Übung 11 - LTR

1 LTR

Ziel von LTR ist die Robustheit des Systems zu verbessern, vor allem wenn ein Beobachter implementiert ist. In einer solchen Situation gibt es keine Garantie von Robustheit des Systems. Die Grundidee ist einen LQR oder einen Beobachter auszulegen, mit dem ein "gutes" System bekommt, und danach ein LQG mit Tuning Parameter aufzustellen. Der Parameter muss so gewählt werden, dass man das System nah zu dem "guten" System bringt.

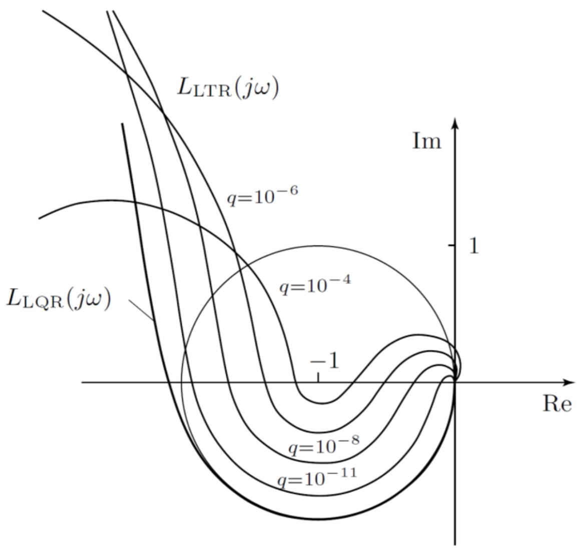

Abbildung 1: LTR.

1.1 β Methode

Um eine bessere Robustheit zu bekommen, verwendet man oft die β Methode.

- LQR: Berechne K mit

$$\frac{1}{r \cdot \beta} \cdot \Phi \cdot B \cdot B^{\mathsf{T}} \cdot \Phi - \Phi \cdot A - A^{\mathsf{T}} \cdot \Phi - Q = 0, \tag{1.1}$$

$$K = R^{-1} \cdot B^{\mathsf{T}} \cdot \Phi. \tag{1.2}$$

  Matlab: `Phi = care(A,B,Q,beta*r*eye(nu))`, `K = 1/r*B'*Phi`.

- LQG: Berechne L mit

$$\frac{1}{q \cdot \beta} \cdot \Psi \cdot C^{\mathsf{T}} \cdot C \cdot \Psi - \Psi \cdot A^{\mathsf{T}} - A \cdot \Psi - \bar{B} \cdot \bar{B}^{\mathsf{T}} = 0, \tag{1.3}$$

$$L^{\mathsf{T}} = \frac{1}{q} \cdot C \cdot \Psi. \tag{1.4}$$

  Matlab: `Psi = care(A',C',B_bar*B_bar',beta*q*eye(ny))`, `L = (1/q*C*Psi)'`.

Bemerkung. Bis jetzt konnte man K und L mit dem Befehl `lqr` bestimmen. Da β nur in der Gleichung für Φ bez. Ψ auftritt (und nicht in der Gleichung für K bez. L) muss man zuerst die Riccati Gleichung lösen (mit `care`) und dann den Gain bestimmen.

1.2 LQG/LQR für $m \geq p$

1. Strecke modellieren, linearisieren und normieren;
2. Definiere Spezifikationen in Frequenzbereich;
3. Lege den Beobachter aus:
   (a) `Psi = care(A',C',B_bar*B_bar',beta*q*eye(ny));`
   (b) `L = (1/q*C*Psi)';`
   (c) Find `B_bar*B_bar'`$\in \mathbb{R}^{n\times n}$, $q > 0$, and $\beta \leq 1$, die die Spezifikationen mit (ungefähr) 3 dB Margin erfüllen;
   (d) Berechne L_{obs} mit
   $$L_{\text{obs}}(s) = C \cdot (s \cdot \mathbb{I} - A)^{-1} \cdot L; \tag{1.5}$$
4. Legen ein LTR "state feedback regulator" aus:
   (a) `K = lqr(A,B,C'*Q_y*C,rho*R_u);`
   (b) Finde `Q_y`$\in \mathbb{R}^{p\times p}$ und `R_u`$\in \mathbb{R}^{m\times m}$ (oft `Q_y = eye(ny)` and `R_u = eye(nu)`);
   (c) Verkleine `rho` bis alle Spezifikationen erfüllt sind;
   (d) Loop gain:
   $$L_{\text{LTR}}(s) = C \cdot (s \cdot \mathbb{I} - A)^{-1} \cdot B \cdot K \cdot (s \cdot \mathbb{I} - (A - B \cdot K - L \cdot C))^{-1} \cdot L; \tag{1.6}$$
5. Regler implementieren.

1.3 LQG/LQR für $p \geq m$

1. Strecke modellieren, linearisieren und normieren;
2. Definiere Spezifikationen in Frequenzbereich;
3. Lege den LQR aus:
   (a) `Phi = care(A,B,Q,beta*q*eye(nu));`
   (b) `K = 1/r*B'*Phi;`
   (c) Finde `Q`$\in \mathbb{R}^{n\times n}$, $r > 0$, and $\beta \leq 1$, die die Spezifikationen mit (ungefähr) 3 dB Margin erfüllen;
   (d) Berechne L_{obs} mit
   $$L_{\text{LQR}}(s) = K \cdot (s \cdot \mathbb{I} - A)^{-1} \cdot B; \tag{1.7}$$
4. Legen ein LTR "state feedback regulator" aus:
   (a) `L = lqr(A',C',B*Q_u*B',mu*R_y)';`
   (b) Finde `Q_u`$\in \mathbb{R}^{m\times m}$ und `R_y`$\in \mathbb{R}^{p\times p}$ (oft `Q_u = eye(nu)` and `R_y = eye(ny)`);
   (c) Verkleine `mu` bis alle Spezifikationen erfüllt sind;
   (d) Loop gain:
   $$L_{\text{LTR}}(s) = C \cdot (s \cdot \mathbb{I} - A)^{-1} \cdot B \cdot K \cdot (s \cdot \mathbb{I} - (A - B \cdot K - L \cdot C))^{-1} \cdot L; \tag{1.8}$$
5. Regler implementieren.