Lehrstuhl für Mathematische Systemtheorie
Universität Stuttgart

BACHELORARBEIT

im Bachelorstudiengang
Mathematik

TITEL DER ARBEIT

Erstgutachter/in:	Prof. Dr. Carsten W. Scherer
Zweitgutachter/in:	....

vorgelegt von

Name
Straße
PLZ + Ort
E-mail

Studienfach
Fachsemester
Matrikelnummer
Abgabetermin: XX.XX.201X

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

O.B.d.A ...

Symbolverzeichnis

$G(s)$	Übertragungsfunktion eines LTI Systems
(A, B, C, D)	Realization des Systems im Zustandsraum

Kapitel 1

Einleitung

Die Einleitung gliedert sich in den 3 Unterpunkten:

1.1 Motivation

Die Einleitung muss den Leser zum weiterlesen bewegen. Hilfreich sind hier manchmal konkrete Beispiele für Anwendungen. Die Problemstellung motivieren, erklären warum die Problemstellung behandelt wird und wofür eine Lösung des Problems wichtig ist.

1.2 Zielsetzung

Motivieren Sie Ihr Thema und geben Sie deutlich die Problemstellung an! Formulieren Sie zu Beginn der Arbeit die Forschungsfrage/n, die Sie in der Arbeit beantworten wollen. In der Regel haben Sie eine Hauptfrage. Zur Unterstützung können dann verschiedene Teilfragen formuliert werden.

1.3 Aufbau der Arbeit

Am Ende der Einleitung muss die Struktur der Arbeit kurz erläutert werden. Die Arbeit ist in x Kapiteln aufgebaut. Im ersten Kapitel wird die Problemstellung ausführlich formuliert. Außerdem werden die in der Literatur vorhandene Verfahren zur Lösung des Problems diskutiert.
Das zweite und dritte Kapitel bilden zusammen der Kern dieser Arbeit. Dort wird das Lösungsansatz vorgestellt und erklärt..... Im letzten Kapitel werden die vorgestellten Algorithmen und Ansätze durch Simulationsergnisse validiert....

Kapitel 2

Stand der Technik

Es ist wichtig, dass die Literatur studiert wird. In der Literatur angebotene Lösungsansätze sollten kurz diskutiert und besprochen werden und die Wahl des Verfahrens für die Bachelorarbeit sollte begründet werden.

Kapitel 3

Grundlagen und Definitionen

3.1 Model of Dynamical System and Frequency Domain

$S(A, B, C, D)$
$G(A, B, C, D)$

$$x(t+1) = Ax(t) + Bu(t),\ t = 0, 1, 2 \ldots, N-1 \tag{3.1}$$

$$y(t) = Cx(t) + Du(t),\ t = 0, 1, 2 \ldots, N \tag{3.2}$$

3.2 Supervectors and Matrix Model

3.2.1 Definition of Matrix Model

3.2.2 Properties of Matrix Model

Kapitel 4

Unit Memory Algorithm

Mit im chapter 3 eingefügten Bezeichnungen, betrachten wir lineares Input/Output Modell

$$y = Gu + d. \tag{4.1}$$

Bezeichnet man mit $r \in \mathbb{R}^m$ den Referenzsignal, so kann das Problem wie folgt formuliert werden:
Finde input $u_\infty \in \mathbb{R}^l$, sodass

$$\begin{aligned} r &= y_\infty \\ \text{mit } y_\infty &= Gu_\infty + d. \end{aligned} \tag{4.2}$$

Da es in der Praktik meistens nicht möglich ist, eine perfekte Lösung zu finden, stellen wir die folgende Forderung auf unsere gewünschte Lösung auf:

$$\forall \varepsilon > 0 : \; ||r - y_\infty|| < \varepsilon. \tag{4.3}$$

Wir wollen nun ein Algorithmus entwerfen, mit dem wir das u_∞ iterativ annähren können. Wir bezeichnen mit $u_0 \in \mathbb{R}^l$ das Anfangsinputsignal und nehmen an, dass $d \in \mathbb{R}^m$ für jede Iteration gleich ist.
Sei $k = 0, 1, 2, \ldots$ der Iterationsindex für die Anzahl der vergangenen Iterationen. Wir bezeichnen mit u_k das Inputsignal bzw. mit y_k das Outputsignal beim Schritt k, wobei die Iteration erfogt nach folgendem Gesetz:

$$y_k = Gu_k + d, \; k \geq 0. \tag{4.4}$$

Dann können wir für jeden Schritt den Fehler

$$e_k := r - y_k, \; k \geq 0 \tag{4.5}$$

definieren. Da wir unser Fehler verkleinen wollen, soll jeder neues u_k sowohl vom Output y_k als auch von den Fehler e_k, e_{k+1} abhängen. So formulieren wir **Unit Memory Algorithm**:

$$u = f(e_k, e_{k+1}, u_k), \tag{4.6}$$

wobei $f : \mathbb{R}^m \times \mathbb{R}^m \times \mathbb{R}^l \to \mathbb{R}^l$ ist eine von k unabhängige Funktion.
Ein besonderer Fall, der Wert zu betrachten ist, ist der Fall in dem wir f gleich einer linearen Funktion setzen. Dann bekommt man folgendes Iterationsgesetz:

$$\begin{aligned} u_{k+1} &= u_k + K_0 e_k + K_1 e_{k+1} \\ y_k &= G u_k + d. \end{aligned} \tag{4.7}$$

Hier bezeichnen K_0 und K_1 Matrizen passenden Dimensionen. Die Wahl von den Matrizen ist der entscheidende Punkt, denn diese direkte Auswirkung auf die Konvergenz und Performance des Systems haben.
In folgenden Unterkapiteln untersuchen wir die Performance- und Konvergenzeigenschaften, sowie die Robustheit des erstellten Models. Wir treffen eine wichtige Annahme, nähmlich dass unser Fehler monoton abnimmt:

$$||e_{k+1}|| < ||e_k|| \text{ for all } k \geq 0. \tag{4.8}$$

Diese Annahme wollen wir im nächsten Kapitel diskutieren.

4.1 Monotone Konvergenz

4.2 Konvergenzkriterien

4.2.1 Konvergenzkriterien für State Space Model

Wir betrachten also den Iterationsgesetz 4.7 und wollen Kriterien an die Matrizen K_0 und K_1 stellen, sodass gewünschte Konvergenz des Fehlers gewährleistet wird.
Aus $Gu = y - d = r - e - d$ bekommen wir

$$e_{k+1} = e_k - GK_0 e_k - GK_1 e_{k+1}, \tag{4.9}$$

und, unter der Annahme, dass $(I + GK_1)$ nicht singular ist, ergibt sich Iterationsgesetz für den Fehlerterm:

$$\begin{aligned} e_{k+1} &= L e_k, \\ \text{mit } L &= (I + GK_1)^{-1}(I - GK_0) \end{aligned} \tag{4.10}$$

und neuer üpdateGesetz für Inputsignal:

$$u_{k+1} = u_k + (K_0 + K_1 L)e_k. \tag{4.11}$$

Die weitere Beobachtung ist, dass die Evolution des Fehlers ergibt sich zu

$$e_k = L^k e_0 \text{ for all } k \geq 0. \tag{4.12}$$

Hieraus kann man folgen, das das notwendige und ausrechende Kriterium für die Konvergenz des Fehlers gegen 0 ist $\rho(L) < 1$, wobei $\rho(L)$ das Spektralradius von L bezeichnet.
Außerdem, aus der Gleichung folgt, dass die Konvergenz bzw. die Schnelle der Konvergenz hängt von den Matrizen G, K_0 und K_1 ab, sowie vom Anfangsfehler e_0. Es ist somit gar nicht einfach, passende Matrizen zu finden. Folgendes Theorem ermöglich dennoch in manchen Fällen sofort die Antwort auf die Frage zu geben, ob dies überhaupt möglich ist.

Theorem 1 *Sprektralradius vom State Space System*

Für ein $m-$input $m-$output diskretes System $S(A, B, C, D)$ und dazugehöriges Operator L, definiert wie oben, auf einem endlichen Zeitintervall gilt: Spektrum von L ist genau gleich der Menge der Eigenwerte von D. Mit anderen Worten:

$$\rho(L) = \rho(D). \tag{4.13}$$

Proof:
TODO Proof ■

Es ist klar, dass die Monotonieeigenschaft der Folge $\{e_k\}_{k\geq 0}$ hängt von der gewälten Norm aus. Dennoch für endlichen Vektorräumen lässt sich dieses Problem umgehen: es reicht aus, nur eine Norm $||\cdot||$ in $\mathbb{R}^m$ zu finden, in der monotone Konvergenz gewährleistet wird. Dann konvergiert die Folge auch in jeder anderen Norm. Dazu gibt es folgender Satz:

Theorem 2

THEOREM 5.6

4.2.2 Relaxation Methods

TODO: Relaxation methods $-(-\_-)-$

4.3 Robustheit

Man möchte die Konvergenz auch dann gewährleisten, wenn das wirkliche Systemsverhalten sich von vom erstellten Modell vorgesagtem Verhalten abweicht. In dargestellten Modell kann man zwei Unsicherheiten einfach sehen, nähmlich

(a) Unsicherheit in dem Wert des Parameters im Plant Modell und

(b) Unsicherheit im Struktur vom Plant.

Parameterunsicherheit kann man ausdrücken dadurch, dass man für ein ungenaues Parameter p Intervallgrenzen angibt und betrachtet $p \in [p_l, p_u]$, wobei p_l und p_u sind bekannte untere und obere Intervallgrenzen. Die Strukturunsicherheit kann zwar viele Formen bei den praktischen Anwendungen annehmen, aber zum Zielen der theoretischen Analyse wird oft als additiv oder multiplikativ angenommen. Mit anderen Worten, falls G stellt Plant Model dar, dann die Unsicherheitsmodelle können als

$$G + \Delta G \text{ (additive Unsicherheitsmodel)}, \tag{4.14}$$
$$UG \text{ (linke multiplikative Unsicherheitsmodel)}, \tag{4.15}$$
$$GU \text{ (rechte multiplikative Unsicherheitsmodel) }, \tag{4.16}$$
$$G^{-1} + V \text{ (inverse additive Unsicherheitsmodel)}, \tag{4.17}$$

mit $\Delta G \in \mathbb{R}^{l \times m}$, $U \in \mathbb{R}^{l \times l}$ und $V \in \mathbb{R}^{m \times l}$.
Um formale Aussagen zu treffen, führt man folgende Definition ein:

Definition 3 *Für ein lineares Model $y = Gu + d$ und dazugehöriger Algorythmus für Input Update sei monotone Konvergenz des Fehlers gegen Null erfüllt:*

$$||e_{k+1}|| < ||e_k|| \text{ und } \lim_{k \to \infty} e_k = 0. \tag{4.18}$$

Wenn die Unsicherheit ist durch den Modelfehler U dargestellt wird, das resultierende ILC System heißt monoton konvergent mit Robustheit im Bezug auf Modelfehler U, falls die Eigenschaft der Monotone Konvergenz 4.18 bleibt erhalten.

Wie es im Unterkapietel 4.2 besprochen wurde, reicht $\rho(L) < 1$ für die Konvergenz des Fehlers gegen Null. Setzt man in $L = (I + GK_1)^{-1}(I - GK_0)$ $K_1 = 0$, so muss $\rho(I - GK_0) < 1$ gelten. Dennnoch bei der Anwendung des Modells auf das reeles Plant $G + \Delta G$, die Matrix L wird zu $I - (G + \Delta G)K_0 = L + \Delta L$ mit $\Delta L = -\Delta G K_0$. Nun Konvergenz ist dann gewährleistet, wenn $||L + \Delta L|| < 1$. $||L + \Delta L|| \leq ||L|| + ||\Delta L||$ schließt den Beweis des folgenden Satzes ab:

Theorem 4

Mit der Notation von oben, ein ausreichendes Kriterium für die monotone Konvergenz mit Robustheit für den ALgorithmus $u_{k+1} = u_k + K_0 e_k$ in der Sicht der additiven Unsicherheit ist

$$||\Delta L|| = ||\Delta G K_0|| < 1 - ||L||. \tag{4.19}$$

Mit dieser einfachen Normenungleichung kann man sofort den Grad der Robustheit des Systems sehen, die immer dann gewährleistet werden kann, wenn $||L|| < 1$.

4.3.1 Robustheit in der Sicht der Zustandsunsicherheit

Es sei nun das Model exakt, aber der Anfangswert bei jeder Iteration kann abweichen. Da im betrachtenden Modell $y = Gu + d$ nur der Term d von dem Anfangswert x_0 abhängt, kann man diese Unsicherheit durch das Ersetzen von d durch einen von dem Iterationsschritt k abhängigen Term d_k. Aus $u_{k+1} = u_k + K_0 e_k + K_1 E_{k+1}$ und dem Modell $y_k = Gu_k + d_k$ neuer Verlauf des Fehlers wird zu

$$e_{k+1} = Le_k + (I + GK_1)^{-1}(d_k + d_{k+1}),\ k \geq 0. \tag{4.20}$$

Kapitel 5

Zusammenfassung und Ausblick

Hier können auch Schlußfolgerungen präsentiert und ein Ausblick gegeben werden.

Literaturverzeichnis

L. Lessard, S. Lall. Proceedings of the IEEE Conference on Decision and Control, p. 1621–1626, 2009.

.1 Beweise

.1.1 Beweis 1

$$1 + 1 = 2 \tag{1}$$

Eidesstattliche Erklärung

Ich erkläre, dass ich meine Bachelor-Arbeit [*Titel der Arbeit*] selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe und dass ich alle Stellen, die ich wörtlich oder sinngemäß aus Veröffentlichungen entnommen habe, als solche kenntlich gemacht habe. Die Arbeit hat bisher in gleicher oder ähnlicher Form oder auszugsweise noch keiner Prüfungsbehörde vorgelegen.
Stuttgart, den XX.XX.2013

____

(Name des Kandidaten)