Moderne Methoden der Regelungstechnik

4. Quadratische Ljapunovfunktionen zur Analyse und Synthese

Prof. Dr. Horst Schulte

Sommer-Semester April – Juli 2021

Hochschule für Technik und Wirtschaft Berlin

University of Applied Sciences

Inhalt

Literatur:

1. Slotine, Applied Nonlinear Control, Prentice Hall, 1991
2. Boyd, et al. , Linear Matrix Inequalities in Systemand Control Theory, 1994

[*] LMI: **L**inear **M**atrix **I**nequality

1. Einführung

Methoden des Reglerentwurf für die Klasse der linearen Zustandsregler

- Annahme: System wird mittels linearer Zustandsraummodelle hinreichend gut beschrieben
- d.h. es wird mit dem folgenden Modell beschrieben

  $d\mathbf{x}/dt = \mathbf{A}\,\mathbf{x} + \mathbf{B}\,\mathbf{u}\,,\qquad \mathbf{y} = \mathbf{C}\,\mathbf{x} + \mathbf{D}\,\mathbf{u}$ *(siehe Abschnitt 2)*

- **Methoden**

  1. Vorgabe der Polstellen des geschlossenen Kreises

  (siehe Tutorial *Ackermannformel für SISO Systeme*)

$$\underbrace{\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\}}_{\text{desired closed loop poles}} = \underbrace{\mathrm{eig}\ \mathbf{(A\text{-}BK)}}_{\text{solution of eigenvalue problem}} \Rightarrow \underbrace{\mathbf{K}}_{\text{gives state feedback gain}}$$

  2. Ljapunov-basierte Vorgabe von Polstellenregionen

  - basiert auf quadratischen Ljapunov Funktionen -> Gegenstand dieser Vorlesung

1. Einführung

Methode: Ljapunov-basierte Vorgabe von Polstellenregionen

- Berechnungsverfahren für SISO und MIMO Systeme

  – im Gegensatz zur exakten Polstellenvorgabe werden gewünschte Polstellenregionen vorgegeben

  – Designansatz: Vorgabe Ljapunov-Funktionskandidaten in einer kanonischen quadratischen Form

  – das Finden einer Ljapunovfunktion wird dabei zurückgeführt auf die Berechnung von positiv definiten Matrizen

  – Berechnung basiert auf numerischen konvexen Optimierungsverfahren für lineare Matrixungleichungen

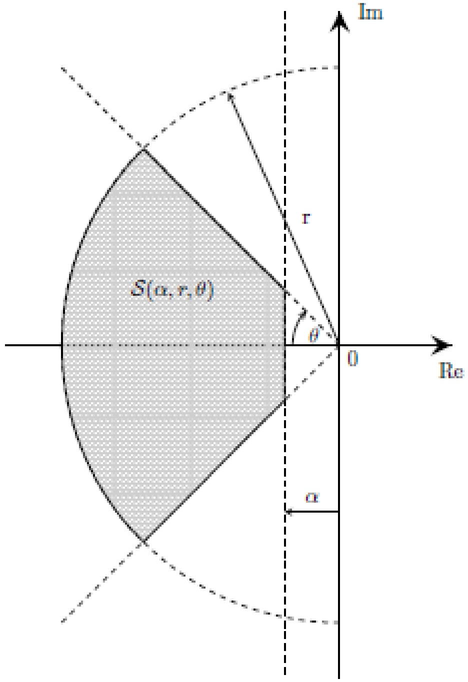
  – Vorteil: sehr flexibel und skalierbar

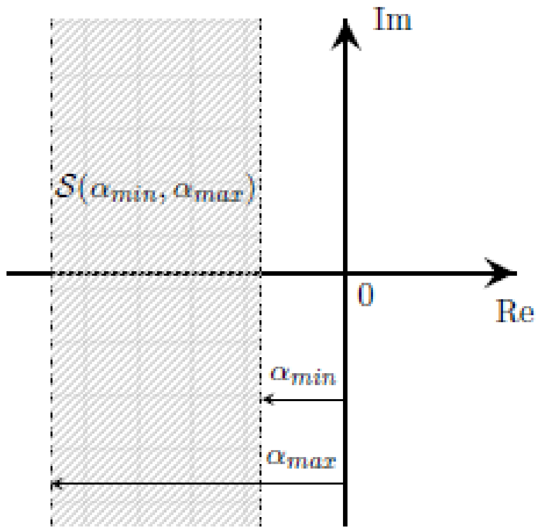
  -> skalierbar: Vorgabe von zusätzlich Anforderung an die Regelung

1. Einführung

Methode: Ljapunov-basierte Vorgabe von Polstellenregionen für den Regelungsentwurf

- Polstellenregionen der gewünschten Eigenwertverteilung





1. Einführung

Motivation

- **Ljapunovfunktionen sind additiv wie die Energien bei der Lagrange Energiemethode**
- Quadratische Funktionen eignen sich als Funktionskandidaten für lineare Systeme
- Gewichete Kombination von linearen Systemen zur Beschreibung von nichtlinearer Systemen können aufgrund der additiven Eigenschaften der Ljapunovfunktion ebenfalls mittels quadratischer Ljapunov Funktionen behandelt werden (Analyse / Synthese)
- das Finden von quadratischen Ljapunovfunktionen wird reduziert auf die berechnung von Linearen Matrixgleichungen und (flexibler) lienaren Matrixungleichungen
- aus diesem Grund werden im folgenden Grundlagen zur Matrixalgebra vorgestellt

2. Grundlagen

2.1 Begriff der positiven / negativen Definitheit

Definition 1: *Positiv definite Matrizen:*

Eine quadratische Matrix mit **M=M**^T ist positiv definit (p.d.) falls gilt:

$$\mathbf{x} \neq 0 \quad => \quad \mathbf{x}^T \mathbf{M}\, \mathbf{x} > 0$$

Definition 2: *Positiv definite Funktion f*(**x**)

Eine Funktion *f*(**x**) ist positiv definit falls für alle **x** gilt *f*(**x**)>0

Mit anderen Worten aus den Definitionen 1 und 2 folgt, dass wenn *f*(**x**) = $\mathbf{x}^T \mathbf{M}\, \mathbf{x} > 0$ gilt, die Matrix **M** stets positiv definit ist

2. Grundlagen

2.1 Begriff der positiven / negativen Definitheit

Definition 3: *Negativ definite Matrizen:*

Eine quadratische Matrix mit **M=M**^T ist negativ definit (n.d.) falls gilt:

$$\mathbf{x} \neq \mathbf{0} \quad => \quad \mathbf{x}^T \mathbf{M} \mathbf{x} < 0$$

Definition 4: *Negativ definite Funktion f*(**x**)

Eine Funktion *f*(**x**) ist negativ definit falls für alle **x** gilt *f*(**x**)<0

Mit anderen Worten aus den Definitionen 3 und 4 folgt, dass wenn

f(**x**) = $\mathbf{x}^T \mathbf{M} \mathbf{x} < 0$ gilt, die Matrix **M** stets negativ definit ist

2. Grundlagen

2.1 Begriff der positiven / negativen Definitheit

Definition 5: *Semi Definitheit*

Eine Matrix bzw. Funktion $f(\mathbf{x})$ ist semi positiv / negativ definit falls statt <> auch die Gleichheit zulässig ist, d.h. $\leq$ $\geq$

Defintion 6: Obere und untere Schranke der quadratischen Funktion .

Es gilt stets falls **M** p.d ist

$$\lambda_{min}(\mathbf{M})\,\|\mathbf{x}\|^2 \leq \mathbf{x}^T\mathbf{M}\mathbf{x} \leq \lambda_{max}(\mathbf{M})\|\mathbf{x}\|^2$$

_min / _max sind die kleinsten / größten Eigenwerte der Matrix **M**

2.1 Begriff der positiven / negativen Definitheit

Numerisches Beispiel für $\lambda_{min}(\mathbf{M})\,\|\mathbf{x}\|^2 \leq \mathbf{x}^T\mathbf{M}\mathbf{x} \leq \lambda_{max}(\mathbf{M})\,\|\mathbf{x}\|^2$

$$M = \begin{pmatrix} 5 & 3 \\ 3 & 4 \end{pmatrix} \qquad \mathrm{eig}\, M \doteq \{1.4586 \;,\; 7.5414\} = \begin{cases} \lambda_{min}(M) = 1.4586 \\ \lambda_{max}(M) = 7.5414 \end{cases}$$

Berechne mit $\det(\lambda I - M) = 0$

$$\det\left[\begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} - \begin{pmatrix} 5 & 3 \\ 3 & 4 \end{pmatrix}\right] = \det\begin{bmatrix} \lambda-5 & -3 \\ -3 & \lambda-4 \end{bmatrix} = 0$$

$$(\lambda-5)(\lambda-4) - 9 = 0$$

$$\lambda^2 - 9\lambda + 20 - 9 = 0$$

$$\lambda^2 - 9\lambda + 11 = 0$$

$$\lambda_{1,2} = +\frac{9}{2} \pm \sqrt{\frac{9^2}{4} - 11} = 4{,}5 \pm \sqrt{9{,}25} = \begin{cases} 1{,}4586 \\ 7{,}5414 \end{cases}$$

$$x = \begin{pmatrix} 1 \\ 2 \end{pmatrix} : \quad \|x\|^2 = 1^2 + 2^2 = 5 \qquad 1{,}4586 \cdot 5 \leq \underbrace{x^T M x}_{33} \leq 7.5414 \cdot 5$$

$$x \begin{pmatrix} -1 \\ 3 \end{pmatrix} : \quad \|x\|^2 = (-1)^2 + 3^2 = 10 \qquad 1.4586 \cdot 10 \leq 23 \leq 7.5414 \cdot 10$$

2. Grundlagen

2.2 Kriterien zum Prüfen der positiven / negativen Definitheit

I) Eigenwerte

Eine quadratische symmetrische (bzw. hermitesche) Matrix ist genau dann

positiv definit,	wenn alle Eigenwerte größer als null sind;
positiv semidefinit,	wenn alle Eigenwerte größer oder gleich null sind;
negativ definit,	wenn alle Eigenwerte kleiner als null sind;
negativ semidefinit,	wenn alle Eigenwerte kleiner oder gleich null sind und
indefinit,	wenn positive und negative Eigenwerte existieren.

Anmerkung: kleiner / größer Null bezieht sich auf den Realteil der Eigenwerte

2.2 Kriterien zum Prüfen der positiven / negativen Definitheit

II) Kriterium mit Hauptminoren

Eine **symmetrische Matrix A ist genau dann positiv definit, wenn alle *führenden Hauptminoren* von A positiv sind**. Aus der Tatsache, dass A genau dann negativ definit ist, wenn − A positiv definit ist, ergibt sich: **A ist genau dann negativ definit, wenn die Vorzeichen der *führenden Hauptminoren* alternieren**, das heißt, falls alle ungeraden führenden Hauptminoren negativ und alle geraden positiv sind.

führende Hauptminoren (Quelle: Wikpedia)

Führende Hauptminoren sind spezielle Hauptminoren, die dadurch entstehen, dass man die Ausgangsmatrix „von ihrem Ende“ her sukzessive um jeweils eine Zeile und Spalte verkürzt und die Determinanten der sich ergebenden Untermatrizen berechnet. So liefert etwa die 3×3-Matrix

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

die folgenden 3 Untermatrizen

$$A_1 = (1), \quad A_2 = \begin{pmatrix} 1 & 2 \\ 4 & 5 \end{pmatrix}, \quad A_3 = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix},$$

2. Grundlagen

2.2 Kriterien zum Prüfen der positiven / negativen Definitheit

aus denen sich anschließend die folgenden 3 *führenden* Hauptminoren berechnen lassen:

- Führender Hauptminor 1. Ordnung: $\det(A_1) = 1;$
- Führender Hauptminor 2. Ordnung: $\det(A_2) = \begin{vmatrix} 1 & 2 \\ 4 & 5 \end{vmatrix} = -3;$
- Führender Hauptminor 3. Ordnung: $\det(A_3) = \begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix} = 0\,.$

Wie zu sehen, gibt es dabei nur einen Hauptminor 3. Ordnung, der zugleich *führend* ist, nämlich die Determinante der gesamten Matrix.

Im dem oben genannten Beispiel ist die Matrix ***A*** weder positiv noch negativ definit, d.h. diese ist indefinit. Zwei weitere Beispiele siehe die nächste Seite:

2.2 Kriterien zum Prüfen der positiven / negativen Definitheit

Beispiele:

$$A_a = \begin{pmatrix} 1 & 2 \\ 2 & 8 \end{pmatrix} \qquad \det(1) = 1$$

$$\det\begin{pmatrix} 1 & 2 \\ 2 & 8 \end{pmatrix} = 1 \cdot 8 - 4 = 4$$

↝ alle führenden Hauptminoren sind positiv ↝ A ist positiv definit

$$A_b = \begin{pmatrix} -1 & -2 \\ 6 & -3 \end{pmatrix} \qquad \left.\begin{aligned} &\det(1) = -1 \\ &\det\begin{pmatrix} 1 & 2 \\ 6 & 3 \end{pmatrix} = 3 + 12 = \cancel{9}\ 15 \end{aligned}\right\} \text{ alterniert}$$

↝ Vorzeichen der Hauptminoren alternieren ↝ A ist negativ definit

ungerade Hauptminore ist negativ, gerade ist positiv

Anmerkung Kriterium mit Eigenwerten

$\mathrm{eig}(A_a) = \{0.4685,\ 8.5311\}$ alle Eigenwerte sind größer Null

$\mathrm{eig}(A_b) = \{-2 \pm j3{,}31\}$ " " sind kleiner Null

2. Grundlagen

2.3 Quadratische Matrixgleichungen zur Stabilitätsanalyse von LTI Systemen

Quadratische Matrixgleichungen ergeben sich aus der quadratischen Lyapunov Funktion $V = \mathbf{x}^T \mathbf{P} \mathbf{x}$

mit der Ableitung $\dot{V} = \dot{\mathbf{x}}^T \mathbf{P} \mathbf{x} + \mathbf{x}^T \mathbf{P} \dot{\mathbf{x}}$

Zur Stabilitätsanalyse des autonomen LTI Systems $\dot{\mathbf{x}} = \mathbf{A} \mathbf{x}$ wird die rechte Seite in die Ableitung der Ljapunovfunktion eingesetzt und ein p.d. **Q** eingeführt

$$\dot{V} = \dot{\mathbf{x}}^T \mathbf{P} \mathbf{x} + \mathbf{x}^T \mathbf{P} \dot{\mathbf{x}} = - \mathbf{x}^T \mathbf{Q} \mathbf{x}$$

$$\mathbf{A}^T \mathbf{P} + \mathbf{P} \mathbf{A} = -\mathbf{Q}$$

2. Grundlagen

2.3 Quadratische Matrixgleichungen zur Stabilitätsanalyse von LTI Systemen

aus Slotine:

The question, thus, is to determine whether the symmetric matrix **Q** defined by the so-called *Lyapunov equation* (3.19) above, is itself *p.d.* If this is the case, then *V* satisfies the conditions of the basic theorem of section 3.4, and the origin is globally asymptotically stable. However, this "natural" approach may lead to inconclusive result, *i.e.*, **Q** may be not positive definite even for stable systems.

Daus heißt, das Finden einer Ljapunov Funktion wird reduziert auf das Bestimmen von positiv definiten Matrizen **P** und **Q** , welche die Matrixgleichung erfüllen

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\,\mathbf{A} = -\mathbf{Q}$$

2. Grundlagen

2.3 Quadratische Matrixgleichungen zur Stabilitätsanalyse von LTI Systemen

Lösungweg:

- choose a positive definite matrix $\mathbf{Q}$
- *solve* for $\mathbf{P}$ from the Lyapunov equation $\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} = -\mathbf{Q}$
- check whether $\mathbf{P}$ is *p.d*

2.3 Quadratische Matrixgleichungen zur Stabilitätsanalyse von LTI Systemen

Beispiel aus Slotine, Seite 97

Example 3.18: Consider again the second-order system of Example 3.17. Let us take $\mathbf{Q} = \mathbf{I}$ denote $\mathbf{P}$ by

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix}$$

where, due to the symmetry of $\mathbf{P}$, $p_{21} = p_{12}$. Then the Lyapunov equation is

$$\begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} \begin{bmatrix} 0 & 4 \\ -8 & -12 \end{bmatrix} + \begin{bmatrix} 0 & -8 \\ 4 & -12 \end{bmatrix} \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

2. Grundlagen

2.3 Quadratische Matrixgleichungen zur Stabilitätsanalyse von LTI Systemen

Beispiel aus Slotine, Seite 97

whose solution is

$$p_{11} = 5/16\,,\quad p_{12} = p_{22} = 1/16$$

The corresponding matrix

$$\mathbf{P} = \frac{1}{16}\begin{bmatrix} 5 & 1 \\ 1 & 1 \end{bmatrix}$$

is positive definite, and therefore the linear system is globally asymptotically stable. Note that we have solved for **P** directly, without using the more cumbersome expression (3.20). □

3. Stabilitätsanalyse mittels Linearer Matrix Ungleichungen

Ansatz: Quadratische Ljapunov Funktion

$$V = \underline{x}^T P \underline{x}, \quad P > 0, \quad P = P^T$$

Analyse der Stabilität (einfache Stabilität)

$$\dot{V} \leq 0$$

asymp. Stabilität

$$\dot{V} < 0$$

Exp. Stabilität

$$\dot{V} < -2\alpha V$$

mit $\alpha > 0$ als Decay Rate

Siehe die Kriterien aus Lecture 1

1. Kriterium zum Nachweis der einfachen Stabilität Slide 19
2. Kriterium zum Nachweis der asymptotischen Stabilität Slide 20
3. Kriterium zum Nachweis der exponentiellen Stabilität Slide 21

3. Stabilitätsanalyse mittels Linearer Matrix Ungleichungen

Untersuchung der Erfüllung der Kriterein für ein gegebenes ***autonomes Systems*** (also ein System ohne steuerbarenEingang)

$$\dot{\underline{x}} = A\,\underline{x} \qquad A \in \mathbb{R}^{n\times n}, \quad \underline{x} \in \mathbb{R}^{n}$$

$$\dot{V} = \frac{d}{dt}\left(\underline{x}^T P\,\underline{x}\right) = \dot{\underline{x}}^T P\,\underline{x} + \underline{x}^T P\,\dot{\underline{x}}$$

Ersetzen von $\dot{\underline{x}}$ durch das Zustandsraummodell $\rightarrow$

$$\dot{V} = (A\,\underline{x})^T P\,\underline{x} + \underline{x}^T P A\,\underline{x}$$

$$= \underline{x}^T A^T P\,\underline{x} + \underline{x}^T P A\,\underline{x}$$

$$\dot{V} = \underline{x}^T \left(A^T P + P A\right) \underline{x} < 0$$

3. Stabilitätsanalyse mittels Linearer Matrix Ungleichungen

Dies ist erfüllt falls $A^T P + P A < 0$ gilt, also

die zusammengesetzte Matrix neg. Definit ist.

- Daraus ergeben sich die Kriterien zur asymptotisch Stabilität mit Linearen Matrix Ungleichungen

$$\boxed{\begin{array}{l} P > 0 \\ A^T P + P A < 0 \end{array}}$$

- Variable ist P
- A ist durch das System / Modell gegeben

$\rightarrow$ System $\dot{\underline{x}} = A\, \underline{x}$ ist asymptotisch stabil

3. Stabilitätsanalyse mittels Linearer Matrix Ungleichungen

- Mit dem Kriterium der exponentiellen Stabilität $\dot{V} < -2\alpha V$

folgt

$$\underbrace{\underline{x}^T (A^T P + P A)\, \underline{x}}_{\dot{V}} < \underbrace{-2\alpha\ \underline{x}^T P\, \underline{x}}_{-2\alpha V}$$

$$\underline{x}^T \left(A^T P + P A + 2\alpha P\right) \underline{x} < 0$$

Dies ist erfüllt falls die zusammengesetzte Matrix neg. definit ist,

also

$$A^T P + P A + 2\alpha P < 0$$

gilt.

3. Stabilitätsanalyse mittels Linearer Matrix Ungleichungen

- Zusammengefaßt ergibt sich das Kriterium zur exponentielle Stabilität mit geforderter decay rate

$$\boxed{\begin{aligned} &P > 0 \\ &A^T P + P A + 2\alpha P < 0 \end{aligned}}$$

Anmerkung zur decay rate α:

$V(\underline{x}) = V_0\, e^{-2\alpha t}$ mit $V_0(\underline{x}_0) =: V_0$ $\quad \underline{x}_0 := \underline{x}(t=0)$

erfüllt die DGL. $\dot{V} = -2\alpha V$ weil

$$\underbrace{-2\alpha V_0 e^{-2\alpha t}}_{\dot{V}} = -2\alpha \underbrace{V_0\, e^{-2\alpha t}}_{V}$$

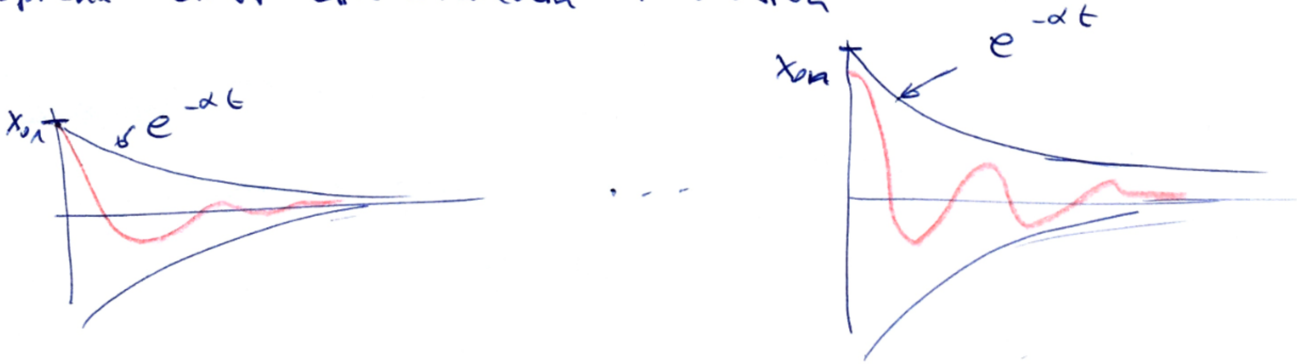
3. Stabilitätsanalyse mittels Linearer Matrix Ungleichungen

$$V(\underline{x}(t)) < V_0 \, e^{-2\alpha t}$$

$$\underline{x}^T P \, \underline{x} < \underline{x}_0^T e^{-\alpha t} P \, \underline{x}_0 \, e^{-\alpha t} = \underbrace{\underline{x}_0^T P \, \underline{x}_0}_{V_0} \, e^{-2\alpha t}$$

mit $\underline{x}(t) < \underline{x}_0 \, e^{-\alpha t}$

entspricht einer einhüllenden Funktion



Anmerkung Ende

3. Stabilitätsanalyse mittels Linearer Matrix Ungleichungen

Numerisches Lösen von Linearen Matrix Ungleichungen (LMIs) mittels konvexer Optimierungsverfahren

1. Matlab Toolbox : Robust Control Toolbox

   Enthält LMI Solvers, LMI Controller Synthesis

2. Sedumi Interface

   Vereinfacht den Aufsatz von LMI's mittels

   WYSIWYG Editor

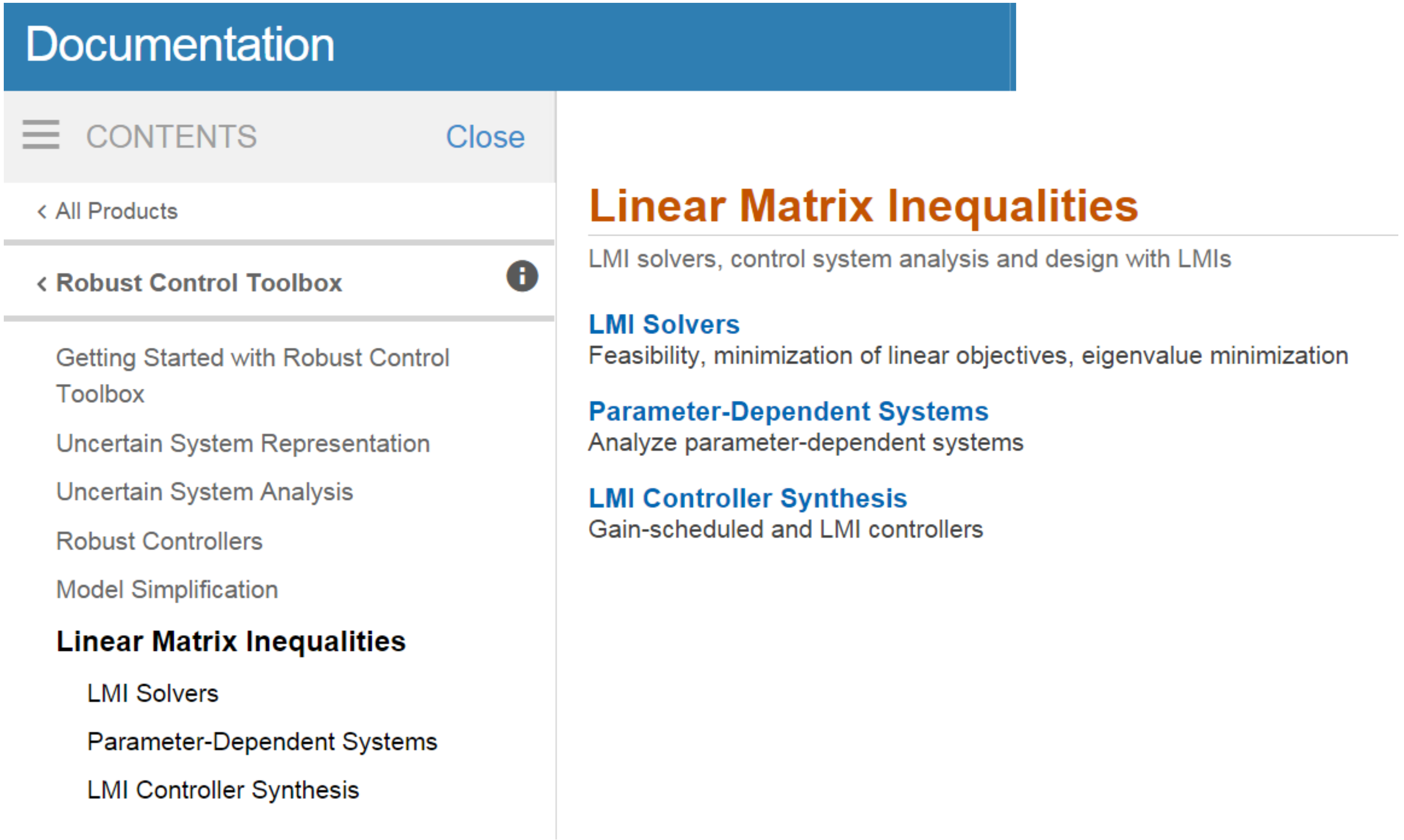
3. Yalmip für Matlab and Python

   siehe https://yalmip.github.io

3. Stabilitätsanalyse mittels Linearer Matrix Ungleichungen

Matlab Toolbox mit LMI Solver

Befehle (siehe help „x“) oder „Documentation“



Documentation

CONTENTS Close

‹ All Products

‹ **Robust Control Toolbox**

Getting Started with Robust Control Toolbox

Uncertain System Representation

Uncertain System Analysis

Robust Controllers

Model Simplification

Linear Matrix Inequalities

LMI Solvers

Parameter-Dependent Systems

LMI Controller Synthesis

Linear Matrix Inequalities

LMI solvers, control system analysis and design with LMIs

LMI Solvers
Feasibility, minimization of linear objectives, eigenvalue minimization

Parameter-Dependent Systems
Analyze parameter-dependent systems

LMI Controller Synthesis
Gain-scheduled and LMI controllers

3. Stabilitätsanalyse mittels Linearer Matrix Ungleichungen

LMI Solvers

Feasibility, minimization of linear objectives, eigenvalue minimization

Functions

`getlmis`	Internal description of LMI system
`lmiedit`	Specify or display systems of LMIs as MATLAB expressions
`lmiterm`	Specify term content of LMIs
`lmivar`	Specify matrix variables in LMI problem
`newlmi`	Attach identifying tag to LMIs
`setlmis`	Initialize description of LMI system
`dellmi`	Remove LMI from system of LMIs
`delmvar`	Remove one matrix variable from LMI problem
`setmvar`	Instantiate matrix variable and evaluate all LMI terms involving this matrix variable

`dec2mat`	Given values of decision variables, derive corresponding values of matrix variables
`decinfo`	Describe how entries of matrix variable X relate to decision variables
`decnbr`	Total number of decision variables in system of LMIs
`lmiinfo`	Information about variables and term content of LMIs
`lminbr`	Return number of LMIs in LMI system
`mat2dec`	Extract vector of decision variables from matrix variable values
`matnbr`	Number of matrix variables in system of LMIs

`defcx`	Help specify cTx objectives for mincx solver
`feasp`	Compute solution to given system of LMIs
`gevp`	Generalized eigenvalue minimization under LMI constraints
`mincx`	Minimize linear objective under LMI constraints

3. Stabilitätsanalyse mittels Linearer Matrix Ungleichungen

Matlab Toolbox mit LMI Solver

Code Beispiele zum Nachweis der asymptotischen und exponentiellen Stabilität mit quadratischen Ljapunov Funktionen

```
%-----------------------------------------------
% test_LMI_functions.m
%-----------------------------------------------

%A = [1 20; 20 40];
A = [-1 -20; 20 -40];
alpha = 10;


P_asy = func_criteria_asyStability(A);
eig(P_asy)


P_exp = func_criteria_expStability(A,alpha);
eig(P_exp)
```

3. Stabilitätsanalyse mittels Linearer Matrix Ungleichungen

Nachweis der asymptotischen Stabilität

```
%------------------------------------------------------------------
% func_criteria_asyStability      14.6.2021
%------------------------------------------------------------------
% Matlab function for solving LMI for proving aymptotic stability
%------------------------------------------------------------------
function Psol = func_criteria_asyStability(A)


%--------------------------------------------------
% 1. Beginning of a LMI program
%--------------------------------------------------
setlmis([]);


%--------------------------------------------------
% 2. Define all LMI variable
%--------------------------------------------------
varType = 1;    % symmetric block diagonal
[n,~]=size(A);
P = lmivar(varType,[n 1]);  % specify variable P as square symmetric (n,n) matrix
```

3. Stabilitätsanalyse mittels Linearer Matrix Ungleichungen

```
%------------------------------------------------------------
% 3. Define all LMI terms lmiterm(TERMID,A,B,FLAG)   L(Z) < R(Z)
%------------------------------------------------------------
% with TERMID(1) = +n  ->  left-hand side of the n-th L
%      TERMID(1) = -n  ->  right-hand side of the n-th LMI
%lmiterm([-1 1 1 P],-1,1);         % LMI constraint -P < 0 <=> P > 0
%%lmiterm([ 1 1 1 P], 1,1);        % LMI constraint P > 0

%lmiterm([1 1 1 P],A',1,'s');      % A'P + PA  < 0
%lmiterm([1 1 2 0],1);            % 0
%lmiterm([1 2 2 P],-1,1);         % P > 0

lmiterm([1 1 1 P],1,A,'s');      % A'P + PA  + 2 alpha P < 0
lmiterm([1 1 1 P],2*alpha,1)
lmiterm([-2 1 1 P],1,1);         % P > 0
%------------------------------------------------------------



%------------------------------------------------------------
% 4. Solve LMI problem
%------------------------------------------------------------

LMISYS = getlmis;                % Declare the whole LMI problem
[tmin,xfeas] = feasp(LMISYS);    % Solve the LMI problem
disp(tmin)

Psol = dec2mat(LMISYS,xfeas,P);  % Get the numerical value of P
```

Nachweis der exponentiellen Stabilität für gewünschte Decay Raten alpha

```
%--------------------------------------------------------------------------------
% func_criteria_expStability                                          14.6.2020
%--------------------------------------------------------------------------------
% Matlab function for solving a LMI for exponential stability of closed
% loop state feedback controlled systems
%--------------------------------------------------------------------------------
function Psol = func_criteria_expStability(A,alpha)

%-------------------------------------------------------------
% 1. Beginning of a LMI program
%-------------------------------------------------------------
setlmis([]);


%-------------------------------------------------------------
% 2. Define all LMI variables
%-------------------------------------------------------------
varType = 1;    % symmetric block diagonal
[n,~]=size(A);
P = lmivar(varType,[n 1]);   % specify variable P as square symmetric (n,n) matrix
```

3. Stabilitätsanalyse mittels Linearer Matrix Ungleichungen

```
%---------------------------------------------------------------
% 3. Define all LMI terms lmiterm(TERMID,A,B,FLAG)   L(Z) < R(Z)
%---------------------------------------------------------------
% with TERMID(1) = +n  ->  left-hand side of the n-th L
%      TERMID(1) = -n  ->  right-hand side of the n-th LMI
%lmiterm([-1 1 1 P],-1,1);         % LMI constraint -P < 0 <=> P > 0
%%lmiterm([ 1 1 1 P], 1,1);        % LMI constraint P > 0

%lmiterm([1 1 1 P],A',1,'s');      % A'P + PA  < 0
%lmiterm([1 1 2 0],1);            % 0
%lmiterm([1 2 2 P],-1,1);         % P > 0

lmiterm([1 1 1 P],1,A,'s');      % A'P + PA  + 2 alpha P < 0
lmiterm([1 1 1 P],2*alpha,1)
lmiterm([-2 1 1 P],1,1);         % P > 0
%---------------------------------------------------------------


%---------------------------------------------------------------
% 4. Solve LMI problem
%---------------------------------------------------------------

LMISYS = getlmis;                % Declare the whole LMI problem
[tmin,xfeas] = feasp(LMISYS);    % Solve the LMI problem
disp(tmin)

Psol = dec2mat(LMISYS,xfeas,P);  % Get the numerical value of P
```

4.1 Einfache Zustandsrückführung ohne Referenzwert d.h. ohne Folgeregelung

① Zustandsrückführung $\underline{u} = -K\,\underline{x}$ (1)

für das Modell der Regelstrecke

$$\dot{\underline{x}} = A\underline{x} + B\underline{u} \qquad (2)$$

Annahme: alle Zustände $\underline{x}$ sind erfaßbar und das System ist steuerbar. d.h.

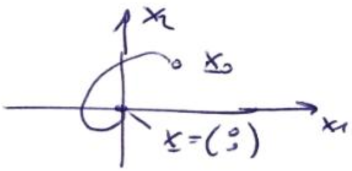
$\mathrm{rank}(Q_s) = n$ mit

$$Q_s = (B, AB, \dots, A^{n-1}B)$$

ohne Tracking $\underline{x}_{ref} \neq \underline{0}$

d.h. $\underline{x}_0 \xrightarrow{\underline{u}} \underline{0}$

als Ziel der Regelung



4. Reglersynthese mittels Linearer Matrix Ungleichungen

Zustandsraummodell des geschlossenen Kreises ergibt sich durch das Ersetzen von $\underline{u}$ in (2) mit dem Regelgesetz (1)

$$\dot{\underline{x}} = A\underline{x} + B(-K\underline{x}) = (A - BK)\underline{x} \qquad (3)$$

Regelziel:

Gesucht wird eine Matrix K, so dass das geschlossene System (3) den Nachweis der exponentiellen Stabilität nach Ljapunov erfüllt,

siehe Lecture 1. Stabilitätsbegriff nach Ljapunov, Slide 21

4. Reglersynthese mittels Linearer Matrix Ungleichungen

- Der Nachweis der exponentiellen Stabilität zur Synthese (Entwurf) erfolgt mit einer LMI Formulierung.
- Hierbei ist sowohl P als auch K eine Variable, welche eine LMI erfüllen muss. Die Herleitung ~~erfolgt~~ ergibt sich wie folgt.

- Ansatz: Quadratische Ljapunov Funktion

$$V(\underline{x}) = \underline{x}^T P \, \underline{x} \qquad \text{muss } V(\underline{x}) > 0 \text{ erfüllen}$$

$$\curvearrowright \ \hat{=} \ P > 0$$

Wie zuvor bei der Analyse bilden wir die Ableitung von $V(x)$:

$$\frac{dV(\underline{x})}{dt} = \dot{\underline{x}}^T P \underline{x} + \underline{x}^T P \dot{\underline{x}}$$

und ersetzen $\dot{\underline{x}}$ bzw. $\dot{\underline{x}}^T$ durch die rechte Seite der Zustandsdiffgl. In diesem Fall enthält diese die zu bestimmende K Matrix des Reglers als $\dot{\underline{x}} = (A - BK)\underline{x}$

$$\frac{dV(\underline{x})}{dt} = \dot{V}(\underline{x}) = \left[(A - BK)\underline{x}\right]^T P \underline{x} + \underline{x}^T P (A - BK)\underline{x}$$

4. Reglersynthese mittels Linearer Matrix Ungleichungen

$$\dot V(\underline{x}) = \underline{x}^T (A - BK)^T P \underline{x} + x^T (PA - PBK)\underline{x}$$

$$\dot V(\underline{x}) = \underline{x}^T \left[A^T P - K^T B^T P + PA - PBK \right] \underline{x}$$

Zum Nachweis der exponentiellen Stabilität muss das Kriterium

$$\dot V(x) < -2\alpha V(\underline{x})$$

erfüllt sein (siehe Lecture 1, Slide 21)

$$\curvearrowright \quad \dot V(\underline{x}) + 2\alpha V(x) < 0$$

4. Reglersynthese mittels Linearer Matrix Ungleichungen

Somit ergibt sich mit der zuvor ermittelten Ableitung von $V(\underline{x})$ die Ungleichung

$$\cancel{A-B} \qquad \underline{x}^T\left(A^TP + PA - K^TB^TP - PBK\right)\underline{x} + 2\alpha\, \underline{x}^T P \underline{x} < 0$$

$$\underbrace{\underline{x}^T\left(A^TP + PA - K^TB^TP - PBK + 2\alpha P\right)\underline{x}}_{f(\underline{x})|_{P,K}} < 0$$

$f(\underline{x})$ ist negativ definit d.h. $f(\underline{x}) < 0$ falls die zusammengesetzte Matrix

$$A^TP + PA - K^TB^TP - PBK + 2\alpha P < 0$$

negativ definit ist.

Zu beachten ist, daß die Matrixungleichung mit P und K als Variablen keine Lineare Matrix Ungleichung ist weil die Terme PBK und $K^T B^T P$ nicht linear in P und K sind!

Um das zu erreichen wird eine Hilfsvariable eingeführt und diese ~~vor~~ links und rechts mit der zusammengesetzten Matrix multipliziert

$$P^{-1} = X \quad , \qquad X \cdot \left| \; A^T P + PA - K^T B^T P - PBK + 2\alpha P < 0 \; \right| \cdot X$$

$$(P^{-1})^T = X^T \qquad X A^T \underbrace{PX}_{I} + \underbrace{XP}_{I} A X - X K^T B^T \underbrace{PX}_{I} - \underbrace{XP}_{I} B K X + 2\alpha X < 0$$

$$\leadsto X = X^T$$

für ein gesuchtes α

4. Reglersynthese mittels Linearer Matrix Ungleichungen

$$X A^T + A X - \underbrace{X^T}_{X} K^T B^T - B K X + 2\alpha X < 0$$

Mit der neuen Variablen $M = K X$, $M^T = X^T K^T = X K^T$

folgt eine Formulierung als Lineare Matrixungleichung mit

den Matrixvariablen $P = \boxed{X}^{-1}$ und M. Aus M und X

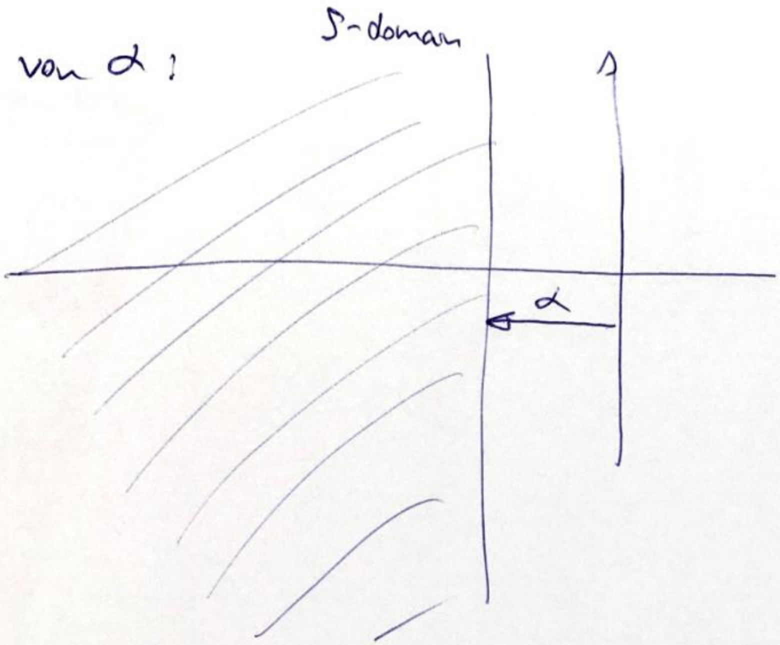
wird K ermittelt zu $K = M X^{-1}$ mit $M \in \mathbb{R}^{m \times n}$, $X \in \mathbb{R}^{n \times n}$

Zusammenfassend erhalten wir das LMI Kriterium der

exponentiellen Stabilität zur Reglersynthese von $u = -K \underline{x}$

$$\boxed{\begin{aligned} &X A^T + A X - M^T B^T - B M + 2\alpha X < 0 \\ &X > 0 \end{aligned}} \Rightarrow P, K$$

Wahl von α:



Mit der Anwendung des LMI Kriteriums wird ein M und P bestimmt

woraus mit $M = K X$

$K = M X^{-1}$ folgt

Es kann gezeigt werden, dass damit

$$\mathrm{Re}\{\mathrm{eig}(A - BK)\} < \alpha$$

erfüllt ist!

4. Reglersynthese mittels Linearer Matrix Ungleichungen

4.2 LMI Reglersynthese inkl. Matlab Code für einfache Zustandsrückführung

LMI Reglersynthese mit geforderter Decay rate α

$$XA^T + AX - M^T B^T - BM + 2\alpha X < 0$$
$$X > 0$$

für die Regelstrecke $\quad \dot{\underline{x}} = A\underline{x} + B\underline{u}$

mit dem Regelungsgesetz $\quad \underline{u} = -K\underline{x} \quad$ wobei $\quad K = MX^{-1}$

4. Reglersynthese mittels Linearer Matrix Ungleichungen

Code Beispiel: LMI Reglersynthese mit geforderter decay rate

```
%--------------------------------------------------------------------------------
% func_criteria_Statefeedback_design_exp_dynamic                5.7.2021
%--------------------------------------------------------------------------------
% Matlab function for solving a LMI design problem for state feedback controller
% design with guaranteed exponential closed loop dynamics
%--------------------------------------------------------------------------------
function [K, LMIsys] = func_criteria_Statefeedback_design_exp_dynamic(A,B,alpha)


%-------------------------------------------------------------
% 1. Initialize the creation of a system of LMIs
%-------------------------------------------------------------
setlmis([]);


%-------------------------------------------------------------
% 2. Specify LMI variables: X and M
%-------------------------------------------------------------
varType_symBlock        = 1;  % symmetric block diagonal
varType_fullRectangular = 2;  % full rectangular

[n,~]=size(A);
[~,m]=size(B);


X = lmivar(varType_symBlock,[n 1]);          % specify variable X as square symmetric (n,n) matrix
M = lmivar(varType_fullRectangular,[m n]);   % specify variable M as full rectangular (m,n)
```

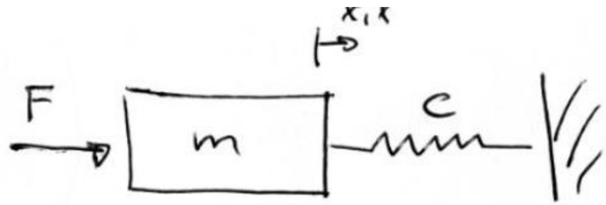
4. Reglersynthese mittels Linearer Matrix Ungleichungen

Code Beispiel: LMI Reglersynthese mit geforderter decay rate

```
%--------------------------------------------------------------
% 3. Specify all LMI terms lmiterm(TERMID,A,B,FLAG)   L(Z) < R(Z)
%--------------------------------------------------------------
% with TERMID(1) = +n  ->  left-hand side of the n-th L
%      TERMID(1) = -n  ->  right-hand side of the n-th LMI
%--------------------------------------------------------------
%   X A' + A X  - M'B' - BM + 2 alpha X < 0
%   X > 0
%--------------------------------------------------------------
lmiterm([1 1 1 X],A,1,'s');
lmiterm([1 1 1 X],2*alpha,1);
lmiterm([1 1 1 M],-B,1,'s');
lmiterm([-2 1 1 X],1,1);         % X > 0
%--------------------------------------------------------------


%--------------------------------------------------------------
% 4. Solve LMI problem
%--------------------------------------------------------------

LMIsys = getlmis;                % Declare the whole LMI problem
[tmin,xfeas] = feasp(LMIsys);    % Solve the LMI problem
disp(tmin)

X_sol = dec2mat(LMIsys,xfeas,X);  % Get solution as the numerical value of X
M_sol = dec2mat(LMIsys,xfeas,M);  % Get solution as the numerical value of M

K = M_sol * inv(X_sol);
```

Beispiel: Aktive Dämpfung eines mechanischen Oszillators

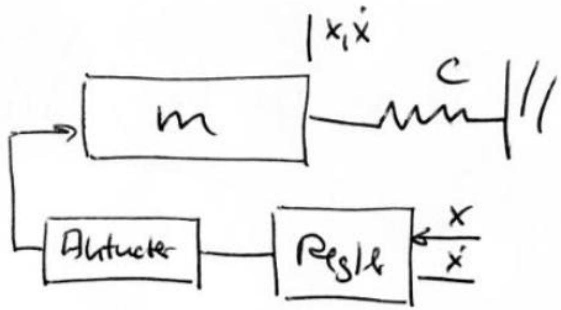
Strecke



$m = 2$ kg

$c = 1$ N/m

Regelziel: Aktive Dämpfung der mech. Schwingung mittels Zustandsregelung



$x, \dot{x}$ sind Meßgrößen

4. Reglersynthese mittels Linearer Matrix Ungleichungen

Lösung: 1. Schritt: Analytische Modellbildung

$$m\ddot{x} = F - cx \qquad \text{Bewegungsgl. nach Newton} \quad m\ddot{x} = \sum_{i=1}^{p} F_i$$

Mit $x_1 = x$, $x_2 = \dot{x}$ folgt (sowie $u = F$)

$$m\dot{x}_2 = u - c x_1$$

$$\dot{x}_2 = -\frac{c}{m} x_1 + \frac{1}{m} u$$

$$\dot{x}_1 = x_2$$

$$\leadsto \quad \dot{\underline{x}} = \begin{pmatrix} 0 & 1 \\ -c/m & 0 \end{pmatrix} \underline{x} + \begin{pmatrix} 0 \\ 1/m \end{pmatrix} u, \quad A = \begin{pmatrix} 0 & 1 \\ -c/m & 0 \end{pmatrix}, \quad \underline{B} = \begin{pmatrix} 0 \\ 1/m \end{pmatrix}$$

$$\text{mit} \quad \underline{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \qquad A = \begin{pmatrix} 0 & 1 \\ -1/2 & 0 \end{pmatrix}, \quad \underline{B} = \begin{pmatrix} 0 \\ 1/2 \end{pmatrix}$$

2. Schritt: Formulierung als LMI Problem mittels m-function

```
%-------------------------------------------------------------------------------
% example_Statefeedback_design_exp_dynamic.m                6.7.2021
%-------------------------------------------------------------------------------
clear all

% specify plant model with system and input matrix
A  = [ 0     1;
      -1/2   0];
B  = [0;
      1/2];

% design parameter, decay rate
alpha = 2;

%-----------------------------------------------
% spefify and solve LMIs for controller design
%-----------------------------------------------
% X A' + A X  - M'B' - BM + 2 alpha X < 0
% X < 0
[K,LMIsys] = func_criteria_Statefeedback_design_exp_dynamic(A,B,alpha);
%-----------------------------------------------

%lmiinfo(LMISYS)

% test closed loop dynamics
eig(A-B*K)
```

Beispiele Aufgabe (siehe Beisp. Anwendung Ackermannformel)

```
 Solver for LMI feasibility problems L(x) < R(x)
    This solver minimizes  t  subject to  L(x) < R(x) + t*I
    The best value of t should be negative for feasibility

 Iteration   :    Best value of t so far

     1                        0.127933
     2                       -0.040475

 Result:  best value of t:    -0.040475
          f-radius saturation:  0.000% of R =  1.00e+09

   -0.0405


K =

   40.1586   14.0256

eig(A-B*K)

ans =

  -3.5064 + 2.8783i
  -3.5064 - 2.8783i
```

4.3 Zustandsrückführung mit Folgeregelung (Problemformulierung)

$$\underline{u} = K_I \int_0^t (\underline{y}_{ref} - \underline{y})\, d\tau - K_x\, \underline{x}$$

Vorbereitung für die LMI Synthese

Erweitertes Modell (siehe Lectüre "Tutorial Zustandsrückführung mit Ref. wert vorgabe)

$$\dot{\underline{\tilde{x}}} = \tilde{A}\, \underline{\tilde{x}} + \tilde{B}\, u + \begin{pmatrix} 0_{n\times 1} \\ \underline{y}_{ref} \end{pmatrix}$$

mit $\tilde{A} = \begin{pmatrix} A & 0 \\ -C & 0 \end{pmatrix}$, $\tilde{B} = \begin{pmatrix} B \\ 0 \end{pmatrix}$, $\underline{\tilde{x}} = \begin{pmatrix} x \\ \underline{x}_I \end{pmatrix}$, $\underline{x}_I = \int_0^t (\underline{y}_{ref} - \underline{y})\, d\tau$

Regelgesetz

$$\underline{u} = -K_x\, \underline{x} + K_I\, \underline{x}_I = -\tilde{K}\, \underline{\tilde{x}}, \quad \tilde{K} = (K_x \;\; -K_I)$$

4. Reglersynthese mittels Linearer Matrix Ungleichungen

4.4 LMI Synthese für Zustandsrückführung mit Folgeregelung

$$X\tilde{A}^T + \tilde{A}X - \tilde{M}^T\tilde{B}^T - \tilde{B}\tilde{M} + 2\alpha X < 0$$

$X > 0$ mit $X \in \mathbb{R}^{(n+p)\times(n+p)}$

für die erweiterte Regelstrecke $\dot{\tilde{\underline{x}}} = \tilde{A}\tilde{\underline{x}} + \tilde{B}u$

mit dem Regelgesetz

$$\underline{u} = -K_x\underline{x} + K_I x_I$$

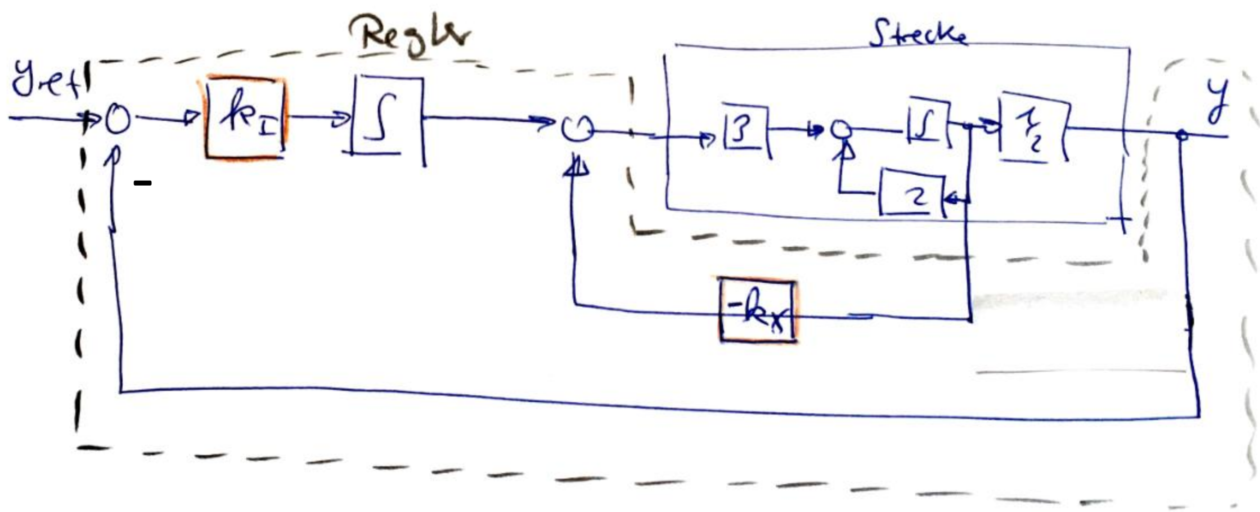
$$\underline{u} = -K_x\underline{x} + K_I\int_0^t (y_{ref} - \underline{y})\,d\tau$$

4.4 LMI Synthese für Zustandsrückführung mit Folgeregelung

Beispiel: geg. instabile Strecke

$$\dot{x} = 2x + 3u \quad , \quad y = \tfrac{1}{2} x$$

Gesucht ist eine Folgeregelung der Form



4.4 LMI Synthese für Zustandsrückführung mit Folgeregelung

Erstellen des erweiterten Modells

$$\tilde{A} = \begin{pmatrix} A & 0 \\ -C & 0 \end{pmatrix}$$ mit $A = 2$, $C = 1/2$ folgt daraus

$$\tilde{A} = \begin{pmatrix} 2 & 0 \\ -1/2 & 0 \end{pmatrix}$$ sowie $$\tilde{B} = \begin{pmatrix} B \\ 0 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \end{pmatrix}$$ da $B = 3$

LMI Formulierung

$$X\tilde{A}^T + \tilde{A}X - \tilde{M}^T\tilde{B}^T - \tilde{B}\tilde{M} + 2\alpha X < 0 \,, \quad X > 0$$

$X \in \mathbb{R}^{2\times 2}$

4. Reglersynthese mittels Linearer Matrix Ungleichungen

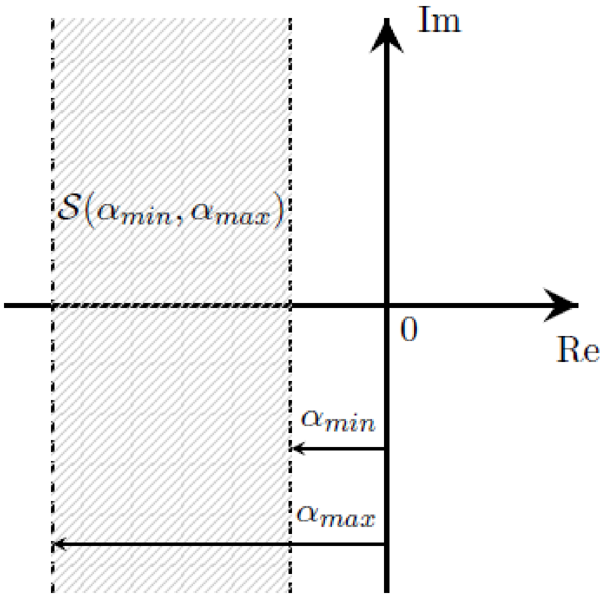
```
%-----------------------------------------------
% Beispiel_Folgeregelung.m
%-----------------------------------------------

A = [2  0;  -1/2  0];
B = [3 0]';

alpha = 4;


[K, LMIsys] = func_criteria_Statefeedback_design_exp_dynamic(A,B,alpha);
```

```
 Solver for LMI feasibility problems L(x) < R(x)
    This solver minimizes  t  subject to  L(x) < R(x) + t*I
    The best value of t should be negative for feasibility

 Iteration   :    Best value of t so far

     1                        0.102368
     2                        0.016330
     3                       -0.796270

 Result:  best value of t:    -0.796270
          f-radius saturation:  0.001% of R =  1.00e+09

   -0.7963

>> eig(A-B*K)

ans =

  -6.0566 + 5.2189i
  -6.0566 - 5.2189i

>> K

K =

    4.7044  -42.6132
```

4.5 Design mit erweiterter Vorgabe der Polestellenregionen

Motivation: Passgenauere Vorgabe der Closed Loop Dynamik

1.) Pole region $\mathcal{S}(\alpha_{\min}, \alpha_{\max})$: The striped pole region $\mathcal{S}(\alpha_{\min}, \alpha_{\max})$ illustrated in Figure C.2 is defined by two parameters where $\alpha_{\max} > \alpha_{\min} > 0$.



4. Reglersynthese mittels Linearer Matrix Ungleichungen

LMI Synthese mit streifenförmiger Poleregion (für ein LTI System wobei i,j = 1)

$$\mathbf{A}_i\mathbf{X} + \mathbf{X}\mathbf{A}_i^T - \mathbf{B}_i\mathbf{M}_j - \mathbf{M}_j^T\mathbf{B}_i^T + 2\alpha_{\min}\mathbf{X} \prec 0$$

$$\mathbf{A}_i\mathbf{X} + \mathbf{X}\mathbf{A}_i^T - \mathbf{B}_i\mathbf{M}_j - \mathbf{M}_j^T\mathbf{B}_i^T + 2\alpha_{\max}\mathbf{X} \succ 0$$

$$\mathbf{X} \succ 0$$

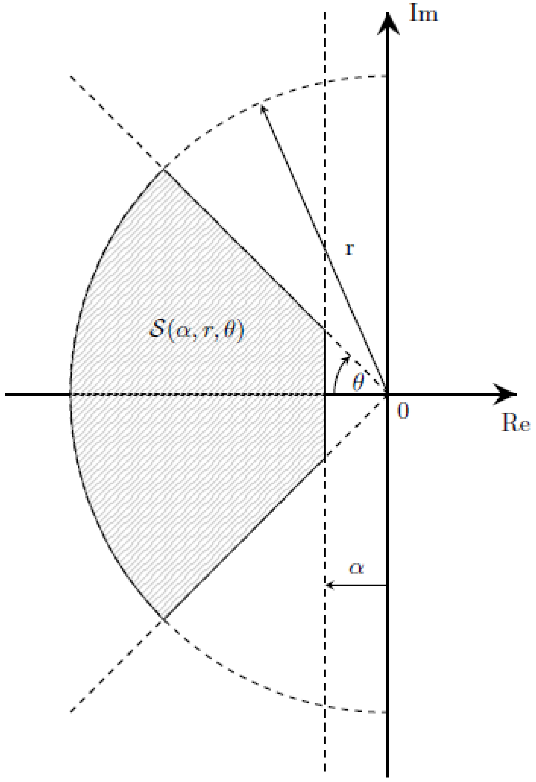
Falls eine Lösung für die obere LMI exisitiert, so ergibt sich die Matrix der Zustandsrückführung zu

$$\mathbf{K}_j = \mathbf{M}_j\mathbf{X}^{-1}$$

wobei die Eigenwerte des geschlossenen Kreises in der Polstellenregion $\mathcal{S}(\alpha_{\min}, \alpha_{\max})$ liegen.

4. Reglersynthese mittels Linearer Matrix Ungleichungen

2. Polregion $\mathcal{S}(\alpha, r, \theta)$



4. Reglersynthese mittels Linearer Matrix Ungleichungen

LMI Synthese für LTI Systeme wobei mit i,j=1

$\mathbf{M}_j,\ j = 1, 2, \ldots, N_r$ *such that the LMI problem*

$$\mathbf{\Gamma}_{ij}^1 = \mathbf{A}_i\mathbf{X} + \mathbf{X}\mathbf{A}_i^T - \mathbf{B}_i\,\mathbf{M}_j - \mathbf{M}_j^T\mathbf{B}_i^T + 2\alpha\mathbf{X}\,,$$

$$\mathbf{\Gamma}_{ij}^2 = \begin{pmatrix} (\mathbf{A}_i\mathbf{X} + \mathbf{X}\mathbf{A}_i^T - \mathbf{B}_i\mathbf{M}_j - \mathbf{M}_j^T\mathbf{B}_i^T)\sin\theta & (\mathbf{A}_i\mathbf{X} - \mathbf{X}\mathbf{A}_i^T - \mathbf{B}_i\mathbf{M}_j + \mathbf{M}_j^T\mathbf{B}_i^T)\cos\theta \\ (\mathbf{X}\mathbf{A}_i^T - \mathbf{A}_i\mathbf{X} - \mathbf{B}_i\mathbf{M}_j - \mathbf{M}_j^T\mathbf{B}_i^T)\cos\theta & (\mathbf{A}_i\mathbf{X} + \mathbf{X}\mathbf{A}_i^T - \mathbf{B}_i\mathbf{M}_j - \mathbf{M}_j^T\mathbf{B}_i^T)\sin\theta \end{pmatrix},$$

$$\mathbf{\Gamma}_{ij}^3 = \begin{pmatrix} -r\mathbf{X} & \mathbf{A}_i\mathbf{X} - \mathbf{B}_i\mathbf{M}_j \\ \mathbf{X}\mathbf{A}_i^T - \mathbf{M}_j^T\mathbf{B}_i^T & -r\mathbf{X} \end{pmatrix} \qquad \text{(C.37)}$$

$$\mathbf{X} \succ 0$$

Falls eine Lösung für die obere LMI exisitiert, so ergibt sich die Zustandsrückführung zu

$$\mathbf{K}_j = \mathbf{M}_j\,\mathbf{X}^{-1}$$

wobei die Eigenwerte des geschlossenen Kreises in der Polstellenregion $\mathcal{S}(\alpha, r, \theta)$ liegen