Regler . . . 3
1.1.7 Numerische Optimierung . . . 3
1.2 Spezifikationen im Frequenzbereich . . . 3
1.2.1 Peaking Limitations . . . 3
1.2.2 Multiplikative Spezifikationen der Sensitivität . . . 4
1.2.3 Modellunsicherheit . . . 4

2 MIMO-Systeme **6**
2.1 Linearisierung . . . 6
2.2 Steuerbarkeit . . . 6
2.3 Beobachtbarkeit . . . 6
2.4 Lyapunov-Stabilität . . . 6
2.5 Transformation in Frequenzbereich . . . 6
2.6 Nyquist . . . 6
2.7 Pole und Nullstellen . . . 7
2.7.1 Pole . . . 7
2.7.2 Nullstellen . . . 7
2.7.3 Richtung . . . 7
2.8 Relative-Gain Array RGA . . . 7
2.8.1 2×2 System . . . 7
2.8.2 Allgemein . . . 7
2.9 Singularwerte . . . 7
2.10 Frequenzantworten von MIMO-Systemen . . . 8

3 Synthese von MIMO Regelsystemen **8**
3.1 Zustandsvektor-Rückführung mit LQR . . . 8
3.1.1 Optimierungsproblem . . . 8
3.1.2 Matrix-Riccati Gleichung . . . 8
3.1.3 Lösung des Optimierungs-Problems . . . 9
3.1.4 Matlab . . . 9
3.2 Erweiterungen von LQR . . . 9
3.2.1 LQRI . . . 9
3.2.2 Finite Horizon LQR . . . 9
3.2.3 LQR Feedforward Control Systems . . . 9
3.3 Eigenschaften von LQR Kontroller . . . 10
3.4 Zustandsbeobachter . . . 10
3.5 Zustandsrückführung mit Zustandsbeobachter - LQG . . . 11
3.5.1 Matlab . . . 11
3.6 Erweiterte LQG Controller . . . 11
3.6.1 LQG Reference Tracking Controllers . . . 11
3.6.2 LQG Controller mit Integral Action - LQGI . . . 11
3.7 Robustness Recovery - LTR . . . 11
3.7.1 naiver Ansatz . . . 11
3.7.2 Reguläre LTR Methode . . . 12

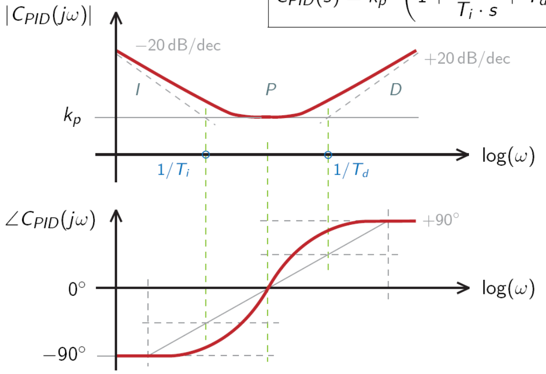
4 MATLAB **12**

1 Reglerauslegung

1.1 PID-Regler

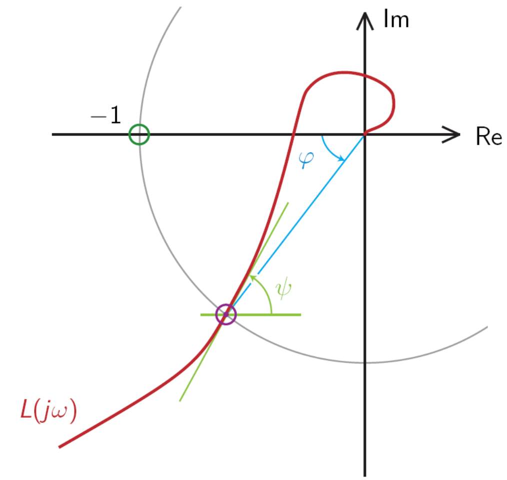
1.1.1 Allgemeines

$$\boxed{C_{PID}(s) = k_p \cdot \left(1 + \frac{1}{T_i \cdot s} + T_d \cdot s\right)}$$



1.1.2 Closed-Form Cross-Over Specification

Ziel dieses Verfahrens ist es, bei bekanntem $P(s)$ die Werte k_p, T_i und T_d so zu bestimmen, dass gewünschtes ω_c, φ und ψ erfüllt sind.



ω_c, φ und ψ können gemäss Grafik frei bestimmt werden.

r_P = Betrag von $P(j\omega_c)$
φ_P = Phase von $P(j\omega_c)$
(von positiver reeller Achse aus!)
$r'_P = \frac{\partial r_P(\omega)}{\partial \omega}\,|_{\omega=\omega_c}=$ Ableitung von r_P
$\varphi'_P = \frac{\partial \varphi_P(\omega)}{\partial \omega}\,|_{\omega=\omega_c}=$ Ableitung von φ_P

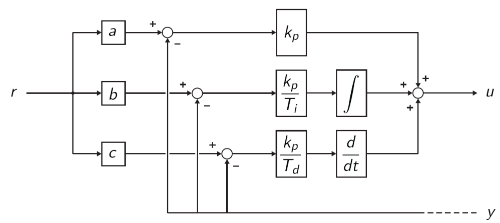
Beachte: $P(j\omega_c) = r_P \cdot e^{j \cdot \varphi_P}$
Bsp: $P = a + i \cdot b$, $r_P = \sqrt{a^2+b^2}$
$\varphi_P = \frac{\pi}{2} \cdot sign(b) - arctan\left(\frac{a}{b}\right)$
Bsp: $P = \frac{1}{a+i \cdot b}$, $r_P = \frac{1}{\sqrt{a^2+b^2}}$
$\varphi_P = arctan\left(\frac{a}{b}\right) - \frac{\pi}{2} \cdot sign(b)$

$\boxed{k_p = \frac{-1}{r_P}\cos(\varphi - \varphi_P)}$; $\boxed{T_i = \frac{1}{T_d \cdot \omega_c^2 - \tan(\varphi - \varphi_P) \cdot \omega_c}}$; $\frac{d}{dx} arctan(x) = \frac{1}{1+x^2}$

$$\boxed{T_d = \frac{1}{2} \cdot [\tan(\psi - \varphi_P) \cdot (\frac{r'_P}{r_P} - \varphi'_P \cdot \tan(\varphi - \varphi_P)) + \tan(\varphi - \varphi_P) \cdot (\frac{1}{\omega_c} - \frac{r'_P}{r_P}) - \varphi'_P]}$$

Anmerkungen: Stabilität ist mit diesem Verfahren nicht garantiert, das muss mit Nyquist oder einem anderen Verfahren überprüft werden.

1.1.3 Sollwertgewichtung

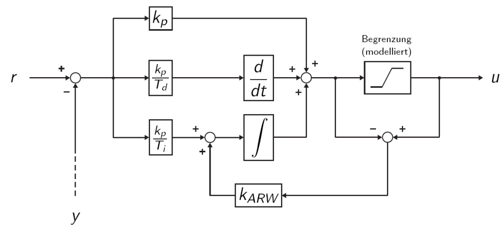


Um das Regelverhalten des PID-Reglers zu verbessern, wird dieser gemäss Grafik erweitert. Wird $r(t)$ nun verändert, hat das unterschiedliche einflüsse auf den P-, I- und den D-Teil des Reglers. Meist wird $a < 1$, $b = 1$ und $c = 0$ gewählt.

$b = 1$: So wird gewährleistet, dass der Steady-State Fehler des geschlossenen Regelsystems 0 wird, also $\lim_{t\to\infty} e(t) = 0$
$c = 0$: Verhindert ungewolltes Störungsverhalten des geschlossenen Regelsystems, wenn sich $r(t)$ ändert. Konkret: wird $r(t)$ sprunghaft verändert, für der D-Teil zu einem sehr grossem Fehlersignal (Ableitung eines Impuls $\rightarrow$ böse)

1.1.4 Anti-Windup

Die meisten Aktuatoren eines Systems weisen eine Begrenzung auf. Das hat zur Folge, dass lediglich Signale $u_{min} < u(t) < u_{max}$ weitergeleitet werden. Bei grossen Störungen oder grossen Änderungen der Führungsgrössen werden Stellsignale ausserhalb der Begrenzung verlangt. Das Regelsystem arbeitet dann „open loop“. Der Regelfehler wird durch den Integrator des Reglers jedoch weiter integriert. Der zu grosse I-Anteil muss dann nach Umkehrung des Wirkungssinns zuerst wieder abgebaut werden, was zu starkem Überschwingen führt.



Mit dieser Korrektur kann dieser Effekt umgangen werden. $k_{ARW} > 0$ muss dabei experimentell bestimmt werden. Ein grosses k_{ARW} führt üblicherweise zu weniger Überschwingen, aber auch zu kleinerem t_{90}.

1.1.5 Aström-Hägglund-Regel

Ähnlich wie das Ziegler-Nichols-Verfahren dient die AHR zur PID-Reglerauslegung (und auch PI) bei unbekannter Strecke $P(s)$. Man erreicht damit häufig eine bessere Performance als mit ZNV, jedoch kann keine Garantie für Stabilität und Robustheit gewährleistet werden.

Vorgehen:

1. I- und D-Anteil des Reglers ausschalten ($T_i = \infty$, $T_d = 0$)
2. k_p solange erhöhen, bis das System grenzstabil wird
3. kritische Verstärkung k_p^* und kritische Periode $T^* = \frac{2\pi}{\omega^*}$ ablesen
4. Regler-Parameter gemäss Tabelle festlegen

Des weiteren muss der DC-Gain der Strecke, also $|P(0)|$, bekannt sein. Der kritische Abstand μ kann frei gewählt werden, wobei das Regelverhalten durch die Wahl stark beeinflusst wird. $\mu = .5$ ist ein aggressives Design, bei dem Störungen schneller ausgeglichen werden. $\mu = .7$ ist hingegen ein robusteres Design.

$\boxed{x = \alpha_{0,x} \cdot e^{\alpha_{1,x}\cdot\kappa + \alpha_{2,x}\cdot\kappa^2}}$, mit $\kappa = \frac{1}{|P(0)|\cdot k_p^*}$ und $x = \{\frac{k_p}{k_p^*}, \frac{T_i}{T^*}, \frac{T_d}{T^*}, a\}$

Für den **PI-Regler** gelten folgende Beziehungen:

	$\mu = 0.7$			$\mu = 0.5$		
x	α_0	α_1	α_2	α_0	α_1	α_2
k_p/k_p^*	0.053	2.9000	-2.6	0.13	1.9	-1.30
T_i/T^*	0.900	-4.4000	2.7	0.90	-4.4	2.70
a	1.100	-0.0061	1.8	0.48	0.4	-0.17

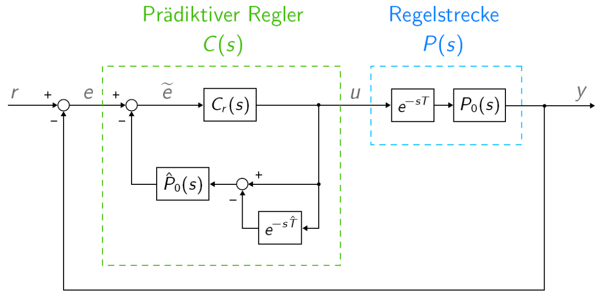
Für den **PID-Regler** werden folgende Werte verwendet:

	$\mu = 0.7$			$\mu = 0.5$		
x	α_0	α_1	α_2	α_0	α_1	α_2
k_p/k_p^*	0.33	-0.31	-1.00	0.72	-1.60	1.20
T_i/T^*	0.76	-1.60	-0.36	0.59	-1.30	0.38
T_d/T^*	0.17	-0.46	-2.10	0.15	-1.40	0.56
a	0.58	-1.30	3.50	0.25	0.56	-1.20

1.1.6 Prädiktive PI-Regler

Ist im System eine Totzeit vorhanden, die nicht vernachlässigbar ist, sind übliche PID-Regler ungeeignet, mach weicht auf prädiktive Regler aus. Die Totzeit lässt sich aber natürlich damit nicht beheben. Verfahren nach Smith:

Üblicherweise wird ein PT1-Element gewählt, $T(s) = \frac{k}{\tau s+1}e^{-Ts}$, die Totzeit muss die gleiche sein wie die von der Strecke, die Zeitkonstante τ kann frei gewählt werden. $k = 1$ ist üblicherweise der Fall, ansonsten hat man einen statischen Nachlauffehler.



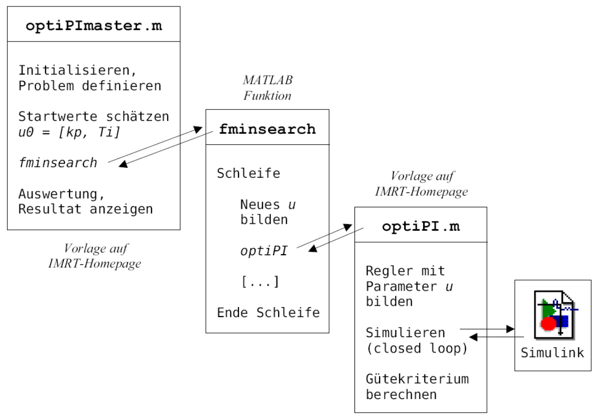
Ausgangslage:
approximierte Strecke:
$P(s) = P_0(s) \cdot e^{-sT}$
Spezifikation für das Regelsystem:
$T(s) = T_0(s) \cdot e^{-sT}$
$T(s) = \frac{L(s)}{1+L(s)} = \frac{C(s)P(s)}{1+C(s)P(s)}$
$\Rightarrow \boxed{C(s) = \frac{1}{P(s)} \cdot \frac{T(s)}{1-T(s)}}$

1.1.7 Numerische Optimierung

Anhand Berechnungen wird eine Lösung gefunden, die im besten Fall ein lokales Optimum garantiert, aber keine optimale Lösung im ganzeinheitlichen Sinn.
Beispiel für ein Gütekriterium:

$$\boxed{J = \mu_1 \cdot \int_0^\infty e^2(t)dt + \mu_2 \cdot \max_t(y(t) - 1) + \mu_3 \cdot (1 - \min_\omega(|1 + L(j\omega)|))}$$

1. Term: Sollwertabweichung, 2. Term: Überschwingen, 3. Term: Robustheit
In Matlab realisiert:



1.2 Spezifikationen im Frequenzbereich

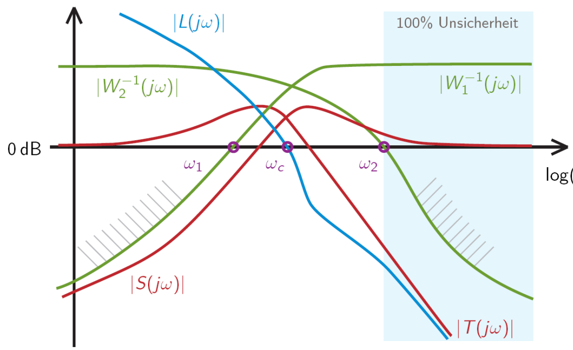
1.2.1 Peaking Limitations

Im Bereich der Durchtrittsfrequenz ω_c sind $|S(j\omega)|$ und $|T(j\omega)|$ meist grösser als 1. Damit die resultierende Verstärkung von Störungen und Rauschen eingeschränkt werden

kann, werden folgende Spezifikationen vorgegeben:
$||S||_\infty < S_{max}$, $||T||_\infty < T_{max}$, $S_{max}, T_{max} > 1$
Daraus lassen sich garantierte Schranken für die Verstärkungs- und Phasenreserve ableiten:

$$\boxed{\min\{1-\frac{1}{T_{max}}, \frac{S_{max}}{S_{max}+1}\} < \gamma < \max\{1+\frac{1}{T_{max}}, \frac{S_{max}}{S_{max}-1}\}}$$

$$\boxed{\varphi = 2 \cdot \arcsin\left(\frac{1}{2 \cdot \min\{T_{max}, S_{max}\}}\right)}$$



Üblicherweise wird $\left|W_{1,2}^{-1}(j\omega)\right|$ im Bodeplot aufgetragen, so lässt sich besser erkennen, ob die Bedingungen wirklich erfüllt sind, d.h. ob $|T(j\omega)|$ immer unter $\left|W_2^{-1}(j\omega)\right|$ und $|S(j\omega)|$ immer unter $\left|W_1^{-1}(j\omega)\right|$ ist.

1.2.2 Multiplikative Spezifikationen der Sensitivität

Nominelle Regelgüte Für $|S(j\omega)|$ kann mit $|W_1(j\omega)|$ eine obere Schranke gewählt werden, die für alle Frequenzen ω erfüllt sein muss.

$$\boxed{|S(j\omega)| \cdot |W_1(j\omega)| < 1, \forall\omega} \Leftrightarrow \boxed{|W_1(j\omega)| < |1+L(j\omega)|, \forall\omega}$$

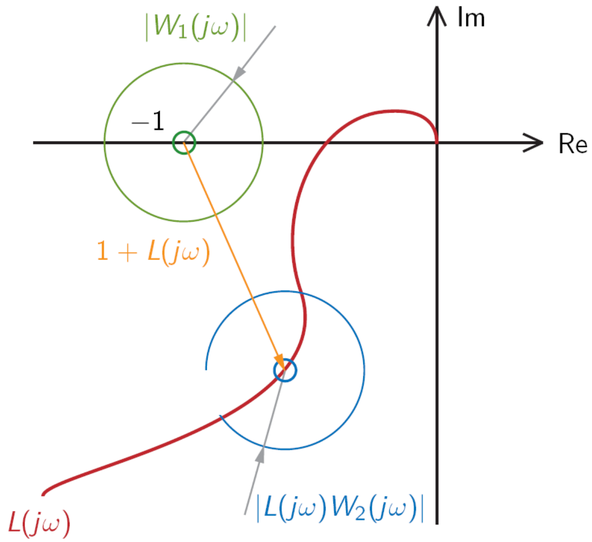
Robuste Stabilität Durch die Modellunsicherheit $|W_2(j\omega)|$ ist eine Schranke für $|T(j\omega)|$ gegeben.

$$\boxed{|T(j\omega)| \cdot |W_2(j\omega)| < 1, \forall\omega} \Leftrightarrow \boxed{|L(j\omega)| \cdot |W_2(j\omega)| < |1+L(j\omega)|, \forall\omega}$$

Robuste Regelgüte

$$\boxed{|W_1(j\omega) \cdot S(j\omega)| + |W_2(j\omega) \cdot T(j\omega)| < 1}$$

$$\Leftrightarrow \boxed{|W_1(j\omega)| + |W_2(j\omega) \cdot L(j\omega)| < |1+L(j\omega)|}$$



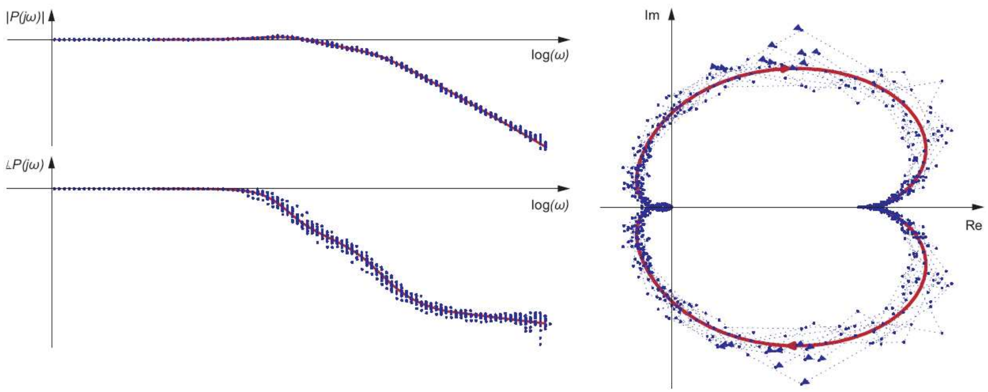
Anforderungen an W_1:

- muss für hohe Frequenzen < 1 sein
- bei kleinen Frequenzen > 1 notwendig (Störungsunterdrückung)
- Durchtrittsfrequenz $2 \cdot \omega_{W_1} < \omega_c$

1.2.3 Modellunsicherheit

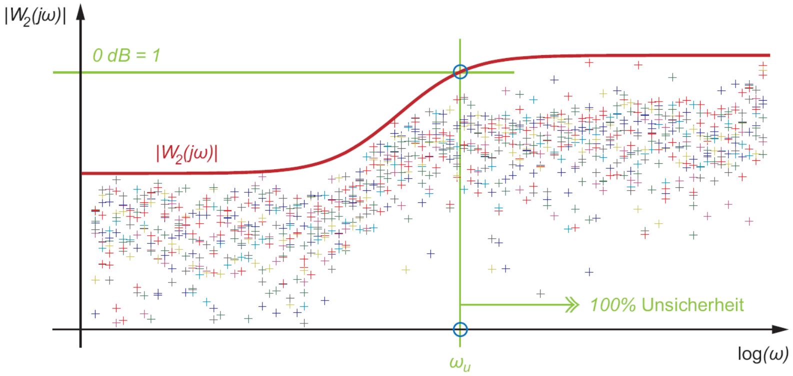
Ist die Übertragungsfunktion eines Systems unbekannt, lässt sich nur approximiert bestimmen. Es ist notwendig, eine Unsicherheit $W_2(s)$ zu bestimmen. So kann garantiert werden, dass die Strecke $P_t(s)$ in einem bestimmten Bereich ist:
$P_t(s) \in S = \{P(s) \cdot (1 + W_2(s)\Delta(s)) \mid ||\Delta(s)|| \leq 1, \arg\{\Delta(s)\} \in [-\pi, \pi]\}$

Unsicherheitsbestimmung anhand von Messdaten Notwendig sind K Messungen mit je I Werten: $\omega \in \mathbb{R}^I$, $M \in \mathbb{R}^{I \times K}$, $\Phi \in \mathbb{R}^{I \times K}$

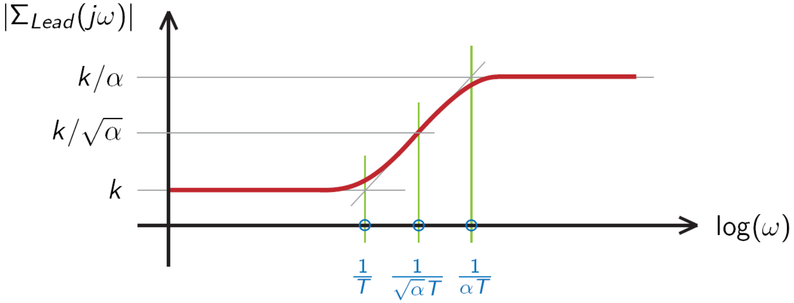


Vorgehen:

1. Schätzung der Unsicherheit: $\left|\frac{m_{i,k}\cdot e^{j\cdot\varphi_{i,k}}}{m_i\cdot e^{j\cdot\varphi_i}}-1\right|<|W_2(j\omega)|$
2. Unsicherheitsschranke $|W_2(j\omega)|$ iterativ bestimmen:



Die Kurve $|W_2(j\omega)|$ wird anhand von Lead-Elementen bestimmt. Es sollten der Einfachheit wegen so wenig Elemente wie möglich verwendet werden.



$$\boxed{\Sigma_{Lead}(s)=k\cdot\frac{T\cdot s+1}{\alpha\cdot T\cdot s+1}, 0<\alpha<1}$$

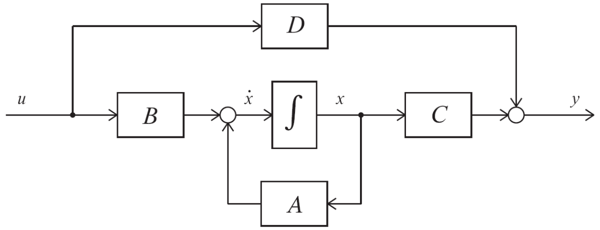
1. $W_{2,0}(j\omega)=k_0$
2. $W_{2,1}(j\omega)=W_{2,0}(j\omega)\cdot\frac{T_1\cdot s+1}{\alpha_1\cdot T_1\cdot s+1}$
3. $W_{2,i}(j\omega)=W_{2,i-1}(j\omega)\cdot\frac{T_i\cdot s+1}{\alpha_i\cdot T_i\cdot s+1}$

2 MIMO-Systeme

Die Systemmodellierung von MIMO-Systemen (m Eingänge, p Ausgänge, n Zustandsvariablen) erfolgt analog zu den SISO-Systemen, einzig die Matrizen-Grössen ändern sich. Linearisierung, Normierung und Transformation in den Frequenzbereich wird gleich gehandhabt.

2.1 Linearisierung

Betriebspunkt ist allgemein $x = x_0$ und $u = u_0$. Oft ist dies der Gleichgewichtspunkt.



$$\begin{aligned} \tfrac{d}{dt}x(t) &= A \cdot x + B \cdot u \\ y(t) &= C \cdot x + D \cdot u \end{aligned}$$

$$x(t) \in \mathbb{R}^n$$
$$u(t) \in \mathbb{R}^m$$
$$y(t) \in \mathbb{R}^p$$

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_{\substack{x=x_0,\\ u=u_0}} \in \mathbb{R}^{n \times n}, \quad B = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial u_1} & \cdots & \frac{\partial f_n}{\partial u_m} \end{bmatrix}_{\substack{x=x_0,\\ u=u_0}} \in \mathbb{R}^{n \times m},$$

$$C = \begin{bmatrix} \frac{\partial g_1}{\partial x_1} & \cdots & \frac{\partial g_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_p}{\partial x_1} & \cdots & \frac{\partial g_p}{\partial x_n} \end{bmatrix}_{\substack{x=x_0,\\ u=u_0}} \in \mathbb{R}^{p \times n}, \quad D = \begin{bmatrix} \frac{\partial g_1}{\partial u_1} & \cdots & \frac{\partial g_1}{\partial u_m} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_p}{\partial u_1} & \cdots & \frac{\partial g_p}{\partial u_m} \end{bmatrix}_{\substack{x=x_0,\\ u=u_0}} \in \mathbb{R}^{p \times m}$$

2.2 Steuerbarkeit

Ein System ist genau dann vollständig steuerbar, wenn die Steuerbarkeitsmatrix $\mathcal{R}_n$ vollen Rang hat.

$$\boxed{\mathcal{R}_n = \begin{pmatrix} B & A \cdot B & A^2 \cdot B & \cdots & A^{n-1} \cdot B \end{pmatrix} \in \mathbb{R}^{n \times (n \cdot m)}}$$

$Rang(\mathcal{R}_n) = n \Leftrightarrow \det(\mathcal{R}_n) \neq 0 \Longrightarrow$ System vollständig steuerbar

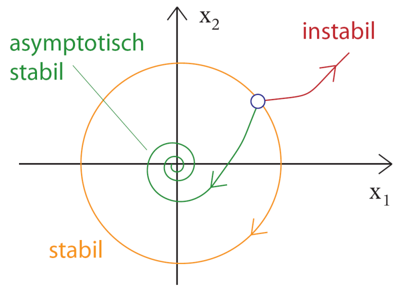
2.3 Beobachtbarkeit

Ein System ist genau dann vollständig beobachtbar, wenn die Beobachtbarkeitsmatrix $\mathcal{O}_n$ vollen Rang hat.

$$\mathcal{O}_n = \begin{pmatrix} C \\ C \cdot A \\ C \cdot A^2 \\ \cdots \\ C \cdot A^{n-1} \end{pmatrix} \in \mathbb{R}^{(m \cdot n) \times n}$$

$Rang(\mathcal{O}_n) = n \Leftrightarrow \det(\mathcal{O}_n) \neq 0 \Longrightarrow$ System vollständig beobachtbar

2.4 Lyapunov-Stabilität



λ_i sind die Eigenwerte der Matrix A:

$Re(\lambda_i) \leq 0, \forall i \Rightarrow$ **stabil**
Ausname: mehrfache EW mit $Re(\lambda_i) = 0$ und
A nicht diagonalisierbar $\Rightarrow$ **instabil**
$Re(\lambda_i) < 0, \forall i \Rightarrow$ **asymptotisch stabil**
$Re(\lambda_i) > 0$, für mindestens ein $\lambda_i \Rightarrow$ **instabil**

2.5 Transformation in Frequenzbereich

Frobeniusform oder andere Vereinfachungen sind nicht mehr möglich.

$$\boxed{P(s) = C \cdot (sI - A)^{-1} \cdot B + D = \frac{C \cdot Adj(sI - A) \cdot B}{det(sI - A)} + D}$$

$$P(s) = \begin{bmatrix} P_{1,1}(s) & P_{1,2}(s) & \cdots & P_{1,m}(s) \\ P_{2,1}(s) & P_{2,2}(s) & \cdots & P_{2,m}(s) \\ \vdots & \vdots & \ddots & \vdots \\ P_{p,1}(s) & P_{p,2}(s) & \cdots & P_{p,m}(s) \end{bmatrix}$$

2.6 Nyquist

Das Nyquist-Theorem kann auch für MIMO-System angewendet werden:
Das Closed-Loop System ist genau dann und nur dann stabil, wenn der Nyquistplot

$\mathcal{N} = \det[I + P(j\omega) \cdot C(j\omega)]$, $\omega \in [-\infty, +\infty]$ **den Ursprung (!) genau $\rho + \sigma/2$ mal im Gegenuhrzeigersinn umrundet.**
Allerdings ist das Theorem für MIMO-Systeme nicht so mächtig wie für SISO-Systeme. Das liegt daran, dass das Berechnen der Determinanten Informationen „zerstört“, die den Zusammenhang zwischen den einzelnen Ein- und Ausgängen beschreiben. Das System kann nun zwar stabil sein, aber eine schlechte Performance haben.
Für diagonal-dominante Systeme ist dieses Theorem aber durchaus ganz gut anwendbar.

2.7 Pole und Nullstellen

Minoren: Die Minoren einer Matrix sind alle einzelnen Einträge sowie alle möglichen „Determinanten“. Beispiel für eine 2×3-Matrix:
$\begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix} \Rightarrow$ Minoren: $a, b, c, d, e, f, a \cdot e - b \cdot d, a \cdot f - c \cdot d, b \cdot f - c \cdot e$
maximale Minoren: Diejenigen, an denen am meisten Matrix-Einträge beteiligt sind, im Falle obigen Beispieles die letzten 3.

2.7.1 Pole

Die Pole von $P(s)$ sind die Nullstellen des kleinsten gemeinsamen Vielfachen der Divisoren aller Minoren von $P(s)$
Die Ordnung des Systems minimaler Ordnung entspricht also auch der Anzahl Pole. Beachte: Alle Pole sind Pole der einzelnen Übertragungsfunktionen in $P(s)$, aber dennoch lassen sich die Pole nicht direkt ablesen, weil die Vielfachheit anders ist.

2.7.2 Nullstellen

Die Nullstellen von $P(s)$ sind die Nullstellen des grössten gemeinsamen Teilers der Zähler aller maximalen Minoren von $P(s)$, wobei die Minoren so normiert sind, dass sie denselben Divisoren haben wie das Pol-Polynom bei der Pol-Berechnung
Ist $P(s)$ quadratisch, entsprechen die Nullstellen den Polen von $P^{-1}(s)$

2.7.3 Richtung

Es können Pole und Nullstellen bei derselben Frequenz auftreten, die aber nicht gekürzt werden dürfen. Kürzung tritt nur auf, falls die Richtung dieselbe ist.
Richtung von Nullstellen: $P(s)\mid_{s=\zeta_i} \cdot \delta_{\zeta,i}^{in} = 0 \cdot \delta_{\zeta,i}^{out}$
Richtung von Polen: $P(s)\mid_{s=\pi_i} \cdot \delta_{\pi,i}^{in} = \infty \cdot \delta_{\pi,i}^{out}$

2.8 Relative-Gain Array RGA

RGA dient zur Analyse, ob das $m \times m$ MIMO-System über m SISO-Controller geregelt werden kann, oder ob wirklich MIMO-Regelung unabdingbar ist.
diagonal dominant: Ein System ist dd, falls $RGA(s) \approx I$
Folge: dd-Systeme können wie MIMO-Systeme behandelt werden
Plottet man die RGA-Einträge abhängig von s, lassen sich Bereiche ablesen, in denen das System annähernd diagonal dominant ist. Wählt man die Durchtrittsfrequenz in diesem Bereich, kann ein SISO-Ansatz verwendet werden.

2.8.1 2×2 System

Für solche Systeme lässt sich die RGA-Matrix wie folgt anschaulich herleiten:
$u_1 \to y_1$: $y_1 = \frac{P_{11}(1+P_{22}C_{22}) - P_{12}C_{22}P_{21}}{1+P_{22}C_{22}} \cdot u_1$

- open-loop ($C_{22} \approx 0$): $y_1 = P_{11} \cdot u_1$
- closed-loop ($P_{22}C_{cc} \gg 1$): $y_1 = \frac{P_{11}P_{22} - P_{12}P_{21}}{P_{22}}$

Das Element $(1,1)$ der RGA-Matrix ist definiert als das Verhältnis der open-loop zur closed-loop Übertragungsfunktion:

$$\boxed{[RGA]_{11} = P_{11} / \left[\frac{P_{11}P_{22} - P_{12}P_{21}}{P_{22}}\right] = \frac{P_{11}P_{22}}{P_{11}P_{22} - P_{12}P_{21}}}$$

Analog dazu $u_1 \to y_2$:

$$\boxed{[RGA]_{12} = \frac{-P_{12}P_{21}}{P_{11}P_{22} - P_{12}P_{21}} = 1 - [RGA]_{11}}$$

Es lässt sich leicht zeigen, dass $[RGA]_{22} = [RGA]_{11}$ und $[RGA]_{12} = [RGA]_{21}$
Damit die $P(s)$ diagonal dominant ist, muss $[RGA]_{12}$ klein sein und darum $P_{12}P_{21} \ll P_{11}P_{22}$

2.8.2 Allgemein

$\boxed{[RGA](s) = P(s). \times (P^{-1})^T}$ ($. \times$ bedeutet elementweise Multiplikation)
Beobachtungen:

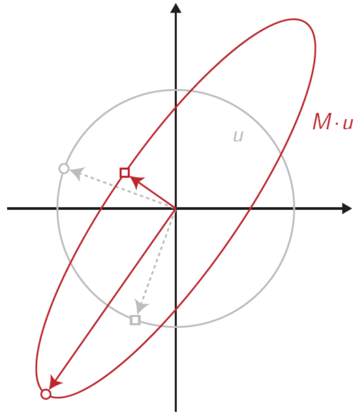
- Sowohl Reihen als auch Spalten von RGA aufsummiert ergeben 1
- RGA einer Rechts- oder Links-Dreieckmatrix ist die Einheitsmatrix

2.9 Singularwerte

$\sigma_i = \sqrt{\lambda_i}$, λ_i sind die Eigenwerte von $\bar{M}^T \cdot M$
Meist sind nur der maximale und der minimale Singularwert interessant, damit lassen sich Worst-Case-Abschätzungen machen.

Für die induzierte (euklidische) Norm $||M||$ der Gleichung $y = M \cdot u$ gilt also:
$||M|| = \max_{||u|| \neq 0} \frac{||y||}{||u||} = \max_{||u||=1} ||M(u)|| = \max_i \{\sigma_i\}$

Geometrische Interpretation in $\mathbb{R}^2$:



Kreis u wird auf Ellipse $y = M \cdot u$ abgebildet, wobei maximale und minimale Verstärkung aus den Hauptachsen qualitativ ersichtlich sind.
$\sigma_{\min}\{M\} \leq \frac{||y||}{||u||} \leq \sigma_{max}\{M\}$

2.10 Frequenzantworten von MIMO-Systemen

Eingangsvektor: $u(t)$, Ausgangsvektor: $y(t)$

$$u(t) = \begin{pmatrix} \mu_1 \cdot \cos(\omega t + \varphi_1) \cdot h(t) \\ \mu_2 \cdot \cos(\omega t + \varphi_2) \cdot h(t) \\ \dots \\ \mu_m \cdot \cos(\omega t + \varphi_m) \cdot h(t) \end{pmatrix}, \; y(t) = \begin{pmatrix} \nu_1 \cdot \cos(\omega t + \psi_1) \cdot h(t) \\ \nu_2 \cdot \cos(\omega t + \psi_2) \cdot h(t) \\ \dots \\ \nu_m \cdot \cos(\omega t + \psi_m) \cdot h(t) \end{pmatrix}$$

Phasor: $\boxed{e^{j\psi} \cdot \nu = P(j \cdot \omega) \cdot e^{j\phi} \cdot \mu}$, $\phi = \text{diag}(\varphi_i)$, $\psi = \text{diag}(\psi_i)$

Konservative Grenzen für den Phasor:

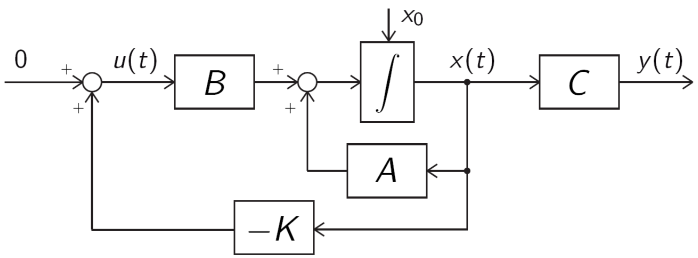
$\boxed{\min_i \sigma_i\{P(j \cdot \omega)\} \leq ||e^{j\psi} \cdot \nu|| \leq \max_i \sigma_i\{P(j\omega)\}}$, mit $||e^{j\phi} \cdot \mu|| = 1$

3 Synthese von MIMO Regelsystemen

Annahmen:

- $x(t)$ sei zugänglich
- $\hat{x}(t)$ wird abgeschätzt
- $u(t) = -K \cdot \hat{x}(t)$

3.1 Zustandsvektor-Rückführung mit LQR



Gegeben: $\dot{x}(t) = A \cdot x(t) + B \cdot u(t)$, $x(t) \in \mathbb{R}^n$, $u(t) \in \mathbb{R}^m$
Gesucht: $u(t) = -K \cdot x(t)$, also ein Regler mit Zustandsvektor-Rückführung

3.1.1 Optimierungsproblem

Daraus entsteht ein Optimierungsproblem: $\boxed{J(t) = \int_0^\infty (x^T \cdot Q \cdot x + u^T \cdot R \cdot u) dt}$

Häufige Wahl:

- $R = R^T > 0$, meist $R = r \cdot I_{m \times m}$
- $Q = Q^T = \tilde{C}^T \cdot \tilde{C} \geq 0$, im Allgemeinen $\tilde{C} \neq C$

Wahl der Gewichtungsmatrizen: mit geeigneter Wahl von Q lassen sich die Sollwertabweichungen der einzelnen Zustände gezielt *bestrafen*. Mit R lässt sich der zum Regeln benötigte Energieaufwand beeinflussen.

- $r \to 0$: *cheap control*, d.h. billige Steuerenergie, der Stellvektor wird nur schwach bestraft
- $r \to \infty$: *expensive control*, d.h. d.h. teure Steuerenergie, der Stellvektor wird möglichst stark bestraft

3.1.2 Matrix-Riccati Gleichung

$\boxed{\phi \cdot B \cdot R^{-1} \cdot B^T \cdot \phi - \phi \cdot A - A^T \cdot \phi - Q = 0}$, mit $\phi = \phi^T$

Weiter muss $\phi = \begin{pmatrix} \phi_1 & \phi_2 \\ \phi_2 & \phi_3 \end{pmatrix}$ positiv definit sein. Somit gibt es eine eindeutige Lösung für ϕ. Hilfreiche Bedingungen für eine 2x2-Matrix, die sich aus *positiv definit* ableiten:

- $\phi_1 > 0$
- $\phi_1 \cdot \phi_3 > \phi_2^2$
- und somit auch $\phi_3 > 0$

Hinreichende Bedingungen, damit eine Lösung für ϕ existiert:

- $\{A, B\}$ vollständig steuerbar
- $\{A, \tilde{C}\}$ vollständig beobachtbar

3.1.3 Lösung des Optimierungs-Problems

$\boxed{K = R^{-1} \cdot B^T \cdot \phi}$

Daraus resultiert: $\dot{x}(t) = (A - B \cdot K) \cdot x(t)$. Dabei ist $A - B \cdot K$ eine **Hurwitz-Matrix**, hat also alles Eigenwerte mit negativem Realteil.
Anmerkung: Der resultierende Regler ist optimal im Sinne des Gütekriteriums. Das bedeutet freilich nicht, dass es sich um den best möglichen Regler handelt!

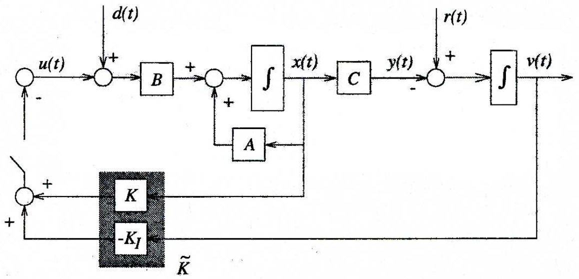
3.1.4 Matlab

- $K = lqr(A, B, Q, R)$

3.2 Erweiterungen von LQR

3.2.1 LQRI

Beim Standard LQR Ansatz ist kein Integrator vorhanden. Hat die Strecke selbst keine offenen Integratoren, müssen welche eingeführt werden über eine Erweiterung des LQR Ansatzes. Mit LQRI werden bei jedem Ausgang je ein Integrator angehängt.
Erweitertes System:



$\frac{d}{dt}\tilde{x}(t) = \tilde{A} \cdot \tilde{x}(t) + \tilde{B}_u \cdot u(t) + \tilde{B}_r \cdot r(t) + \tilde{B}_d \cdot d(t)$
mit $\tilde{x}(t) = \begin{bmatrix} x(t) \\ v(t) \end{bmatrix} \in \mathbb{R}^{n+m}$, $\tilde{A} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix}$, $\tilde{B}_u = \tilde{B}_d = \begin{bmatrix} B \\ 0 \end{bmatrix}$, $\tilde{B}_r = \begin{bmatrix} 0 \\ I \end{bmatrix}$

Nun kann wie bei LQR das Optimierungsproblem gelöst werden: (mit $\tilde{C} = \begin{bmatrix} C & 0 \\ 0 & \gamma \cdot I \end{bmatrix}$)

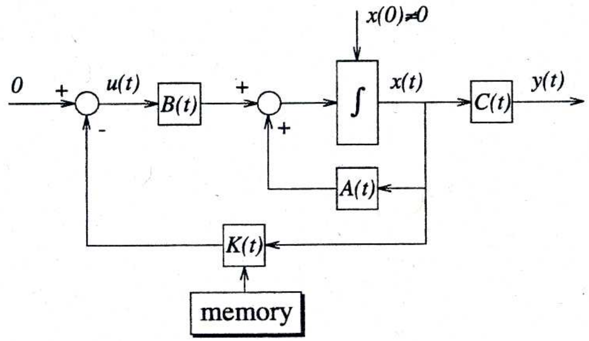
- $A \to \tilde{A}$
- $B \to \tilde{B}_u$
- $Q \to \tilde{C}^T \cdot \tilde{C}$
- $R \to R = r \cdot I$

Daraus folgt die Lösung $\tilde{K} = \begin{bmatrix} K & -K_I \end{bmatrix}$.
$\gamma > 0$ ist ein skalarer Tuning Parameter, mit dem das Verhalten des Integrators beeinflusst werden kann.

Matlab

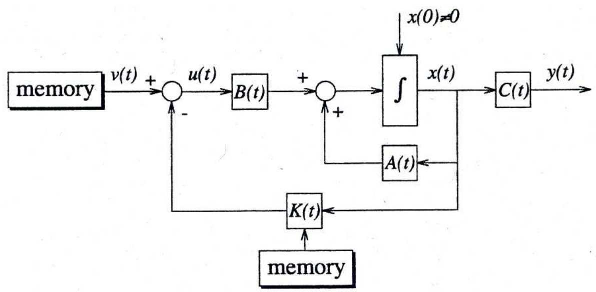
- $\tilde{K} = lqr(\tilde{A}, \tilde{B}_u, \tilde{C}' \cdot \tilde{C}, r \cdot eye(m, m))$

3.2.2 Finite Horizon LQR



Sind die Systemmatrizen zeitvariant, weicht man auf die Finite Horizon Methode aus.
Das Gütekriterium nimmt folgende Gestalt an:
$J(u) = x^T(t_b) \cdot P \cdot x(t_b) + \int_{t_a}^{t_b} [x^T(u(t)) \cdot Q(t) \cdot x(u(t)) + u^T(t) \cdot R(t) \cdot u(t)] dt$
wobei $P \in \mathbb{R}^{n \times n}$ und $P = P^T \geq 0$. Die Lösung dieses Optimierungsproblems lässt sich mit der differentiellen Matrix-Riccati Gleichung finden:
$\frac{d}{dt}\phi(t) = \phi(t) \cdot B(T) \cdot R^{-1}(t) \cdot B^T(t) \cdot \phi(t) - \phi(t) \cdot A(t) - A^T(t)\phi(t) - Q(t)$
mit der zusätzlichen End-Bedingung: $\phi(t_b) = P$
Daraus folgt dann analog: $K(t) = R^{-1}(t) \cdot B^T(t) \cdot \phi(t)$

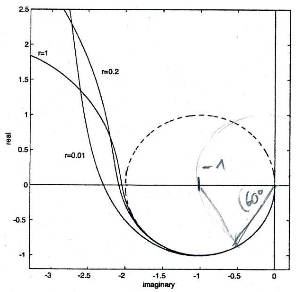
3.2.3 LQR Feedforward Control Systems



Dieser Ansatz wird verwendet, wenn der Ausgang $y(t)$ dem Referenzsignal $r(t)$ so genau wie möglich folgen und dabei ein Minimum an Kontroll-Energie verwendet werden soll.
Gütekriterium: $J(u) = [r(t_b) - y(u(t_b))]^T \cdot P \cdot [r(t_b) - y(u(t_b))] + ...$
$... + \int_{t_a}^{t_b} \{[r(t) - y(u(t))]^T \cdot Q(t) \cdot [r(t) - y(u(t))] + u^T(t) \cdot R(t) \cdot u(t)\} dt$

Optimum wiederum mit der differentiellen Matrix-Riccati Gleichung:
$\frac{d}{dt}\phi(t) = \phi(t)B(T)R^{-1}(t)B^T(t)\phi(t) - \phi(t)A(t) - A^T(t)\phi(t) - C^T(t)Q(t)C(t)$
mit der zusätzlichen Start-Bedingung: $\phi(t_b) = C^T(t_b) \cdot P \cdot C(t_b)$
Schliesslich: $K(t) = R^{-1}(t) \cdot B^T(t) \cdot \phi(t)$

3.3 Eigenschaften von LQR Kontroller

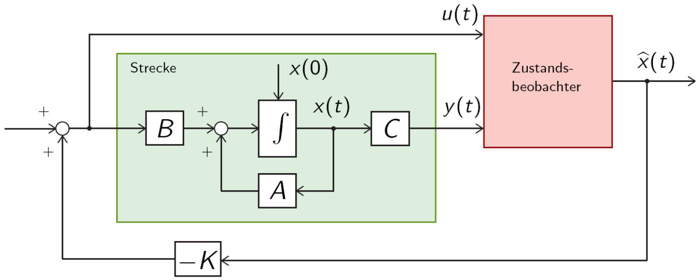


- $A - B \cdot K$ ist Hurwitz, garantiert also asymptotisch stabiles verhalten
- $\mu_{LQR} = min_\omega\{I + L_{LQR}(s)\} \geq 1$, d.h. gute Robustheit
- Verstärkungsreserve: $[0.5, \infty]$
- Phasenreserve: ± 60

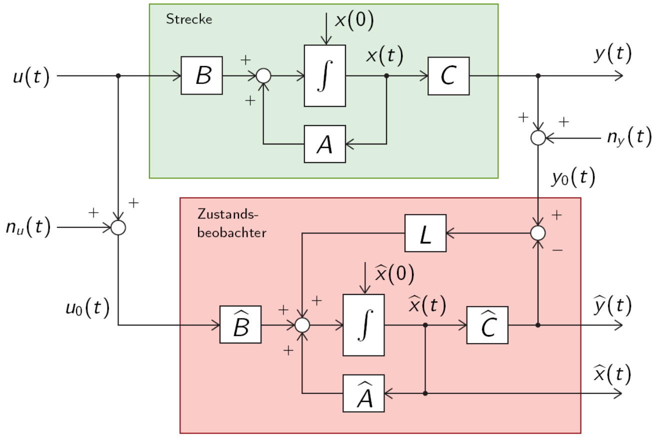
Robustheit kann noch verbessert werden durch einbringen eines Faktors $\beta > 1$:
$\frac{1}{\beta}\phi_\beta \cdot B \cdot R^{-1} \cdot B^T \cdot \phi_\beta - \phi_\beta \cdot A - A^T \cdot \phi_\beta - Q = 0$
Das führt zu $\mu_\beta = min_\omega\{\beta I + L(j\omega)\} \geq \beta$
Wahl von r in $R = r \cdot I$: mit $r \ll 1$ lassen sich Anfangsauslenkungen $x(0)$ sehr viel schneller regulieren als mit $r \gg 1$

3.4 Zustandsbeobachter

Problem der Zustandsvektor-Rückführung: In der Praxis sind die internen Zustände der Strecke meist nicht bekannt.
Lösung: Abschätzung der Zustände mit einem Zustandsbeobachter.



Umgeschrieben und ausführlicher:



Beobachter-Dynamik:
$\frac{d}{dt}\hat{x}(t) = \hat{A} \cdot \hat{x}(t) + \hat{B} \cdot u_0(t) + \hat{L} \cdot (y_0(t) - \hat{y}(t))$
$\hat{y}(t) = \hat{C} \cdot \hat{x}(t)$
Der Beobachter ist im wesentlichen eine Kopie der Strecke, somit: $A = \hat{A}$, $B = \hat{B}$ und $C = \hat{C}$. Wird $n_u(t) = n_y(t)$ vernachlässigt, sind $x(t)$ und $\hat{x}(t)$ gleich, vorausgesetzt $x(0) = \hat{x}(0)$. Dies ist aber allgemein nicht möglich.
Problembeschreibung: mit $\bar{x}(t) = x(t) - \hat{x}(t) \in \mathbb{R}^n$
$\frac{d}{dt}\bar{x}(t) = \frac{d}{dt}x(t) - \frac{d}{dt}\hat{x}(t) = ... = [A - L \cdot C] \cdot \bar{x}(t)$
Somit: Ist $A - L \cdot C$ eine Hurwitz-Matrix, wird der Fehler $\bar{x}(t)$ asymptotisch gegen 0 konvergieren. Die Geschwindigkeit der Konvergenz kann mit L beeinflusst werden.
Das Problem kann umgeschrieben werden: $A - L \cdot C \rightarrow (A - L \cdot C)^T = A^T - C^T \cdot L^T$
Analogie zum LQR-Problem (dualer Ansatz):

- $A \rightarrow A^T$
- $B \rightarrow C^T$
- $Q = \bar{C}^T \cdot \bar{C} \rightarrow \bar{B} \cdot \bar{B}^T$
- $R = r \cdot I \rightarrow q \cdot I$

Die Lösung kann also mit einer analogen **Matrix-Riccati Gleichung** gefunden werden:

$$\boxed{\frac{1}{q} \cdot \psi \cdot C^T \cdot C \cdot \psi - \psi \cdot A^T - A \cdot \psi - \bar{B} \cdot \bar{B}^T = 0}$$

Und die **Beobachterverstärkung** ergibt analog: $\boxed{L^T = \frac{1}{q} \cdot C \cdot \psi}$

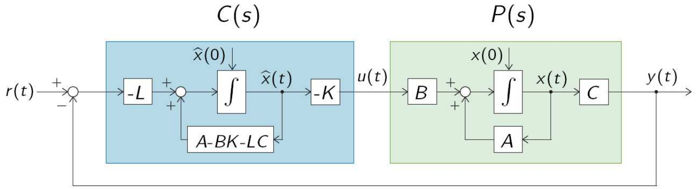
Hinreichende Bedingungen für eine Lösung:

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

Bedingungen für ψ:

- $\psi_1 > 0$
- $\psi_1 \cdot \psi_3 > \psi_2^2$
- und somit auch $\psi_3 > 0$

3.5 Zustandsrückführung mit Zustandsbeobachter - LQG



$L_{LQG}(s) = C \cdot [sI - A]^{-1} \cdot B \cdot K \cdot [sI - (A - B \cdot K - L \cdot C)]^{-1} \cdot L$
Stabilität: Nach Umformumg mit Separationsprinzip lässt sich zeigen, dass die Eigenwerte von $(A - B \cdot K)$ und $(A - L \cdot C)$ untersucht werden müssen.
Robustheit: $\mu_{LQG} = min_{\omega}\{\sigma\{I + L_{LQG}(j\omega)\}\}$ untersuchen

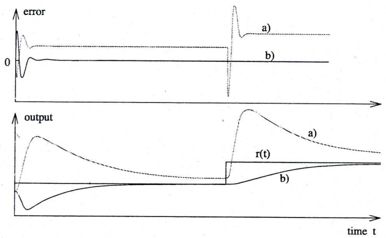
3.5.1 Matlab

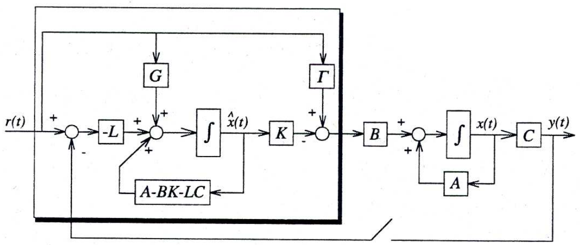
- $K = lqr(A, B, Q, R)$
- $L = lqr(A', C', B \cdot B', q)'$

3.6 Erweiterte LQG Controller

3.6.1 LQG Reference Tracking Controllers

Berechnet man die Schrittantwort eines Systems mit Standard LQG, wird man feststellen, dass der Fehler $\bar{x}$ nicht gegen 0 konvergiert. Lösung:



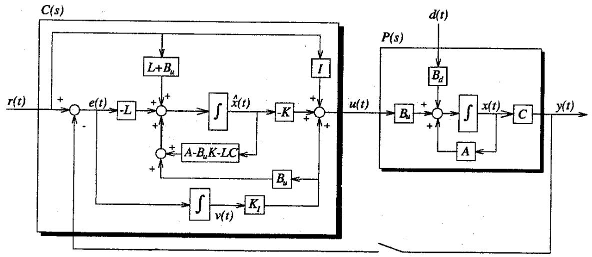


$\frac{d}{dt}\tilde{x}(t) = \tilde{A}_{cl} \cdot \tilde{x}(t) + \tilde{B}_r \cdot \Gamma \cdot r(t)$, $y(t) = \tilde{C} \cdot \tilde{x}(t)$
$\tilde{x}(t) = \begin{bmatrix} x(t) \\ \hat{x}(t) \end{bmatrix}$, $\tilde{A}_{cl} = \begin{bmatrix} A & -BK \\ LC & A - BK - LC \end{bmatrix}$, $\tilde{B}_r = \begin{bmatrix} B \\ B \end{bmatrix}$, $\tilde{C} = \begin{bmatrix} C & 0 \end{bmatrix}$
Eingeführte Matrizen: G und Γ:

- $\Gamma = -[\tilde{C} \cdot \tilde{A}_{cl}^{-1} \cdot \tilde{B}_r]^{-1}$
- $G = L + B \cdot \Gamma$

Diese Systemerweiterung führt (bei gleichbleibenden Matrizen K und L!) zu denselben Robustheitseigenschaften.

3.6.2 LQG Controller mit Integral Action - LQGI



Closed-Loop Dynamik: von $\begin{bmatrix} r \\ d \end{bmatrix} \rightarrow y$
$\frac{d}{dt}\begin{bmatrix} x \\ \hat{x} \\ v \end{bmatrix} = \begin{bmatrix} A & -B_uK & B_uK_I \\ 0 & A - B_uK - LC & B_uK_I \\ -C & 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ \hat{x} \\ v \end{bmatrix} + \begin{bmatrix} B_u & B_d \\ B_u & 0 \\ I & 0 \end{bmatrix}\begin{bmatrix} r \\ d \end{bmatrix}$ Design Vorgang:

1. Das erweiterte System $\{\tilde{A}, \tilde{B}_u, \tilde{C}\}$ wie in 3.2.1 bilden und $\tilde{K} = \begin{bmatrix} K & -K_I \end{bmatrix}$ berechnen
   $\tilde{K} = lqr(\tilde{A}, \tilde{B}_u, \tilde{C}^T \cdot \tilde{C}, r \cdot I)$
2. Für das Originalsystem $\{A, B, C\}$ den Observer Gain L berechnen mit dem dualen Ansatz in 3.4
   $L = lqr(A^T, C^T, B_u \cdot B_u^T, q \cdot I)$

3.7 Robustness Recovery - LTR

LQG Controller haben ein potentielles Robustheitsproblem, deshalb ist ein iterativer Ansatz für dieses Problem notwendig: Loop Transfer Recovery (LTR). Gebraucht wird dafür ein LQR-Controller, der nach bekannten Verfahren bestimmt werden muss.
Nachteil von LTR: Perfektes LTR hat hohe Verstärkung bei hohen Frequenzen zur Folge

3.7.1 naiver Ansatz

Bedingung: minimalphasiges System (darf aber instabil sein). Gesucht wird ein Controller C, so dass $C(s) \cdot P(s) = L_{LQR}(s)$
Offensichtliche Lösung: $C(s) = \frac{L_{LQR}(s)}{P(s)} = \frac{k \cdot Adj(sI-A) \cdot b}{c \cdot Adj(sI-A) \cdot b}$
Dieser Controller kürzt alle Nullstellen der Strecke mit seinen Polen und ersetzt diese

Nullstellen durch die LQR Nullstellen, die bekanntermassen einen guten loop gain machen. Allerdings ist dieses $C(s)$ nicht realisierbar, weil der Grad $n-1$ vom Zähler grösser ist als der Grad m vom Nenner. Ausweg: Kombination mit einer ausreichenden Anzahl von Lag-Elementen mit gleicher Zeitkonstante τ:
$C(s) = \frac{k \cdot Adj(sI-A) \cdot b}{c \cdot Adj(sI-A) \cdot b \cdot (\tau s+1)^{\kappa}}$
Resultierende Loop Transfer Function: $L(s) = \frac{k \cdot Adj(sI-A) \cdot b}{det(sI-A) \cdot (\tau s+1)^{\kappa}}$
Wahl der Eckfrequenz $\frac{1}{\tau}$: Darf Limiten durch Modellunsicherheit und Rauschen nicht überschreiten, aber nicht zu tief sein, um die Instabilitäten der Strecke zu kompensieren.

3.7.2 Reguläre LTR Methode

1. K nach LQR Methode berechnen, $K = lqr(A, B_u, Q, rI)$
2. Observer Gain L berechnen, $L = lqr(A', C', B_u \cdot B_u', qI)'$
3. Open-Loop Frequenzbereich und Closed-Loop Zeitbereich Systemverhalten analysieren
4. Schritt 2 und 3 wiederholen mit sinkendem q bis gutes Systemverhalten resultiert

4 MATLAB

Starten einer Simulation aus einem m-File
$[T, X, Y]$ = sim('model',tfin,[],[])
Simuliert die Datei model.mdl von 0 bis tfin
T: Zeitvektor
X: ???
Y: Output, die einzelnen Ausgänge sind als Vektoren zu einer Matrix zusammengefügt

Modelantwort plotten
plot(T,Y(:,1)) plottet die Systemantwort des 1. Ausganges

Plot bearbeiten
ylabel('y(t)')
xlabel('t')
legend('Ausgang1','Ausgang2',..)
title('Systemantwort')

Place
K=Place(A,B,P) berechnet die Matrix K so, dass die Eigenwerte von $A - B \cdot K$ denjenigen im Vektor P entsprechen