

Überprüfung perspektivischer Beobachtbarkeit linearer zeitinvarianter Systeme

Masterarbeit
an der
Technischen Universität Graz

vorgelegt von
Martin Stadler

ausgeführt am
Institut für Regelungs- und Automatisierungstechnik

Mai 2010



Danksagung

Ich bedanke mich an dieser Stelle bei meinen Eltern, die mir mein Studium überhaupt ermöglicht haben. Weiters bedanke ich mich bei meinem Betreuer Professor Nicolaos Dourdoumas und bei Sonja Moschik, die mich beide hervorragend durch die Arbeit begleitet haben und immer für Fragen und Diskussionen zur Verfügung standen. Ganz besonders bedanke ich mich bei meiner Verlobten Eva, daß sie mich in dieser Zeit ausgehalten und unterstützt hat.

Kurzfassung

In der vorliegenden Arbeit wird die perspektivische Beobachtbarkeit betrachtet. Nach derzeitigem Stand der Forschung sind notwendige und hinreichende Bedingungen bekannt, unter denen ein lineares zeitinvariantes System perspektivisch beobachtbar ist. Allerdings sind alle bekannten Bedingungen schwer zu überprüfen. Es wird gezeigt, daß bei einer bekannten Bedingung die Überprüfung von unendlich vielen Parametern auf eine endliche Menge reduzierbar ist und somit die Überprüfung in der Praxis verwendbar wird.

Es wird eine alternative Formulierung vorgestellt und es werden zusätzliche hinreichende Bedingungen gezeigt. Weiters wird die numerische Stabilität der Überprüfung untersucht und es wird ein numerisch stabiler Algorithmus vorgestellt.

Die Arbeit schließt mit einer Definition für ein Maß für die perspektivische Beobachtbarkeit, um die Eigenschaft quantitativ beurteilen zu können.

Schlagwörter: perspektivische Beobachtbarkeit, Rangermittlung, Eigenvektor-Test, numerische Minimierung, Simplex-Verfahren, perspektivisches Beobachtbarkeitsmaß

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

Graz, am
(Unterschrift)

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen	3
2.1	Lineare zeitinvariante Systeme	3
2.2	Reguläre Zustandstransformation	4
2.2.1	Modaltransformation	4
2.3	Steuerbarkeit	5
2.4	Beobachtbarkeit	6
2.5	Dualität von Steuerbarkeit und Beobachtbarkeit	7
2.6	Perspektivische Beobachtbarkeit	7
2.6.1	Beispiele perspektivischer Systeme	8
2.6.2	Einschränkungen dieser Arbeit	12
2.6.3	Bekannte Ergebnisse	12
2.6.4	Derzeitiger Wissensstand	14
2.6.5	Alternative Formulierung	14
2.6.6	Invarianz gegenüber regulären Zustandstransformationen	15
2.7	Steuerbarkeitsmaße	16
2.8	Beobachtbarkeitsmaße	16
3	Überprüfung perspektivischer Beobachtbarkeit	17
3.1	Rangermittlung	17
3.2	Kandidaten für die Überprüfung	20
3.3	Alternatives Kriterium	20
3.3.1	Kommutivität eines Matrizenproduktes	21
3.4	Ergebnis	24
4	Hinreichende Bedingungen perspektivischer Beobachtbarkeit	25
4.1	Systeme mit skalarem Ausgang	25
4.2	Die Ausgangsmatrix besitzt Rang 1	26
4.3	Die Ausgangsmatrix besitzt Rang n	26
4.4	Zusammenhänge mit der Beobachtbarkeit	27

4.4.1	Nicht beobachtbare Systeme	27
4.4.2	Perspektivisch beobachtbare Systeme	27
4.5	Systeme in Modalform	28
4.5.1	Ausgangsmatrix mit einer Nullspalte	28
4.5.2	Zwei linear abhängige Spalten in der Ausgangsmatrix	29
5	Numerisch stabile Überprüfung der perspektivischen Beobachtbarkeit	31
5.1	Numerische Probleme bei der Überprüfung der Rangbedingung	31
5.2	Formulierung als Minimierungsproblem	32
5.3	Eigenschaften des Minimierungsproblems	33
5.3.1	Beispiele	33
5.4	Lösungsansätze	36
5.4.1	Test nach Gao und Neumann	36
5.4.2	Numerische Optimierungsverfahren 1. und 2. Ordnung	37
5.4.3	Genetische Algorithmen	37
5.4.4	Simplex-Verfahren	42
5.4.5	Simplex-Verfahren mit „intelligenten“ Startpunkten	42
5.4.6	Simplex-Verfahren mit zufällig gewählten Startpunkten	44
5.5	Algorithmus	44
6	Maß für die perspektivische Beobachtbarkeit	47
6.1	Steuerbarkeitsmaß nach Paige	47
6.1.1	Übertragung auf die perspektivische Beobachtbarkeit	48
6.2	Steuerbarkeitsmaß nach Eising	48
6.2.1	Übertragung auf die perspektivische Beobachtbarkeit	49
6.2.2	Berechnung des Maßes für die perspektivische Beobachtbarkeit	51
6.2.3	Beispiel	52
6.2.4	Probleme	52
7	Zusammenfassung und Ausblick	55
A	Implementierung der Algorithmen	57
A.1	Überprüfung perspektivischer Beobachtbarkeit	57
A.2	Überprüfung der Eigenvektor-Bedingung	59
A.3	Ermittlung des perspektivischen Beobachtbarkeitsmaßes	61
A.4	Testumgebung	65

Kapitel 1

Einleitung

In der vorliegenden Arbeit wird die perspektivische Beobachtbarkeit betrachtet. Das Konzept wird von Grund auf vorgestellt und sich daraus ergebende Fragen werden behandelt.

In Kapitel 2 werden die Grundlagen angeführt, die in der vorliegenden Arbeit verwendet werden. Dabei wird vorausgesetzt, daß die mathematischen Konzepte, vor allem der Linearen Algebra, bekannt sind. Die verwendeten Konzepte der Systemtheorie werden hingegen erklärt.

In Kapitel 3 wird gezeigt, wie in der Praxis überprüft werden kann, ob ein System die Eigenschaft perspektivische Beobachtbarkeit besitzt.

Verschiedene hinreichende Bedingungen zur Überprüfung der perspektivischen Beobachtbarkeit werden in Kapitel 4 gezeigt. Anhand dieser Bedingungen kann in bestimmten Spezialfällen die perspektivische Beobachtbarkeit sehr einfach und effizient überprüft werden.

In Kapitel 5 wird die numerische Stabilität der Überprüfung der perspektivischen Beobachtbarkeit untersucht und die auftretenden Probleme behandelt.

Abschließend wird in Kapitel 6 versucht aus bereits bekannten Definitionen für Steuerbarkeitsmaße ein Maß für die perspektivische Beobachtbarkeit abzuleiten und zu berechnen.

Im Anhang A finden sich Implementierungen der in dieser Arbeit entwickelten Algorithmen.

Kapitel 2

Grundlagen

In diesem Kapitel werden die grundlegenden Konzepte, mit denen man sich in der vorliegenden Arbeit beschäftigt, vorgestellt. Vergleiche dazu [1]. Aus Gründen der Einfachheit betrachtet man nur lineare zeitinvariante Systeme. Die verwendeten Konzepte sind zwar auch für nichtlineare bzw. zeitvariante Systeme definiert, diese Systeme sind allerdings mathematisch sehr viel schwieriger zu behandeln.

2.1 Lineare zeitinvariante Systeme

Die zu untersuchenden Systeme werden in der Formulierung

$$\frac{dx}{dt} = \mathbf{A}x(t) + \mathbf{B}u(t) \quad (2.1)$$

$$y(t) = \mathbf{C}x(t) + \mathbf{D}u(t) \quad (2.2)$$

angegeben¹. Der Zustandsvektor x besteht aus n Zustandsvariablen, die den inneren Systemzustand beschreiben. Der Eingangsvektor u besteht aus m Eingangsgrößen und der Ausgangsvektor y aus p Ausgangsgrößen. Die vier Matrizen sind konstant und haben die dazu passenden Dimensionen: die Systemmatrix \mathbf{A} ist eine $(n \times n)$ -Matrix, die Eingangsmatrix \mathbf{B} ist eine $(n \times m)$ -Matrix, die Ausgangsmatrix \mathbf{C} ist eine $(p \times n)$ -Matrix und die Durchgriffsmatrix \mathbf{D} ist eine $(p \times m)$ -Matrix. Üblicherweise wird $\mathbf{D} = 0$ angenommen und für eine kompaktere Schreibweise werden zwei Vereinfachungen getroffen: Die zeitliche Ableitung des Zustandsvektors wird mit \dot{x} bezeichnet und die zeitlichen Abhängigkeiten der Größen u, x, y werden nicht explizit angeschrieben. Dadurch ergibt sich die bekannte Darstellung

$$\dot{x} = \mathbf{A}x + \mathbf{B}u \quad (2.3)$$

$$y = \mathbf{C}x \quad (2.4)$$

¹Zur besseren Lesbarkeit werden Matrizen fett gedruckt dargestellt.

für ein lineares zeitinvariantes System.

2.2 Reguläre Zustandstransformation

Die Wahl der Zustandsvariablen ist bei der Darstellung eines Systems nach (2.3) und (2.4) nicht eindeutig. Durch Transformation kann quasi das Koordinatensystem der Variablen verändert werden.

Ausgehend vom Zustandsvektor x kann ein neuer Zustandsvektor $x = \mathbf{T}z$ eingeführt werden. Um eine eindeutige Zuordnung zu gewährleisten muß die $(n \times n)$ -Matrix \mathbf{T} regulär sein. Ferner wurde sie als konstant gewählt. Gleichung (2.3) wird zu

$$\mathbf{T}\dot{z} = \mathbf{A}\mathbf{T}z + \mathbf{B}u \quad | \cdot \mathbf{T}^{-1} \text{ von links} \quad (2.5)$$

$$\dot{z} = \underbrace{\mathbf{T}^{-1}\mathbf{A}\mathbf{T}}_{\mathbf{A}_z} z + \underbrace{\mathbf{T}^{-1}\mathbf{B}}_{\mathbf{B}_z} u \quad (2.6)$$

und Gleichung (2.4) wird zu

$$y = \underbrace{\mathbf{C}\mathbf{T}}_{\mathbf{C}_z} z \quad (2.7)$$

Es ist nun ein äquivalentes System mit z anstatt x als Zustandsvektor entstanden:

$$\dot{z} = \mathbf{A}_z z + \mathbf{B}_z u \quad (2.8)$$

$$y = \mathbf{C}_z z \quad (2.9)$$

Relevante Eigenschaften eines Systems bleiben bei einer regulären Zustandstransformation erhalten bzw. Kenngrößen wie zum Beispiel die Gewichtsfunktion oder die Übertragungsfunktion verändern sich nicht.

2.2.1 Modaltransformation

Unter den beliebig vielen Zustandstransformationen gibt es einen besonders interessanten Spezialfall. Dabei wird das System so umgewandelt, daß nach der Transformation die einzelnen Differentialgleichungen in entkoppelter Form vorliegen. Das bedeutet, daß die zeitliche Ableitung einer Zustandsvariable nur mehr von der Zustandsvariable selbst und von den Eingängen des Systems abhängig ist, nicht aber von den anderen Zustandsvariablen.

Wählt man für die Spalten der Transformationsmatrix Rechtseigenvektoren p_i der

Systemmatrix \mathbf{A} , so ergibt sich folgendes System:

$$\dot{z} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P}z + \mathbf{P}^{-1}\mathbf{B}u \quad (2.10)$$

$$y = \mathbf{C}\mathbf{P}z \quad (2.11)$$

$$\text{mit } \mathbf{P} = \left[\begin{array}{c|c|c|c} p_1 & p_2 & \cdots & p_n \end{array} \right]$$

Um zu kennzeichnen, daß das System modaltransformiert vorliegt, vergibt man für die entstehenden Matrizen neue Symbole:

$$\dot{z} = \mathbf{\Lambda}z + \boldsymbol{\delta}u \quad (2.12)$$

$$y = \bar{\boldsymbol{\delta}}z \quad (2.13)$$

Die Differentialgleichungen des Systems liegen nun in entkoppelter Form vor, das heißt die neu entstandene Systemmatrix $\mathbf{\Lambda}$ besitzt Diagonalfom; die Elemente s_i in der Hauptdiagonale sind die Eigenwerte der Systemmatrix \mathbf{A} .

$$\mathbf{\Lambda} = \left[\begin{array}{cccc} s_1 & & & 0 \\ & s_2 & & \\ & & \ddots & \\ 0 & & & s_n \end{array} \right] \quad (2.14)$$

Um die Transformation durchführen zu können, muß \mathbf{P} regulär sein. Dies ist genau dann der Fall, wenn \mathbf{A} n linear unabhängige Eigenvektoren besitzt. Sind alle Eigenwerte von \mathbf{A} verschieden ($s_i \neq s_j \forall i, j, i \neq j$), dann besitzt \mathbf{A} jedenfalls n linear unabhängige Eigenvektoren. Bei mehrfachen Eigenwerten ist dies nicht mehr garantiert; hier kann es weniger als n linear unabhängige Eigenvektoren geben und das System kann nicht modaltransformiert werden. Systeme, die modaltransformiert werden können, werden auch **diagonalisierbar** genannt.

Ein weiterer interessanter Spezialfall ist die Transformation in die Jordan-Blockform. Diese Transformation ist immer möglich, auch bei mehrfachen Eigenwerten. Die transformierte Systemmatrix besitzt dann so genannte Jordan-Blöcke in der Hauptdiagonale, die die mehrfachen Eigenwerte repräsentieren. Sind alle Eigenwerte verschieden, so ist die Jordan-Blockform zur Diagonalfom ident.

2.3 Steuerbarkeit

Der Begriff Steuerbarkeit wird wie folgt definiert:

Das System (2.3), (2.4) wird **steuerbar** genannt, wenn durch geeignete Wahl der Eingangsgröße u der Zustandsvektor x in endlicher Zeit T aus einem *beliebig* vorgegebenen Anfangszustand $x(0)$ in den *beliebig* vorgegebenen Endzustand $x(T)$ bewegt werden kann. [1, S. 78]

Daraus ergeben sich unter anderem die folgenden notwendigen und hinreichenden Kriterien:

1. Kriterium nach Kalman: Die Steuerbarkeitsmatrix \mathbf{S}_u besitzt vollen Rang.

$$\text{rang } \mathbf{S}_u = \text{rang} \begin{bmatrix} \mathbf{B} & \mathbf{A}\mathbf{B} & \dots & \mathbf{A}^{n-1}\mathbf{B} \end{bmatrix} = n \quad (2.15)$$

2. Rangkriterium nach Popov-Belevitch-Hautus: Die Matrix \mathbf{H}_u hat Rang n für alle $\lambda = s_i$, wobei s_i die Eigenwerte der Systemmatrix \mathbf{A} sind.

$$\text{rang } \mathbf{H}_u = \text{rang} \begin{bmatrix} \lambda \mathbf{I} - \mathbf{A} & \mathbf{B} \end{bmatrix} = n \quad (2.16)$$

3. Eigenvektor-Test nach Popov-Belevitch-Hautus: Kein Links-Eigenvektor von \mathbf{A} steht normal auf \mathbf{B} .

$$\rho^T \mathbf{A} = s \rho^T \text{ und } \rho^T \mathbf{B} = 0 \implies \rho = 0 \quad (2.17)$$

Für die Steuerbarkeit eines linearen zeitinvarianten Systems sind nur die System- und die Eingangsmatrix maßgeblich, die Ausgangsmatrix ist hingegen irrelevant. Daher spricht man im Zusammenhang mit der Steuerbarkeit oft nur von dem Matrizenpaar bzw. System (\mathbf{A}, \mathbf{B}) .

2.4 Beobachtbarkeit

Der Begriff Beobachtbarkeit wird wie folgt definiert:

Das System (2.3), (2.4) wird **beobachtbar** genannt, wenn aus der Kenntnis von $u(t)$ und $y(t)$ in einem *endlichen* Zeitintervall $[0, T]$, der *unbekannte* Anfangszustand $x(0)$ bestimmt werden kann. [1, S. 79]

Daraus ergeben sich unter anderem die folgenden notwendigen und hinreichenden Kriterien:

1. Kriterium nach Kalman: Die Beobachtbarkeitsmatrix \mathbf{B}_y besitzt vollen Rang.

$$\text{rang } \mathbf{B}_y = \text{rang} \begin{bmatrix} \mathbf{C} \\ \mathbf{C}\mathbf{A} \\ \vdots \\ \mathbf{C}\mathbf{A}^{n-1} \end{bmatrix} = n \quad (2.18)$$

2. Rangkriterium nach Popov-Belevitch-Hautus: Die Matrix \mathbf{H}_y hat Rang n für alle $\lambda = s_i$, wobei s_i die Eigenwerte der Systemmatrix \mathbf{A} sind.

$$\text{rang } \mathbf{H}_y = \text{rang} \begin{bmatrix} \lambda \mathbf{I} - \mathbf{A} \\ \mathbf{C} \end{bmatrix} = n \quad (2.19)$$

3. Eigenvektor-Test nach Popov-Belevitch-Hautus: Kein Rechts-Eigenvektor von \mathbf{A} steht normal auf \mathbf{C} .

$$\mathbf{A}p = sp \text{ und } \mathbf{C}p = 0 \quad \implies \quad p = 0 \quad (2.20)$$

Für die Beobachtbarkeit eines linearen zeitinvarianten Systems sind nur die System- und die Ausgangsmatrix maßgeblich, die Eingangsmatrix ist hingegen irrelevant. Daher spricht man im Zusammenhang mit der Beobachtbarkeit oft nur von dem Matrizenpaar bzw. System (\mathbf{A}, \mathbf{C}) .

2.5 Dualität von Steuerbarkeit und Beobachtbarkeit

Die beiden Begriffe Steuerbarkeit und Beobachtbarkeit sind eng miteinander verbunden; man nennt sie dual zueinander.

Das System (\mathbf{A}, \mathbf{B}) ist genau dann steuerbar, wenn das System $(-\mathbf{A}^T, \mathbf{B}^T)$ beobachtbar ist. Die Umkehrung davon gilt ebenfalls: (\mathbf{A}, \mathbf{C}) ist genau dann beobachtbar, wenn $(-\mathbf{A}^T, \mathbf{C}^T)$ steuerbar ist.

2.6 Perspektivische Beobachtbarkeit

In [2] wird der Begriff perspektivische Beobachtbarkeit erstmals allgemein eingeführt:

Ein lineares zeitinvariantes System wird **perspektivisch beobachtbar** genannt, wenn aus der Kenntnis von $y(t)$ bis auf d Dimensionen in einem *endlichen* Zeitintervall $[0, T]$, der *unbekannte* Anfangszustand $x(0)$ bis auf d Dimensionen bestimmt werden kann.

Im Gegensatz zur Beobachtbarkeit ist hier der Ausgangsvektor nicht vollständig bekannt, sondern nur $p - d$ Dimensionen davon. Man kann sich das als Projektion eines p -dimensionalen Vektors auf $p - d$ Dimensionen vorstellen. Da nur diese Projektion des Ausgangsvektors bekannt ist, möchte man auch „nur“ eine Projektion des Anfangszustandes des Zustandsvektors auf $n - d$ Dimensionen ermitteln.

Damit überhaupt eine Projektion des Zustandsvektors existieren kann, wird im weiteren Verlauf der Arbeit davon ausgegangen, daß der Zustandsvektor mindestens zweidimensional ist ($n \geq 2$).

Systeme, bei denen eine Projektion des Ausgangsvektors bekannt ist, und bei denen es ausreichend ist, Projektionen der inneren Zustände zu kennen, werden als **perspektivische Systeme** bezeichnet. Solche wurden bereits zwei Jahre vorher in [3] vorgestellt, allerdings nur für den Spezialfall $d = 1$.

2.6.1 Beispiele perspektivischer Systeme

Ein paar Beispiele perspektivischer Systeme sollen das Konzept verdeutlichen:

Gefilmter Drehkörper

In [4] wird als Beispiel ein starrer dreidimensionaler Körper, der sich mit konstanter Winkelgeschwindigkeit um eine feste Achse dreht, verwendet. Ein Punkt (x_t, y_t, z_t) auf dem Körper unterliegt daher der Beziehung

$$\begin{pmatrix} \dot{x}_t \\ \dot{y}_t \\ \dot{z}_t \end{pmatrix} = \mathbf{\Omega} \begin{pmatrix} x_t \\ y_t \\ z_t \end{pmatrix} \quad (2.21)$$

$$\text{mit } \mathbf{\Omega} = \begin{bmatrix} 0 & \omega_1 & \omega_2 \\ -\omega_1 & 0 & \omega_3 \\ -\omega_2 & -\omega_3 & 0 \end{bmatrix}$$

Die Matrix $\mathbf{\Omega}$ ist eine konstante schief-symmetrische Rotationsmatrix. Der Körper wird mit einer Kamera gefilmt. Das entspricht geometrisch einer perspektivischen Projektion; jeder Punkt ungleich null ist, bis auf eine Linie durch den Ursprung des Koordinatensystems, eindeutig bestimmt. Mathematisch kann das mit der Funktion

$$Z : \mathbb{R}^3 - \{(0, 0, 0)\} \mapsto \mathbb{RP}^2 \quad (2.22)$$

ausgedrückt werden, die als

$$(x_t, y_t, z_t) \mapsto [x_t, y_t, z_t] \quad (2.23)$$

definiert ist. Die eckigen Klammern des Vektors stehen für die so genannten homogenen Koordinaten.

Für das Beispiel nimmt man an, daß der Projektionsschirm bei $z_t = 1$ steht. Jeder Punkt (x_t, y_t, z_t) mit $z_t \neq 0$ wird auf den Punkt $(x_t/z_t, y_t/z_t)$ am Schirm projiziert. Dieser Punkt am Schirm repräsentiert alle Punkte auf einer Linie durch den

Ursprung und den Punkt (x_t, y_t, z_t) . Der Aufbau des Beispiels wird durch Abbildung 2.1 illustriert. Das lineare System (2.21) und die Ausgangsfunktion (2.23) sind ein perspektivisches System.

In diesem Beispiel ist der dreidimensionale Ausgangsvektor in einer zweidimensionalen Projektion bekannt. Das System ist perspektivisch beobachtbar, wenn aus der Kenntnis der Projektion in einem endlichen Zeitintervall die Projektion des Anfangszustandes ermittelt werden kann.

Möchte man das Beispiel in die Form (2.3) und (2.4) bringen, so erhält man

$$\dot{x} = \Omega x \quad (2.24)$$

$$y = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} x \quad (2.25)$$

Die Ausgangsmatrix \mathbf{C} wird zur Einheitsmatrix, da eine Projektion des kompletten Zustandsvektors sichtbar ist.

Gefilmter bewegter Drehkörper

Die Verhältnisse werden komplexer, wenn sich der Drehkörper zusätzlich in einer Bewegung befindet [4]. Nun gibt es zwei verschiedene Koordinatensysteme. Das eine besteht aus den Körperkoordinaten (x_b, y_b, z_b) , die sich mit konstanter Geschwindigkeit bewegen:

$$\begin{pmatrix} \dot{x}_b \\ \dot{y}_b \\ \dot{z}_b \end{pmatrix} = Z \quad (2.26)$$

$$\text{mit } Z = \begin{pmatrix} \zeta_x \\ \zeta_y \\ \zeta_z \end{pmatrix} = \text{konstant}$$

Das zweite Koordinatensystem sind die Kamerakoordinaten. Diese rotieren in Bezug auf die Körperkoordinaten:

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = e^{\Omega t} \begin{pmatrix} x_b \\ y_b \\ z_b \end{pmatrix} \quad (2.27)$$

Faßt man nun beide Bewegungen zusammen, so erhält man

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = e^{\Omega t} \left[\begin{pmatrix} x_c(0) \\ y_c(0) \\ z_c(0) \end{pmatrix} + \begin{pmatrix} \zeta_x \\ \zeta_y \\ \zeta_z \end{pmatrix} t \right] \quad (2.28)$$

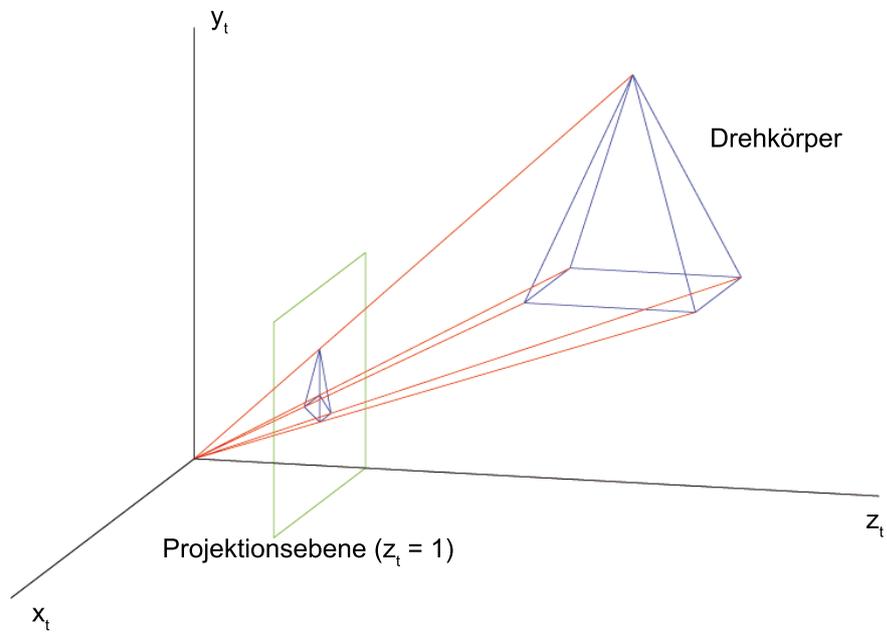


Abbildung 2.1: Gefilmter Drehkörper

Es ist bekannt, daß die Gleichung

$$x(t) = e^{\mathbf{A}t}x(0) \quad (2.29)$$

die Lösung der Differentialgleichung

$$\dot{x} = \mathbf{A}x \quad (\mathbf{A} \text{ konstant}) \quad (2.30)$$

ist. Substituiert man nun

$$Z = \Psi(0) \quad (2.31)$$

mit $\Psi = \begin{pmatrix} \psi_x \\ \psi_y \\ \psi_z \end{pmatrix}$

und multipliziert die Gleichung aus, so erkennt man, daß Gleichung (2.29) zweimal auftritt, einmal allerdings mit dem zusätzlichen Faktor t :

$$\begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} = e^{\Omega t} \begin{pmatrix} x_c(0) \\ y_c(0) \\ z_c(0) \end{pmatrix} + e^{\Omega t} \begin{pmatrix} \psi_x(0) \\ \psi_y(0) \\ \psi_z(0) \end{pmatrix} t \quad (2.32)$$

Daraus ergeben sich folgende zwei Systeme von Differentialgleichungen:

$$\begin{pmatrix} \dot{x}_c \\ \dot{y}_c \\ \dot{z}_c \end{pmatrix} = \Omega \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix} + \begin{pmatrix} \psi_x \\ \psi_y \\ \psi_z \end{pmatrix} \quad (2.33)$$

$$\begin{pmatrix} \dot{\psi}_x \\ \dot{\psi}_y \\ \dot{\psi}_z \end{pmatrix} = \Omega \begin{pmatrix} \psi_x \\ \psi_y \\ \psi_z \end{pmatrix} \quad (2.34)$$

Schließlich erhält man ein lineares zeitinvariantes System der Form

$$\begin{pmatrix} \dot{X}_c \\ \dot{\Psi} \end{pmatrix} = \begin{bmatrix} \Omega & \mathbf{I} \\ \mathbf{0} & \Omega \end{bmatrix} \begin{pmatrix} X_c \\ \Psi \end{pmatrix} \quad (2.35)$$

mit $X_c = \begin{pmatrix} x_c \\ y_c \\ z_c \end{pmatrix}$

für die Dynamik des Beispiels. Die Punkte am Körper werden wieder durch eine Kamera betrachtet, es entsteht die Beobachtungsfunktion

$$Z : \mathbb{R}^6 - B \mapsto \mathbb{RP}^2 \quad (2.36)$$

definiert durch

$$(X_c, \Psi) \mapsto [x_c, y_c, z_c] \quad (2.37)$$

$$\text{und } B = \{(X_c, \Psi) : X_c = 0\} \quad (2.38)$$

Die eckigen Klammern stehen hier wieder für homogene Koordinaten. Die Gleichungen (2.35) und (2.36) bilden ein perspektivisches System.

In diesem Beispiel ist der sechsdimensionale Ausgangsvektor in einer zweidimensionalen Projektion bekannt. Das System ist perspektivisch beobachtbar, wenn aus der Kenntnis der Projektion des Ausgangs in einem endlichen Zeitraum eine zweidimensionale Projektion des Anfangszustandes ermittelt werden kann.

Bringt man die Ausgangsfunktion (2.36) in die Form (2.4), so ist sie als

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{pmatrix} X_c \\ \Psi \end{pmatrix} \quad (2.39)$$

darstellbar, von der man nur eine Projektion auf zwei Dimensionen kennt.

Systeme mit rationalem Ausgang

In [5] wird eine weitere Beispiel-Klasse vorgestellt. Betrachtet man Systeme, zum Beispiel dritter Ordnung, mit einer rationalen Funktion am Ausgang, also

$$\dot{x} = \mathbf{A}x + \mathbf{B}u \quad (2.40)$$

$$y = \frac{c_{11}x_1 + c_{12}x_2 + c_{13}x_3}{c_{21}x_1 + c_{22}x_2 + c_{23}x_3} \quad (2.41)$$

so kann man den Ausgang y als Steigung des Vektors $(y_1, y_2)^T$ betrachten:

$$\begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \mathbf{C} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad (2.42)$$

$$\text{mit } \mathbf{C} = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \end{bmatrix}$$

Oder anders formuliert, man kennt die Projektion $y = \frac{y_1}{y_2}$. Perspektivisch beobachtbar bedeutet nun, daß anhand der Kenntnis von $y(t)$ eine zweidimensionale Projektion des Anfangszustandes x_0 , oder wiederum anders formuliert, die Richtung des Anfangszustandes, ermittelt werden kann.

2.6.2 Einschränkungen dieser Arbeit

Im weiteren Verlauf dieser Arbeit wird die perspektivische Beobachtbarkeit mit zwei durch die Praxis motivierten Einschränkungen betrachtet:

1. Es wird nur der reelle Fall betrachtet. Sowohl die Systeme ($\mathbf{A} \in \mathbb{R}^{(n \times n)}$, $\mathbf{C} \in \mathbb{R}^{(p \times n)}$) als auch die Projektionen sind reell. Bei praktischen Problemen ist dies fast immer der Fall.
2. Es wird nur der Fall $d = 1$ betrachtet, da bereits dieser „einfache“ Fall eine Reihe von Schwierigkeiten bietet.

2.6.3 Bekannte Ergebnisse

In [4] wird ebenfalls der Fall $d = 1$ betrachtet. Es werden also, wiederum anders formuliert, Linien im n -dimensionalen Zustandsraum auf Linien im p -dimensionalen Ausgangsraum abgebildet. Als Ergebnis wird vorgestellt, daß ein System über die komplexen Zahlen \mathbb{C} genau dann perspektivisch beobachtbar ist, wenn

$$\text{rang} \begin{bmatrix} (s_i \mathbf{I} - \mathbf{A})(s_j \mathbf{I} - \mathbf{A}) \\ \mathbf{C} \end{bmatrix} = n \quad (2.43)$$

für alle Eigenwerte s_i, s_j ($i = 1, \dots, n; j = 1, \dots, n$) von \mathbf{A} gilt. Über die reellen Zahlen \mathbb{R} ist diese Bedingung nur hinreichend. Das heißt, wenn die Bedingung erfüllt ist, dann ist das System perspektivisch beobachtbar. Ist die Bedingung allerdings nicht erfüllt, so kann über die perspektivische Beobachtbarkeit des Systems keine Aussage getroffen werden.

In [6] wird der allgemeinere Fall mit der Unkenntnis von d Dimensionen betrachtet. Hier wird das Ergebnis (2.43) verallgemeinert: Ein System ist perspektivisch auf einem d -dimensionalen Unterraum beobachtbar wenn

$$\text{rang} \begin{bmatrix} (s_0\mathbf{I} - \mathbf{A}) & \dots & (s_d\mathbf{I} - \mathbf{A}) \\ \mathbf{C} & & \end{bmatrix} = n \quad (2.44)$$

für alle Eigenwerte s_0, \dots, s_d von \mathbf{A} gilt. Daraus ergeben sich 2^d Überprüfungen für d Dimensionen. Diese Bedingung ist in verschiedenen Fällen unterschiedlich stark:

- Im Fall komplexer Projektionen ist sie hinreichend und notwendig,
- im Fall $d = 0$ ist sie das Rangkriterium nach Popov-Belevitch-Hautus und ebenfalls hinreichend und notwendig,
- im Fall $d = 1$ ist sie Bedingung (2.43),
- für den Fall, daß \mathbf{A} lauter reelle Eigenwerte besitzt, ist sie auch hinreichend und notwendig
- und für alle restlichen Fälle ist sie nur hinreichend.

In [5] wird wiederum nur der Spezialfall $d = 1$ betrachtet. Dort wird als Resultat angeführt, daß ein System über die reellen Zahlen genau dann perspektivisch beobachtbar ist, wenn

$$\text{rang} \begin{bmatrix} (\lambda_1\mathbf{I} - \mathbf{A})(\lambda_2\mathbf{I} - \mathbf{A}) \\ \mathbf{C} \end{bmatrix} = n \quad (2.45)$$

für alle λ_1, λ_2 aus den reellen und konjugiert komplexen Zahlen gilt.

Dieses Resultat wurde 2009 in [7] durch folgendes Beispiel widerlegt:

$$\mathbf{A} = \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 2 & -1 \\ 0 & 0 & 1 & 2 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \quad (2.46)$$

Die Bedingung (2.45) ist für alle reellen und konjugiert komplexen Zahlen erfüllt, doch trotzdem ist das System *nicht* perspektivisch beobachtbar.

Als Ergebnis wird angeführt, daß die Aussage aus [5] nur dann korrekt ist, wenn gewisse zusätzliche Bedingungen für \mathbf{A} gelten: \mathbf{A} muß *stark regulär* sein und die *Irrationalitätsbedingung* erfüllen.

Es seien $s_1, \dots, s_r \in \mathbb{C}$ die unterschiedlichen reellen oder komplexen Eigenwerte von $\mathbf{A} \in \mathbb{R}^{(n \times n)}$. Die Matrix \mathbf{A} heißt **stark regulär** wenn $r = n$ und $\operatorname{Re}\{s_i\} \neq \operatorname{Re}\{s_j\}$ für $i \neq j$ und $s_i \neq \bar{s}_j$. Die Matrix \mathbf{A} erfüllt die **Irrationalitätsbedingung** wenn $\operatorname{Im}\{s_i\}/\operatorname{Im}\{s_j\} \notin \mathbb{Q}$ für $i \neq j, s_i \neq \bar{s}_j$ und $s_i, s_j \notin \mathbb{R}$. Zusammengefaßt bedeutet das, daß die Systemmatrix keine mehrfachen Eigenwerte besitzen darf, die Realteile der konjugiert komplexen Eigenwert-Paare verschieden sein müssen und die Verhältnisse der Imaginärteile der konjugiert komplexen Eigenwert-Paare irrational sein müssen.

2.6.4 Derzeitiger Wissensstand

Der derzeitige Stand der Forschung kann wie folgt zusammengefaßt werden:

Ein System ist genau dann perspektivisch beobachtbar, wenn die Systemmatrix \mathbf{A} stark regulär ist, die Irrationalitätsbedingung erfüllt und

$$\operatorname{rang} \begin{bmatrix} (\lambda_1 \mathbf{I} - \mathbf{A})(\lambda_2 \mathbf{I} - \mathbf{A}) \\ \mathbf{C} \end{bmatrix} = n \quad \forall \lambda_1, \lambda_2 \in \mathbb{R} \text{ oder konj. komplex} \quad (2.47)$$

gilt.

Im weiteren Verlauf der Arbeit wird die reellwertige Matrix

$$\mathbf{H}_{py} = \mathbf{H}_{py}(\lambda_1, \lambda_2) = \begin{bmatrix} (\lambda_1 \mathbf{I} - \mathbf{A})(\lambda_2 \mathbf{I} - \mathbf{A}) \\ \mathbf{C} \end{bmatrix} \quad (2.48)$$

verwendet. Die gewählte Bezeichnung \mathbf{H}_{py} leitet sich von der mit \mathbf{H}_y bezeichneten Matrix des Popov-Belevitch-Hautus-Kriteriums ab. Die Abhängigkeit der Matrix \mathbf{H}_{py} von λ_1 und λ_2 wird aus Gründen der Übersichtlichkeit nicht immer explizit angeschrieben.

2.6.5 Alternative Formulierung

Die Formulierung mit den beiden Parametern λ_1 und λ_2 wird in [4], [5] und [6] bevorzugt und verdeutlicht den Zusammenhang mit dem Popov-Belevitch-Hautus-Rangkriterium. In [7] wird eine äquivalente Formulierung der Bedingung (2.47) bevorzugt:

$$\operatorname{rang} \mathbf{H}_{py}(\alpha, \beta) = \operatorname{rang} \begin{bmatrix} \mathbf{A}^2 + \alpha \mathbf{A} + \beta \mathbf{I} \\ \mathbf{C} \end{bmatrix} = n \quad \forall \alpha, \beta \in \mathbb{R} \quad (2.49)$$

Durch Ausmultiplizieren des Produktes in (2.47) sieht man, daß die beiden Rangbedingungen äquivalent sind:

$$(\lambda_1 \mathbf{I} - \mathbf{A})(\lambda_2 \mathbf{I} - \mathbf{A}) = \mathbf{A}^2 + \underbrace{(-\lambda_1 - \lambda_2)}_{\alpha} \mathbf{A} + \underbrace{\lambda_1 \lambda_2}_{\beta} \mathbf{I} \quad (2.50)$$

Für reelle und konjugiert komplexe λ_i sind die Ausdrücke $(-\lambda_1 - \lambda_2)$ und $\lambda_1 \lambda_2$ immer reell.

2.6.6 Invarianz gegenüber regulären Zustandstransformationen

Wie bereits erwähnt, gibt es Systemeigenschaften, die gegenüber regulären Zustandstransformationen invariant sind (zum Beispiel Steuerbarkeit, Beobachtbarkeit). Es wird nun gezeigt, daß auch die Eigenschaft der perspektivischen Beobachtbarkeit invariant gegenüber regulären Zustandstransformationen ist.

Beweis. Erfüllt \mathbf{A} die Voraussetzungen stark regulär und die Irrationalitätsbedingung, so erfüllt auch die transformierte Systemmatrix $\mathbf{T}^{-1} \mathbf{A} \mathbf{T}$ die Voraussetzungen, da die Eigenwerte ebenfalls invariant gegenüber regulären Transformationen sind. Das Kriterium zur Überprüfung perspektivischer Beobachtbarkeit für das ursprüngliche System (2.3) und (2.4) lautet

$$\text{rang } \mathbf{H}_{py} = \text{rang} \begin{bmatrix} \mathbf{A}^2 + \alpha \mathbf{A} + \beta \mathbf{I} \\ \mathbf{C} \end{bmatrix} = n \quad (2.51)$$

Für das transformierte System (2.8) und (2.9) ist der Rang folgender Matrix auszuwerten:

$$\mathbf{H}_{py,z} = \begin{bmatrix} (\mathbf{T}^{-1} \mathbf{A} \mathbf{T})^2 + \alpha \mathbf{T}^{-1} \mathbf{A} \mathbf{T} + \beta \mathbf{I} \\ \mathbf{C} \mathbf{T} \end{bmatrix} \quad (2.52)$$

$$\text{mit } (\mathbf{T}^{-1} \mathbf{A} \mathbf{T})^2 = \mathbf{T}^{-1} \mathbf{A} \mathbf{T} \mathbf{T}^{-1} \mathbf{A} \mathbf{T} = \mathbf{T}^{-1} \mathbf{A}^2 \mathbf{T}$$

$$\text{und } \mathbf{T}^{-1} \mathbf{I} \mathbf{T} = \mathbf{T}^{-1} \mathbf{T} = \mathbf{I}$$

$$= \begin{bmatrix} \mathbf{T}^{-1} (\mathbf{A}^2 + \alpha \mathbf{A} + \beta \mathbf{I}) \mathbf{T} \\ \mathbf{C} \mathbf{T} \end{bmatrix} \quad (2.53)$$

Der Rang von \mathbf{T} ist gleich n , \mathbf{T}^{-1} besitzt ebenfalls vollen Rang. Deswegen gilt

$$\text{rang } \mathbf{H}_{py} = \text{rang } \mathbf{H}_{py,z} \quad \square$$

und daraus folgt, daß die perspektivische Beobachtbarkeit invariant gegenüber regulären Zustandstransformationen ist.

2.7 Steuerbarkeitsmaße

Die Steuerbarkeit ist eine binäre Eigenschaft. Entweder ein System ist steuerbar, oder nicht. Diese Einteilung ist bei verschiedenen Fragestellungen unzureichend. Deswegen wünscht man sich Maßzahlen, die quantitativ angeben wie gut steuerbar ein System ist. Es gibt verschiedene Vorschläge für Steuerbarkeitsmaße. Eine vielversprechende Definition wird in [8] vorgestellt:

$$\mu(\mathbf{A}, \mathbf{B}) = \min_{\delta\mathbf{A}, \delta\mathbf{B}} \{ \|\delta\mathbf{A}, \delta\mathbf{B}\|_2 : (\mathbf{A} + \delta\mathbf{A}, \mathbf{B} + \delta\mathbf{B}) \text{ nicht steuerbar} \} \quad (2.54)$$

Das Maß repräsentiert den euklidischen Abstand von dem System (\mathbf{A}, \mathbf{B}) zum nächsten nicht steuerbaren System. In [9] wird gezeigt, daß das Maß (2.54) als

$$\mu(\mathbf{A}, \mathbf{B}) = \min_{\lambda \in \mathbb{C}} \sigma_n([\lambda \mathbf{I} - \mathbf{A}, \mathbf{B}]) \quad (2.55)$$

dargestellt werden kann.

Ein anderer Vorschlag für eine Maßdefinition stammt aus [10]. Dort wird das Maß

$$\Delta(\mathbf{A}, \mathbf{B}) = \min_{\delta\mathbf{A}, \delta\mathbf{B}} \{ d_A \|\delta\mathbf{A}\|_2 + d_B \|\delta\mathbf{B}\|_2 : (\mathbf{A} + \delta\mathbf{A}, \mathbf{B} + \delta\mathbf{B}) \text{ nicht steuerbar} \} \quad (2.56)$$

vorgeschlagen, das als

$$\Delta^2(\mathbf{A}, \mathbf{B}) = \min_{x \in \mathbb{C}^n, \|x\|=1} d_A x^H \mathbf{A} (\mathbf{I} - x x^H) \mathbf{A}^T x + d_B x^H \mathbf{B} \mathbf{B}^T x \quad (2.57)$$

berechnet werden kann, wobei x^H den konjugiert transponierten Vektor x symbolisiert.

2.8 Beobachtbarkeitsmaße

Für die Beobachtbarkeit gelten die gleichen Überlegungen und Wünsche wie für die Steuerbarkeit. Aufgrund der Dualität der beiden Eigenschaften kann man aus den Steuerbarkeitsmaßen Beobachtbarkeitsmaße ableiten. Für das System (\mathbf{A}, \mathbf{C}) erhält man ein Beobachtbarkeitsmaß, wenn man das Steuerbarkeitsmaß für das System $(-\mathbf{A}^T, \mathbf{C}^T)$ ausrechnet.

Kapitel 3

Überprüfung perspektivischer Beobachtbarkeit

In Abschnitt 2.6.4 wurde beschrieben, unter welchen Bedingungen ein System perspektivisch beobachtbar ist. Die Überprüfung der Bedingung (2.47) sieht auf den ersten Blick schwierig aus, da die Matrix \mathbf{H}_{py} für alle λ_1 und λ_2 vollen Rang besitzen muß. Bei näherer Betrachtung zeigt sich jedoch, daß man das Kriterium nicht für alle λ_1 und λ_2 überprüfen muß. Man kann eine Liste von Kandidaten, maximal $\mathcal{O}(n^2)$ viele, angeben, deren Überprüfung genügt. Besitzt die Matrix \mathbf{H}_{py} bei allen Kandidaten vollen Rang, so gilt dies auch für alle reelle oder konjugiert komplexe λ_1 und λ_2 . Die vollständige Kandidatenliste wird nun in diesem Kapitel Schritt für Schritt erarbeitet.

3.1 Rangermittlung

Seien $\mathbf{A} \in \mathbb{R}^{(n \times n)}$ und $\sigma(\mathbf{A})$ die Menge der Eigenwerte s_1, \dots, s_n von \mathbf{A} so gilt

$$\begin{aligned} \text{rang } \mathbf{A}_1(\lambda) &= n \quad \forall \lambda \in \mathbb{C} \setminus \sigma(\mathbf{A}) \\ \text{mit } \mathbf{A}_1(\lambda) &= \lambda \mathbf{I} - \mathbf{A} \end{aligned} \quad (3.1)$$

Beweis. Es gilt

$$\text{rang}(\lambda \mathbf{I} - \mathbf{A}) < n \iff \det(\lambda \mathbf{I} - \mathbf{A}) = 0 \quad (3.2)$$

Die Determinante von $\lambda \mathbf{I} - \mathbf{A}$ ist das charakteristische Polynom der Matrix \mathbf{A} und kann als

$$\det(\lambda \mathbf{I} - \mathbf{A}) = \prod_{i=1}^n (\lambda - s_i) \quad (3.3)$$

dargestellt werden. Es ist ersichtlich, daß nur bei $\lambda = s_i$ der Rang kleiner n werden kann und obige Aussage gilt. \square

Man sieht auch, daß die Aussage

$$\text{rang}(\lambda \mathbf{I} - \mathbf{A}) < n \iff \lambda \in \sigma(\mathbf{A}) \quad (3.4)$$

äquivalent zu Aussage (3.1) ist.

Seien $\mathbf{A} \in \mathbb{R}^{(n \times n)}$ und s_1, \dots, s_n die Eigenwerte von \mathbf{A} , dann gilt

$$\begin{aligned} \text{rang } \mathbf{A}_{12}(s_i, s_j) &\leq n - 2 \quad \forall i, j = 1, \dots, n; \quad i \neq j \\ \text{mit } \mathbf{A}_{12}(\lambda_1, \lambda_2) &= (\lambda_1 \mathbf{I} - \mathbf{A})(\lambda_2 \mathbf{I} - \mathbf{A}) \end{aligned} \quad (3.5)$$

Die Matrix \mathbf{A} läßt sich immer als \mathbf{WJW}^{-1} darstellen, wobei \mathbf{J} Jordan-Blockform besitzt. Die obige Aussage läßt sich mit Hilfe dieser Transformation zeigen, allerdings ist der Beweis umständlich anzuschreiben.

Da für das angegebene Kriterium für perspektivische Beobachtbarkeit \mathbf{A} stark regulär vorausgesetzt wird, und alle stark regulären Matrizen diagonalisierbar sind, wird die Aussage (3.5) nur für diagonalisierbare Matrizen gezeigt:

Beweis. Die Matrix \mathbf{A} ist diagonalisierbar, deswegen kann \mathbf{A} als $\mathbf{P}\mathbf{\Lambda}\mathbf{P}^{-1}$ dargestellt werden. Die Spalten der regulären Matrix \mathbf{P} sind Rechts-Eigenvektoren von \mathbf{A} ; $\mathbf{\Lambda}$ ist eine Diagonalmatrix mit den Eigenwerten von \mathbf{A} (siehe auch Abschnitt 2.2.1). Daher gilt

$$\mathbf{A}_1(s_i) = \mathbf{P} \begin{bmatrix} s_i - s_1 & & & & 0 \\ & \ddots & & & \\ & & s_i - s_i = 0 & & \\ & 0 & & \ddots & \\ & & & & s_i - s_n \end{bmatrix} \mathbf{P}^{-1} \quad (3.6)$$

Der Rang der Matrix $\mathbf{A}_1(s_i)$ ist gleich dem Rang von $\mathbf{\Lambda}$, und somit kleiner gleich $n - 1$.

Betrachtet man nun den Rang der Matrix $\mathbf{A}_{12}(s_i, s_j)$, so kann dieser als

$$\mathbf{P} \begin{bmatrix} s_i - s_1 & & & 0 \\ & \ddots & & \\ & & s_i - s_i & \\ 0 & & & \ddots \\ & & & & s_i - s_n \end{bmatrix} \mathbf{P}^{-1} \mathbf{P} \begin{bmatrix} s_j - s_1 & & & 0 \\ & \ddots & & \\ & & s_j - s_j & \\ 0 & & & \ddots \\ & & & & s_j - s_n \end{bmatrix} \mathbf{P}^{-1} = \quad (3.7)$$

$$\mathbf{P} \begin{bmatrix} (s_i - s_1)(s_j - s_1) & & & & \\ & \ddots & & & \\ & & 0(s_j - s_i) & & \\ & & & \ddots & \\ & & & & (s_i - s_j)0 \\ 0 & & & & & \ddots \\ & & & & & & (s_i - s_n)(s_j - s_n) \end{bmatrix} \mathbf{P}^{-1} \quad (3.8)$$

dargestellt werden. Es ergeben sich hier mindestens 2 Nulleinträge in der Diagonalmatrix, aus diesem Grund ist der Rang kleiner gleich $n - 2$. \square

Nun wird untersucht, wie sich der Rang der Matrix

$$\text{rang } \mathbf{A}_{12}(\lambda_1, \lambda_2) \quad \forall \lambda_1, \lambda_2 \in \mathbb{R} \text{ oder konjugiert komplex} \quad (3.9)$$

in Abhängigkeit von zwei Parametern λ_1, λ_2 verhält.

Für eine kompakte Schreibweise wird folgende Notation festgelegt: Mit s_1, \dots, s_n werden die Eigenwerte der Matrix \mathbf{A} symbolisiert, $\sigma_{\mathbb{R}}(\mathbf{A})$ ist die Menge aller reellen Eigenwerte und $\sigma_{\mathbb{C}}(\mathbf{A})$ ist die Menge aller komplexen Eigenwerte.

Es gibt fünf verschiedene Fälle, aus welchen Mengen man die beiden Parameter λ_1 und λ_2 wählen kann:

- i) beide Parameter sind reell aber keine Eigenwerte
($\lambda_1, \lambda_2 \in \mathbb{R} \setminus \sigma_{\mathbb{R}}(\mathbf{A})$)
- ii) Parameter sind konjugiert komplex aber keine Eigenwerte
($\lambda_1 \in \mathbb{C} \setminus \sigma_{\mathbb{C}}(\mathbf{A}), \lambda_2 = \overline{\lambda_1}$)
- iii) beide Parameter sind reell und Eigenwerte
($\lambda_1, \lambda_2 \in \sigma_{\mathbb{R}}(\mathbf{A})$)
- iv) Parameter sind konjugiert komplex und Eigenwerte
($\lambda_1, \lambda_2 \in \sigma_{\mathbb{C}}(\mathbf{A}), \lambda_2 = \overline{\lambda_1}$)
- v) beide Parameter sind reell, einer ist allerdings ein Eigenwert
($\lambda_1 \in \sigma_{\mathbb{R}}(\mathbf{A}), \lambda_2 \in \mathbb{R} \setminus \sigma_{\mathbb{R}}(\mathbf{A})$)

Alle anderen Kombinationen sind aufgrund der Tatsache, daß λ_1 und λ_2 reell oder konjugiert komplex sein müssen, ausgeschlossen.

In den Fällen i) und ii) ist der Rang gleich n . Es wurde bereits gezeigt, daß $\mathbf{A}_1(\lambda_1)$ vollen Rang besitzt. Das Produkt von $\mathbf{A}_1(\lambda_1)\mathbf{A}_1(\lambda_2)$ besitzt dann ebenfalls vollen Rang.

In den Fällen iii) und iv) ist der Rang laut Aussage (3.5) kleiner gleich $n - 2$.

Bleibt noch Fall v) zu betrachten: Der Rang der Matrix $\mathbf{A}_1(\lambda_1)$ ist kleiner gleich $n - 1$, der Rang der Matrix $\mathbf{A}_1(\lambda_2)$ ist gleich n , also ist der Rang des Produktes kleiner gleich $n - 1$, unabhängig der Wahl von λ_2 ! Der Rangverlust wird hier nur mit einem Freiheitsgrad (λ_1) erreicht, der zweite Freiheitsgrad (λ_2) wird nicht verwendet. Deswegen kann der zweite Parameter beliebig gewählt werden, der Einfachheit halber $\lambda_2 = 0$.

Es zeigt sich, daß nur in den Fällen iii), iv) und v) der Rang kleiner n werden kann. Man kann nun eine Menge \mathbb{S} an Kandidaten definieren, deren Überprüfung ausreicht, um den vollen Rang der Matrix $\mathbf{A}_{12}(\lambda_1, \lambda_2)$ zu prüfen:

$$\begin{aligned} \mathbb{S} &= \{(\lambda_1, \lambda_2)\} \\ &= \{(s_1, s_2) \mid s_1, s_2 \in \sigma_{\mathbb{R}}(\mathbf{A})\} \cup && \text{Fall iii)} \\ &\quad \{(s_1, s_2) \mid s_1, s_2 \in \sigma_{\mathbb{C}}(\mathbf{A}), s_1 = \overline{s_2}\} \cup && \text{Fall iv)} \\ &\quad \{(s_1, 0) \mid s_1 \in \sigma_{\mathbb{R}}(\mathbf{A})\} && \text{Fall v)} \end{aligned} \quad (3.10)$$

Oder alternativ, mit Hilfe der Beziehung (2.50), in den Parametern α und β :

$$\begin{aligned} \mathbb{S} &= \{(\alpha, \beta)\} \\ &= \{(-s_1 - s_2, s_1 s_2) \mid s_1, s_2 \in \sigma_{\mathbb{R}}(\mathbf{A})\} \cup \\ &\quad \{(-s_1 - s_2, s_1 s_2) \mid s_1, s_2 \in \sigma_{\mathbb{C}}(\mathbf{A}), s_1 = \overline{s_2}\} \cup \\ &\quad \{(-s_1, 0) \mid s_1 \in \sigma_{\mathbb{R}}(\mathbf{A})\} \end{aligned} \quad (3.11)$$

3.2 Kandidaten für die Überprüfung

Um nun das Kriterium (2.47) bzw. (2.49) zu überprüfen, reicht es, wenn man in Abschnitt 3.1 erarbeitete Kandidatenliste überprüft. Sie deckt alle möglichen Fälle ab, wie der obere Teil $\mathbf{A}_{12}(\lambda_1, \lambda_2)$ den Rang verlieren kann. Nur dann kann auch die gesamte Matrix den Rang verlieren.

3.3 Alternatives Kriterium

Das Rangkriterium nach Popov-Belevitch-Hautus kann auch alternativ als Eigenvektortest formuliert werden (siehe Abschnitt 2.4). Da das Kriterium zur Überprüfung perspektivischer Beobachtbarkeit eine Verallgemeinerung des Rangkriteriums

nach Popov-Belevitch-Hautus ist, ist auch hier eine alternative Formulierung möglich.

Folgende Aussagen sind äquivalent:

- ① $\text{rang } \mathbf{H}_{py} = \text{rang} \begin{bmatrix} (\lambda_1 \mathbf{I} - \mathbf{A})(\lambda_2 \mathbf{I} - \mathbf{A}) \\ \mathbf{C} \end{bmatrix} = n \quad \forall \lambda_1, \lambda_2 \in \mathbb{S}$
- ② $\mathbf{C}p_i \neq 0 \quad \forall i, p_i \in \mathbb{R}^n$
 und
 $\mathbf{C}p_i, \mathbf{C}p_j$ linear unabhängig $\forall i, j; i \neq j; p_i + p_j \in \mathbb{R}^n$

wobei p_1, \dots, p_n die Rechtseigenvektoren

$$\mathbf{A}p_i = s_i p_i \quad (3.12)$$

der Systemmatrix sind.

3.3.1 Kommutivität eines Matrizenproduktes

Um die obige Äquivalenz zu zeigen, ist noch ein Zwischenresultat notwendig, das zuerst bewiesen wird:

Das Produkt

$$(\lambda_1 \mathbf{I} - \mathbf{A})(\lambda_2 \mathbf{I} - \mathbf{A}) \quad (3.13)$$

ist kommutativ für $\lambda_1, \lambda_2 \in \mathbb{C}$.

Beweis. Durch Ausmultiplizieren und wieder herausheben sieht man die Kommutivität:

$$\begin{aligned} (\lambda_1 \mathbf{I} - \mathbf{A})(\lambda_2 \mathbf{I} - \mathbf{A}) &= \mathbf{A}^2 - \lambda_1 \mathbf{A} - \lambda_2 \mathbf{A} + \lambda_1 \lambda_2 \mathbf{I} \\ &= (\lambda_2 \mathbf{I} - \mathbf{A})(\lambda_1 \mathbf{I} - \mathbf{A}) \quad \square \end{aligned} \quad (3.14)$$

Jetzt kann die Äquivalenz zwischen ① und ② gezeigt werden.

Beweis. Zuerst wird ① \Leftrightarrow ② gezeigt.

Dazu nimmt man an, daß

$$\text{rang } \mathbf{H}_{py} < n \quad (3.15)$$

sei. Deswegen

$$\exists p \in \mathbb{R}^n : \mathbf{H}_{py} p = 0 \quad (3.16)$$

Diese Gleichung kann aufgrund der Form von \mathbf{H}_{py} in zwei Gleichungen aufgeteilt werden:

$$(\lambda_1 \mathbf{I} - \mathbf{A})(\lambda_2 \mathbf{I} - \mathbf{A})p = 0 \quad (3.17)$$

$$\mathbf{C}p = 0 \quad (3.18)$$

Nun gibt es zwei Möglichkeiten, wie der Vektor p aussehen kann, damit er die Gleichungen (3.17) und (3.18) erfüllt.

Fall 1: p ist ein reeller Eigenvektor von \mathbf{A} , es gilt

$$(s\mathbf{I} - \mathbf{A})p = 0 \quad (3.19)$$

wobei s der zu p gehörige Eigenwert ist. Jetzt kann man $\lambda_1 = s$ wählen und sieht so, daß die Gleichung (3.17) (für alle λ_2) erfüllt ist:

$$(s\mathbf{I} - \mathbf{A})(\lambda_2 \mathbf{I} - \mathbf{A})p = 0 \quad (3.20)$$

$$(\lambda_2 \mathbf{I} - \mathbf{A}) \underbrace{(s\mathbf{I} - \mathbf{A})p}_0 = 0 \quad (3.21)$$

Also erfüllt jeder reelle Eigenvektor, der $\mathbf{C}p = 0$ erfüllt, die beiden Gleichungen (3.17) und (3.18).

Fall 2: p ist eine Linear-Kombination aus zwei Eigenvektoren:

$$p = \alpha p_i + \beta p_j \quad (3.22)$$

Man wählt nun $\lambda_1 = s_i$ und $\lambda_2 = s_j$, wobei s_i und s_j die zu den Eigenvektoren gehörigen Eigenwerte sind. Dadurch wird Gleichung (3.17) erfüllt:

$$(s_i \mathbf{I} - \mathbf{A})(s_j \mathbf{I} - \mathbf{A})(\alpha p_i + \beta p_j) = 0 \quad (3.23)$$

$$\alpha \underbrace{(s_j \mathbf{I} - \mathbf{A})(s_i \mathbf{I} - \mathbf{A})p_i}_0 + \beta \underbrace{(s_j \mathbf{I} - \mathbf{A})(s_j \mathbf{I} - \mathbf{A})p_j}_0 = 0 \quad (3.24)$$

Da $p \in \mathbb{R}^n$ muß natürlich die Summe der beiden Eigenvektoren auch reell sein. Daraus folgt, daß die dazugehörigen Eigenwerte auch reell bzw. konjugiert komplex sein müssen. Für Gleichung (3.18) gilt dann

$$\mathbf{C}p = 0 \quad (3.25)$$

$$\mathbf{C}(\alpha p_i + \beta p_j) = 0 \quad (3.26)$$

$$\alpha \mathbf{C}p_i + \beta \mathbf{C}p_j = 0 \quad (3.27)$$

Um die Gleichung (3.27) zu erfüllen müssen die Vektoren $\mathbf{C}p_i$ und $\mathbf{C}p_j$ linear abhängig sein.

Also erfüllen zwei Eigenvektoren von \mathbf{A} die beiden Gleichungen (3.17) und (3.18)

wenn die Vektoren $\mathbf{C}p_i$ und $\mathbf{C}p_j$ linear abhängig sind und die Summe der Eigenvektoren reell ist.

Jetzt bleibt noch die Gegenrichtung ① \Rightarrow ② offen:

Hier gibt es wieder zwei mögliche Fälle.

Fall 1: Man nimmt man es

$$\exists p \in \mathbb{R}^n : (s\mathbf{I} - \mathbf{A})p = 0 \text{ und} \quad (3.28)$$

$$\mathbf{C}p = 0 \quad (3.29)$$

Erfüllt ein Vektor die Gleichung (3.28), so erfüllt er auch

$$(s\mathbf{I} - \mathbf{A})p = 0 \quad | \cdot (\lambda_2\mathbf{I} - \mathbf{A}) \text{ von links} \quad (3.30)$$

$$(\lambda_2\mathbf{I} - \mathbf{A})(s\mathbf{I} - \mathbf{A})p = 0 \quad (3.31)$$

Schreibt man nun die beiden Gleichungen (3.31) und (3.29) als eine an, so erhält man

$$\begin{bmatrix} (s\mathbf{I} - \mathbf{A})(\lambda_2\mathbf{I} - \mathbf{A}) \\ \mathbf{C} \end{bmatrix} p = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (3.32)$$

oder anders formuliert (mit $s = \lambda_1$)

$$\text{rang} \begin{bmatrix} (\lambda_1\mathbf{I} - \mathbf{A})(\lambda_2\mathbf{I} - \mathbf{A}) \\ \mathbf{C} \end{bmatrix} < n \quad (3.33)$$

Fall 2: Man nimmt an es

$$\begin{aligned} \exists p_i, p_j : (s_k\mathbf{I} - \mathbf{A})p_k \text{ für } k = i, j, p_i + p_j \in \mathbb{R}^n \text{ und} \\ \mathbf{C}p_i, \mathbf{C}p_j \text{ linear abhängig} \end{aligned} \quad (3.34)$$

Formt man nun die beiden Eigenvektorgleichungen um und definiert $p \equiv \alpha p_i + \beta p_j$ so erhält man

$$(s_i\mathbf{I} - \mathbf{A})p_i = 0 \quad | \cdot \alpha \quad (s_j\mathbf{I} - \mathbf{A})p_j = 0 \quad | \cdot \beta \quad (3.35)$$

$$\alpha(s_i\mathbf{I} - \mathbf{A})p_i + \beta(s_j\mathbf{I} - \mathbf{A})p_j = 0 + 0 \quad (3.36)$$

$$(s_i\mathbf{I} - \mathbf{A})(s_j\mathbf{I} - \mathbf{A})(\alpha p_i + \beta p_j) = 0 \quad (3.37)$$

$$(s_i\mathbf{I} - \mathbf{A})(s_j\mathbf{I} - \mathbf{A})p = 0 \quad (3.38)$$

Bedingung (3.34) kann als

$$\alpha\mathbf{C}p_i + \beta\mathbf{C}p_j = 0 \quad (3.39)$$

$$\mathbf{C}(\alpha p_i + \beta p_j) = 0 \quad (3.40)$$

$$\mathbf{C}p = 0 \quad (3.41)$$

formuliert werden. Fügt man nun die beiden Gleichungen (3.38) und (3.41) zusammen, ergibt sich

$$\begin{bmatrix} (s_i \mathbf{I} - \mathbf{A})(s_j \mathbf{I} - \mathbf{A}) \\ \mathbf{C} \end{bmatrix} p = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (3.42)$$

oder anders formuliert (mit $\lambda_1 = s_i$ und $\lambda_2 = s_j$)

$$\text{rang} \begin{bmatrix} (\lambda_1 \mathbf{I} - \mathbf{A})(\lambda_2 \mathbf{I} - \mathbf{A}) \\ \mathbf{C} \end{bmatrix} < n \quad \square$$

3.4 Ergebnis

Zusammenfassend kann das Ergebnis aus Abschnitt 2.6.4 verbessert werden, indem die Menge der möglichen Parameterwerte eingeschränkt wird. Zusätzlich kann das Kriterium auf eine alternative Weise formuliert werden.

Folgende Aussagen sind äquivalent:

1. Ein System der Form (2.3) und (2.4) ist perspektivisch beobachtbar
2. Die Systemmatrix \mathbf{A} ist stark regulär, erfüllt die Irrationalitätsbedingung und

$$\text{rang} \begin{bmatrix} (\lambda_1 \mathbf{I} - \mathbf{A})(\lambda_2 \mathbf{I} - \mathbf{A}) \\ \mathbf{C} \end{bmatrix} = n \quad \forall (\lambda_1, \lambda_2) \in \mathbb{S} \quad (3.43)$$

$$\begin{aligned} \text{mit } \mathbb{S} = & \{(s_1, s_2) \mid s_1, s_2 \in \sigma_{\mathbb{R}}(\mathbf{A})\} \cup \\ & \{(s_1, s_2) \mid s_1, s_2 \in \sigma_{\mathbb{C}}(\mathbf{A}), s_1 = \overline{s_2}\} \cup \\ & \{(s_1, 0) \mid s_1 \in \sigma_{\mathbb{R}}(\mathbf{A})\} \end{aligned} \quad (3.44)$$

gilt.

3. Die Systemmatrix \mathbf{A} ist stark regulär, erfüllt die Irrationalitätsbedingung und

$$\begin{aligned} \mathbf{C}p_i \neq 0 \quad \forall i, p_i \in \mathbb{R}^n \\ \text{und} \\ \mathbf{C}p_i, \mathbf{C}p_j \text{ linear unabhängig} \quad \forall i, j; i \neq j; p_i + p_j \in \mathbb{R}^n \end{aligned} \quad (3.45)$$

ist erfüllt.

Kapitel 4

Hinreichende Bedingungen perspektivischer Beobachtbarkeit

In diesem Kapitel werden verschiedene hinreichende Bedingungen vorgestellt, mit denen unter bestimmten Voraussetzungen relativ einfach die Eigenschaft der perspektivischen Beobachtbarkeit überprüft werden kann. Es wird immer davon ausgegangen, daß die Systemmatrix \mathbf{A} stark regulär ist und die Irrationalitätsbedingung erfüllt.

4.1 Systeme mit skalarem Ausgang

Systeme der Form (2.3), (2.4) besitzen einen skalaren Ausgang, wenn die Ausgangsmatrix \mathbf{C} eine $(1 \times n)$ -Matrix ist. Solche Systeme sind nicht perspektivisch beobachtbar.

Beweis. Da die Systemmatrix \mathbf{A} die beiden Voraussetzungen erfüllt, kann das System in Diagonalform (2.12), (2.13) transformiert werden. Für die Matrix $\mathbf{H}_{py}(\lambda_1, \lambda_2)$ ergibt sich folgende Struktur:

$$\mathbf{H}_{py} = \begin{bmatrix} (\lambda_1 - s_1)(\lambda_2 - s_1) & & & 0 \\ & (\lambda_1 - s_2)(\lambda_2 - s_2) & & \\ & & \ddots & \\ \bar{\delta}_1 & 0 & & (\lambda_1 - s_n)(\lambda_2 - s_n) \\ & \bar{\delta}_2 & \dots & \bar{\delta}_n \end{bmatrix} \quad (4.1)$$

wobei $\bar{\delta}_1, \dots, \bar{\delta}_n$ die Einträge der transformierten Ausgangsmatrix sind. Die Eigenwerte s_1, \dots, s_n sind entweder reell oder konjugiert komplexe Paare. Weil $n \geq 2$ können mit Hilfe der beiden Parameter λ_1 und λ_2 zwei Einträge in der Diagonale

4.4 Zusammenhänge mit der Beobachtbarkeit

4.4.1 Nicht beobachtbare Systeme

Systeme der Form (2.3), (2.4), die nicht beobachtbar sind, sind nicht perspektivisch beobachtbar.

Beweis. Für ein nichtbeobachtbares System ist das Rangkriterium nach Popov-Belevitch-Hautus (2.19) verletzt, das heißt, es

$$\exists \lambda \in \mathbb{C} : \text{rang} \begin{bmatrix} \lambda \mathbf{I} - \mathbf{A} \\ \mathbf{C} \end{bmatrix} < n \quad (4.6)$$

Daraus folgt, es

$$\exists p \in \mathbb{C}^n : \begin{bmatrix} \lambda \mathbf{I} - \mathbf{A} \\ \mathbf{C} \end{bmatrix} p = 0 \quad (4.7)$$

Gleichung (4.7) kann in folgende zwei Gleichungen aufgespalten werden:

$$(\lambda \mathbf{I} - \mathbf{A})p = 0 \quad | \cdot (\bar{\lambda} \mathbf{I} - \mathbf{A}) \text{ von links} \quad (4.8)$$

$$\mathbf{C}p = 0 \quad (4.9)$$

Erweitert man Gleichung (4.8) und führt sie wieder mit Gleichung (4.9) zusammen, so erhält man

$$\begin{bmatrix} (\bar{\lambda} \mathbf{I} - \mathbf{A})(\lambda \mathbf{I} - \mathbf{A}) \\ \mathbf{C} \end{bmatrix} p = 0 \quad (4.10)$$

oder anders formuliert (mit $\bar{\lambda} = \lambda_1$ und $\lambda = \lambda_2$)

$$\text{rang} \begin{bmatrix} (\lambda_1 \mathbf{I} - \mathbf{A})(\lambda_2 \mathbf{I} - \mathbf{A}) \\ \mathbf{C} \end{bmatrix} < n \quad \square$$

Ist $\lambda \in \mathbb{C} \setminus \mathbb{R}$, so sind λ_1 und λ_2 konjugiert komplex. Anderenfalls ist $\lambda \in \mathbb{R}$ und $\bar{\lambda} = \lambda$ und somit sind λ_1 und λ_2 gleich und reell.

4.4.2 Perspektivisch beobachtbare Systeme

Systeme der Form (2.3), (2.4), die perspektivisch beobachtbar sind, sind beobachtbar.

Beweis. Die Korrektheit folgt aus Abschnitt 4.4.1 und dem Kontrapositionsgesetz: Sind A und B logische Aussagen, so gilt

$$A \rightarrow B \iff \neg B \rightarrow \neg A \quad (4.11)$$

□

4.5 Systeme in Modalform

Sind Systeme in Form (2.12), (2.13) gegeben, so spricht man von der Modalform. Die Spalten der Ausgangsmatrix $\bar{\delta}$ werden als $\bar{\delta}_1, \dots, \bar{\delta}_n$ bezeichnet:

$$\bar{\delta} = \left[\begin{array}{c|c|c|c} \bar{\delta}_1 & \bar{\delta}_2 & \dots & \bar{\delta}_n \end{array} \right] \quad (4.12)$$

In diesem Abschnitt werden zwei strukturelle Bedingungen für Systeme in Modalform vorgestellt, die die Eigenschaft der perspektivischen Beobachtbarkeit zerstören. Liegt ein System in Modalform vor und die beiden Bedingungen sind nicht erfüllt, so ist das System perspektivisch beobachtbar.

4.5.1 Ausgangsmatrix mit einer Nullspalte

Systeme der Form (2.12), (2.13) sind nicht perspektivisch beobachtbar, wenn die Matrix $\bar{\delta}$ eine Nullspalte besitzt. Mathematisch kann das folgendermaßen formuliert werden:

$$\exists i : \bar{\delta}_i = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \implies (\Lambda, \bar{\delta}) \text{ nicht perspektivisch beobachtbar} \quad (4.13)$$

Beweis. Für Systeme der Form (2.12), (2.13) ergibt sich die Matrix \mathbf{H}_{py} als

$$\mathbf{H}_{py}(\lambda_1, \lambda_2) = \begin{bmatrix} (\lambda_1 - s_1)(\lambda_2 - s_1) & & & 0 \\ & (\lambda_1 - s_2)(\lambda_2 - s_2) & & \\ & & \ddots & \\ & 0 & & (\lambda_1 - s_n)(\lambda_2 - s_n) \\ \bar{\delta}_1 & \bar{\delta}_2 & \dots & \bar{\delta}_n \end{bmatrix} \quad (4.14)$$

Wählt man nun für $\lambda_1 = s_i$ und für $\lambda_2 = \bar{s}_i$ so erhält man

$$\mathbf{H}_{py} = \begin{bmatrix} (s_i - s_1)(\bar{s}_i - s_1) & & & 0 \\ & \ddots & & \\ & & (s_i - s_i)(\bar{s}_i - s_i) & \\ & 0 & & \ddots \\ \bar{\delta}_1 & \dots & \bar{\delta}_i & \dots & \bar{\delta}_n \end{bmatrix} \quad (4.15)$$

und man erkennt unter Berücksichtigung von $\bar{\delta}_i = 0$, daß die i -te Spalte aus Nullen besteht. Somit ist

$$\text{rang } \mathbf{H}_{py} < n \quad \square$$

4.5.2 Zwei linear abhängige Spalten in der Ausgangsmatrix

Systeme der Form (2.12), (2.13) sind nicht perspektivisch beobachtbar, wenn die Matrix $\bar{\delta}$ zwei linear abhängige Spalten besitzt und die Summe der „zugehörigen“ Eigenwerte von \mathbf{A} reell ist. Mathematisch kann das als

$$\begin{aligned} \exists i, j : \bar{\delta}_i, \bar{\delta}_j \text{ linear abhängig und } s_i + s_j \in \mathbb{R} \\ \implies (\mathbf{A}, \bar{\delta}) \text{ nicht perspektivisch beobachtbar} \end{aligned} \quad (4.16)$$

formuliert werden.

Beweis. Gleichung (4.14) zeigt die Gestalt der Matrix \mathbf{H}_{py} für Systeme in Modalform. Dieses Mal wird $\lambda_1 = s_i$ und $\lambda_2 = s_j$ gesetzt. Da die Summe der beiden Eigenwerte s_i, s_j reell ist, sind die beiden Parameter λ_1, λ_2 entweder reell oder konjugiert komplex, und diese Wahl ist zulässig. Man erhält somit die Matrix

$$\mathbf{H}_{py} = \begin{bmatrix} (s_i - s_1)(s_j - s_1) & & & & & & & 0 \\ & \ddots & & & & & & \\ & & (s_i - s_i)(s_j - s_i) & & & & & \\ & & & \ddots & & & & \\ & & & & (s_i - s_j)(s_j - s_j) & & & \\ & 0 & & & & \ddots & & \\ \bar{\delta}_1 & \cdots & \bar{\delta}_i & \cdots & \bar{\delta}_j & \cdots & \bar{\delta}_n \end{bmatrix}$$

Weil $\bar{\delta}_i$ und $\bar{\delta}_j$ zueinander parallel sind, sind auch die i -te und die j -te Spalte von \mathbf{H}_{py} zueinander parallel und es gilt

$$\text{rang } \mathbf{H}_{py} < n \quad \square$$

Kapitel 5

Numerisch stabile Überprüfung der perspektivischen Beobachtbarkeit

In Kapitel 3 wurde eine Methode vorgestellt, wie man die Rangbedingung (2.47) bzw. (2.49) direkt überprüfen kann. Diese Überprüfung setzt voraus, daß die Eigenwerte der Systemmatrix \mathbf{A} exakt bekannt sind. Deswegen werden in diesem Kapitel ein Algorithmus erarbeitet, der ohne exakte Kenntnis der Eigenwerte auskommt.

5.1 Numerische Probleme bei der Überprüfung der Rangbedingung

Falls die Eigenwerte der Systemmatrix nicht exakt bekannt sind, werden diese numerisch berechnet. Bei der Berechnung können Fehler auftreten. Die Rangbedingung wird somit an den „falschen“ Stellen ausgewertet und das Ergebnis ist nicht aussagekräftig. Folgendes Beispiel, inspiriert durch [11], zeigt diese Problematik:

$$\mathbf{\Lambda} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix} \quad (5.1)$$

$$\bar{\boldsymbol{\delta}} = \begin{bmatrix} 10 & 5 & 0 \\ 6 & 3 & 1 \end{bmatrix} \quad (5.2)$$

Man erkennt, daß $\bar{\boldsymbol{\delta}}$ zwei „parallele“ Spalten besitzt und die zugehörigen Eigenwerte von $\mathbf{\Lambda}$ reell sind. Somit ist das vorliegende System nach Abschnitt 4.5.2 nicht perspektivisch beobachtbar.

Durch eine reguläre Zustandstransformation mit

$$\mathbf{P} = \begin{bmatrix} 1 & -4 & 7 \\ -3 & 9 & -49 \\ 0 & 1 & 9 \end{bmatrix} \quad (5.3)$$

entsteht das System

$$\mathbf{A} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}^{-1} = \begin{bmatrix} -149 & -50 & -154 \\ 537 & 180 & 546 \\ -27 & -9 & -25 \end{bmatrix} \quad (5.4)$$

$$\mathbf{C} = \bar{\delta}\mathbf{P} = \begin{bmatrix} 1435 & 475 & 1470 \\ 858 & 284 & 879 \end{bmatrix} \quad (5.5)$$

Kennt man diese Zustandstransformation und wertet den Rang von $\mathbf{H}_{py}(\alpha, \beta)$ von \mathbf{A} und \mathbf{C} bei $\alpha = -3$ und $\beta = 2$ mit Hilfe von MATLAB aus, so beträgt der minimalen Singulärwert an dieser Stelle $3 \cdot 10^{-14}$ und der Rangverlust der Matrix wird erkannt.

Kennt man hingegen nur \mathbf{A} und \mathbf{C} und ermittelt mittels MATLAB die Eigenwerte von \mathbf{A} , so erhält man für den maßgeblichen Kandidaten das Wertepaar $(-3 - 5.02 \cdot 10^{-13}, 2 + 1.05 \cdot 10^{-11})$. Der minimale Singulärwert von \mathbf{H}_{py} an dieser Stelle beträgt $3.31 \cdot 10^{-12}$, laut MATLAB besitzt die Matrix vollen Rang.

Bereits die hier vorliegenden Abweichungen im Bereich von 10^{-11} reichen aus, damit die Überprüfung des Kriteriums fehlschlägt. Man wünscht sich daher eine numerisch stabile Methode, die nicht auf die exakte Kenntnis der Eigenwerte angewiesen ist.

5.2 Formulierung als Minimierungsproblem

Sind $\sigma_1, \dots, \sigma_n$ die geordneten Singulärwerte einer Matrix \mathbf{Q} , so gilt

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r > \sigma_{r+1} = \dots = \sigma_n = 0 \iff \text{rang } \mathbf{Q} = r. \quad (5.6)$$

Diese Beziehung wird zur numerischen Bestimmung des Ranges von \mathbf{Q} verwendet. Die Anzahl der Singulärwerte, die über einer vorgegebenen Toleranzschranke liegen, ist der numerische Rang der Matrix.

Will man nun wissen, ob der Rang von \mathbf{Q} kleiner als n ist, so reicht es, zu überprüfen ob der minimale Singulärwert $\sigma_n(\mathbf{Q})$ gleich null ist.

Man kann also Bedingung (2.49) in

$$f_{\mathbf{H}_{py}}(\alpha, \beta) = \sigma_n \left(\begin{bmatrix} \mathbf{A}^2 + \alpha\mathbf{A} + \beta\mathbf{I} \\ \mathbf{C} \end{bmatrix} \right) \neq 0 \quad \forall \alpha, \beta \in \mathbb{R} \quad (5.7)$$

umformulieren. Die Bedingung ist erfüllt, wenn für das globale Minimum

$$\min_{\alpha, \beta \in \mathbb{R}} f_{\mathbf{H}_{py}}(\alpha, \beta) > 0 \quad (5.8)$$

gilt.

Das Problem wurde nun von der Überprüfung des (diskreten) Ranges einer Matrix in ein (kontinuierliches) Minimierungsproblem überführt.

5.3 Eigenschaften des Minimierungsproblems

Die Funktion $f_{\mathbf{H}_{py}}(\alpha, \beta)$ ist stetig und kann mehrere lokale Minima besitzen. Die Abbildungen 5.1 bis 5.4 zeigen das Aussehen der Funktion für die Beispiele (5.9) bis (5.12).

Typischerweise werden solche Minimierungsprobleme mit Hilfe numerischer Optimierungsverfahren gelöst, die von einem Startpunkt ausgehend mit Hilfe der Steigung und der Krümmung der Funktion ein Minimum suchen.

Um diesen Prozeß anschaulicher darzustellen, interpretiert man den Verlauf der Funktion als „Landschaft“. Die Standard-Verfahren können auf dem Weg zum Minimum nur „bergab“, nicht aber „bergauf“ oder „geradeaus“ suchen. So finden diese Verfahren eventuell nur ein lokales Minimum und nicht das (gesuchte) globale Minimum.

Dieses Verhalten kann gut durch Abbildung 5.2 verdeutlicht werden. Die Landschaft besteht aus zwei „Mulden“, die durch einen „Berg“ voneinander getrennt sind. Ein Standard-Verfahren würde, je nach Startpunkt, nur das Minimum der jeweiligen Mulde finden, aber nicht in die andere Mulde wechseln können.

Eine weitere Schwierigkeit für diese Verfahren sind „ebene“ Gebiete. In Abbildung 5.3 erkennt man 3 „Täler“, deren Talböden (im Gegensatz zu natürlichen Tälern) keine Gefälle aufweisen. Die Verfahren würden zwar eines dieser Täler finden, aber die Suche wäre dort zu Ende. Sie können nicht mehr weiter bergab suchen und finden an jeder Stelle im Talboden ein lokales Minimum.

5.3.1 Beispiele

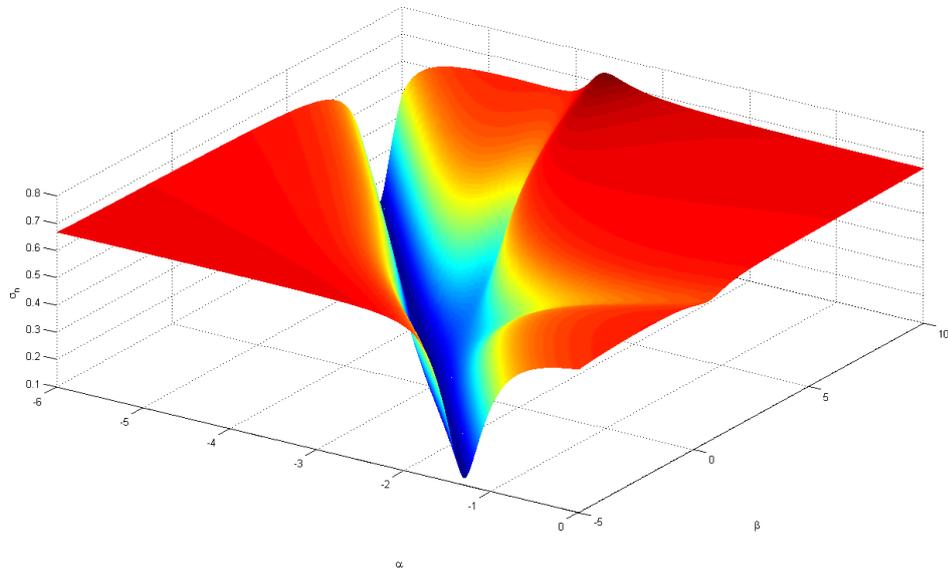


Abbildung 5.1: minimaler Singulärwert von \mathbf{H}_{py} über α, β , Beispiel (5.9)

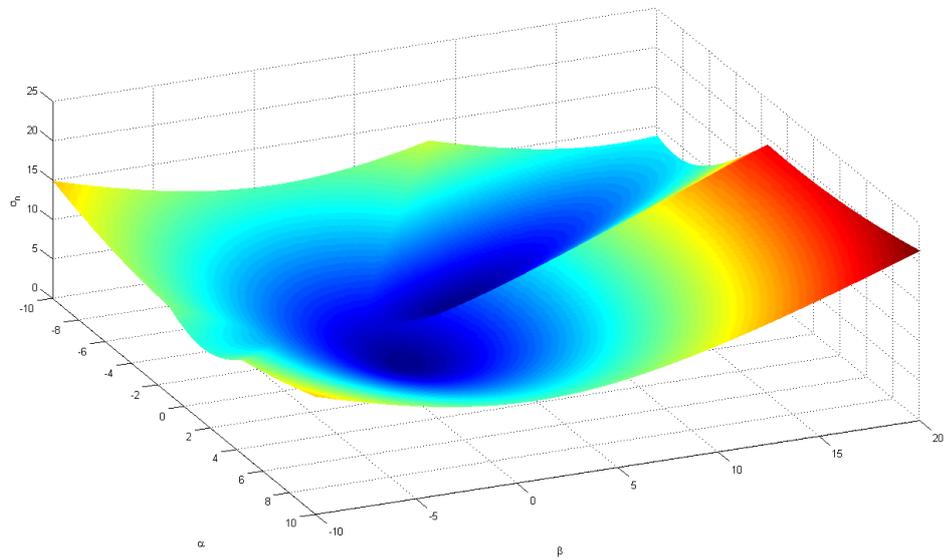
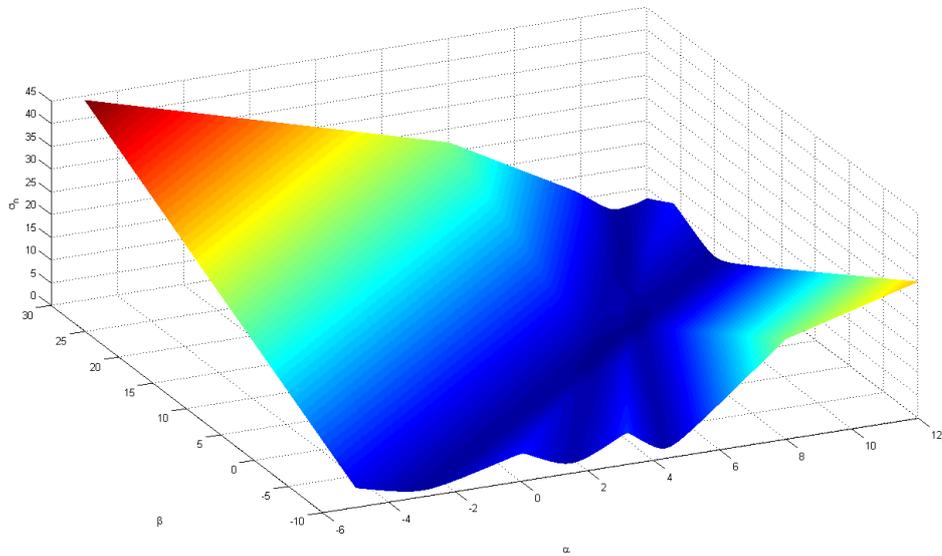
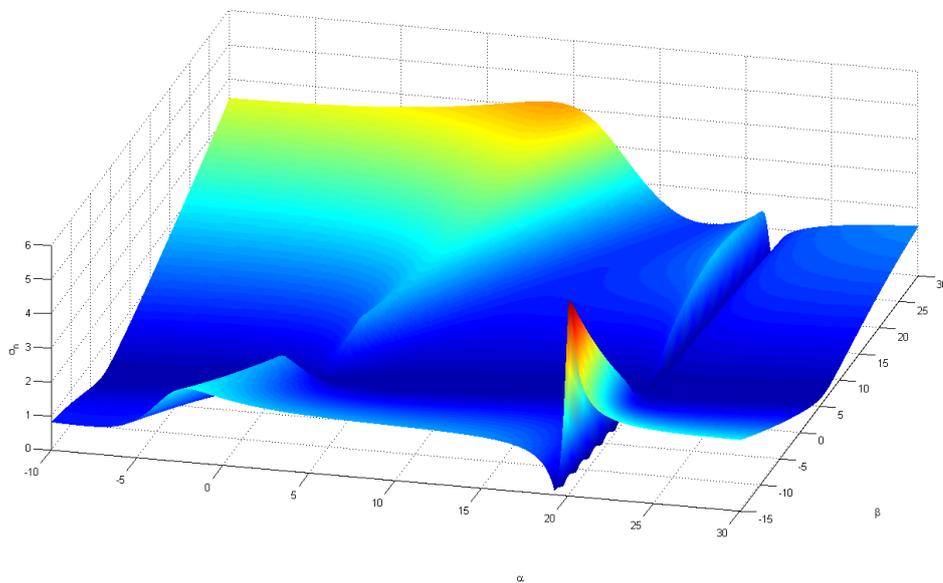


Abbildung 5.2: minimaler Singulärwert von \mathbf{H}_{py} über α, β , Beispiel (5.10)

Abbildung 5.3: minimaler Singulärwert von \mathbf{H}_{py} über α, β , Beispiel (5.11)Abbildung 5.4: minimaler Singulärwert von \mathbf{H}_{py} über α, β , Beispiel (5.12)

Die Abbildungen 5.1 bis 5.4 zeigen den Verlauf des minimalen Singulärwertes der folgenden Beispiele:

$$\mathbf{A}_1 = \begin{bmatrix} -81 & -56 & 57 & 11 \\ 146 & 102 & 106 & 20 \\ 62 & 43 & -46 & 9 \\ 203 & 138 & -149 & 31 \end{bmatrix}, \quad \mathbf{C}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (5.9)$$

$$\mathbf{A}_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 2 & \sqrt{2} \\ 0 & 0 & -\sqrt{2} & 2 \end{bmatrix}, \quad \mathbf{C}_2 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix} \quad (5.10)$$

$$\mathbf{A}_3 = \begin{bmatrix} -6 & 0 & 0 \\ 0 & -4 & 0 \\ 0 & 0 & -2 \end{bmatrix}, \quad \mathbf{C}_3 = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 1 & 1 \end{bmatrix} \quad (5.11)$$

$$\mathbf{A}_4 = \begin{bmatrix} -2 & -43.2 & -68.4 \\ 0 & -79.4 & -118.8 \\ 0 & 39.6 & 59.2 \end{bmatrix}, \quad \mathbf{C}_4 = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 2 & 5 \end{bmatrix} \quad (5.12)$$

Das Beispiel $(\mathbf{A}_1, \mathbf{C}_1)$ stammt aus [6].

5.4 Lösungsansätze

Für die Suche nach dem Wertepaar (α, β) , das den Rang der Matrix \mathbf{H}_{py} verringert, bzw. nach dem globalen Minimum von $f_{\mathbf{H}_{py}}(\alpha, \beta)$ werden eine Reihe von verschiedenen Lösungsansätzen vorgestellt.

5.4.1 Test nach Gao und Neumann

Das Problem der Überprüfung von perspektivischer Beobachtbarkeit ist dem Problem der Überprüfung von Beobachtbarkeit ähnlich. Ein System ist genau dann beobachtbar, wenn das Rangkriterium nach Popov-Belevitch-Hautus (2.19) erfüllt ist. Daraus kann ein Beobachtbarkeitsmaß abgeleitet werden, das angibt, wie gut beobachtbar ein System ist (vergleiche dazu [12]):

$$\mu(\mathbf{A}, \mathbf{C}) = \min_{\lambda \in \mathbb{C}} \sigma_n \left(\begin{bmatrix} \mathbf{A} - \lambda \mathbf{I} \\ \mathbf{C} \end{bmatrix} \right) \quad (5.13)$$

Zur Ermittlung von μ gibt es numerisch stabile Verfahren, wie zum Beispiel [13], [14] oder [15]. Diese Verfahren sind numerische Tests, die überprüfen ob ein Wert

δ eine obere Schranke für $\mu(\mathbf{A}, \mathbf{C})$ ist.

Formt man \mathbf{H}_{py} ein wenig um, so sieht man die erwähnte Ähnlichkeit:

$$\mathbf{H}_{py} = \begin{bmatrix} \mathbf{A}^2 + \alpha\mathbf{A} + \beta\mathbf{I} \\ \mathbf{C} \end{bmatrix} \quad (5.14)$$

$$= \begin{bmatrix} \tilde{\mathbf{A}}_\alpha - \beta\mathbf{I} \\ \mathbf{C} \end{bmatrix} \quad (5.15)$$

$$\text{mit } \tilde{\mathbf{A}}_\alpha = -\mathbf{A}^2 - \alpha\mathbf{A}$$

Für ein festes α entsteht jetzt das Minimierungsproblem

$$\min_{\beta \in \mathbb{R}} \sigma_n \left(\begin{bmatrix} \tilde{\mathbf{A}}_\alpha - \beta\mathbf{I} \\ \mathbf{C} \end{bmatrix} \right) \quad (5.16)$$

mit dem einzigen Unterschied zu (5.13), daß β über \mathbb{R} anstatt über \mathbb{C} minimiert wird. In [14] wird genau für dieses Teilproblem ein Algorithmus angegeben.

So kann man für ein festes α eine obere Schranke $\delta(\alpha)$ finden und die zweidimensionale Suche nach dem globalen Minimum von (5.8) kann auf eine Dimension eingeschränkt werden.

Leider zeigt sich, daß die Suche nach einem globalen Minimum von $\delta(\alpha)$ die selben Schwierigkeiten aufweist, wie das ursprüngliche Problem.

In Abbildung 5.5 ist der Verlauf von $\delta(\alpha)$ für Beispiel (5.11) gezeigt. Man erkennt, daß die dargestellte Funktion n lokale Minima besitzt. Damit treten bei einer numerischen Suche nach dem globalen Minimum die in Abschnitt 5.3 erwähnten Probleme auf.

5.4.2 Numerische Optimierungsverfahren 1. und 2. Ordnung

Numerische Optimierungsverfahren 1. und 2. Ordnung sind dadurch gekennzeichnet, daß mit Hilfe der 1. und 2. Ableitung der Zielfunktion ein Minimum gesucht wird. Grundsätzlich zeigen solche Verfahren die in Abschnitt 5.3 vorgestellten Probleme auf. Zusätzlich läßt sich die Zielfunktion (5.7) des vorliegenden Minimierungsproblems nicht analytisch ableiten. Deswegen ist eine numerische Differentiation notwendig, die allfällige numerische Ungenauigkeiten verstärkt. Aus diesem Grund wird an dieser Stelle nicht weiter auf solche Methoden eingegangen. Zusätzlich wurde festgestellt, daß Methoden 0. Ordnung, die ohne Ableitungen auskommen, ausreichend gut funktionieren.

5.4.3 Genetische Algorithmen

Bei genetischen Algorithmen wird am Anfang zufällig eine Menge (Population) aus verschiedenen Individuen erstellt. Jedes Individuum besitzt ein Genom, das

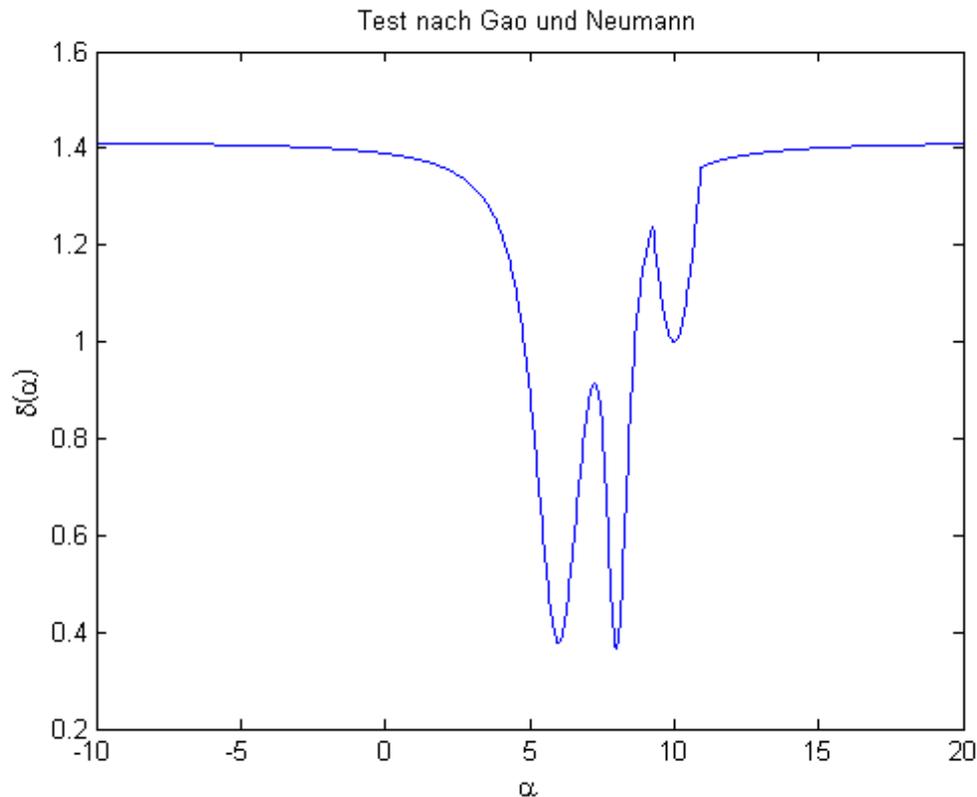


Abbildung 5.5: obere Schranke für den minimalen Singulärwert in Abhängigkeit von α , Beispiel (5.11)

eine mögliche Lösung des Problems repräsentiert. Beim vorliegenden Problem besteht das Genom aus dem Wertepaar (α, β) . Die Individuen werden über viele Generationen hinweg evolviert. Dazu wird für jedes Individuum ein Fitnesswert ermittelt, der proportional zur Qualität der Lösung ist. Im vorliegenden Fall steht ein hoher Fitnesswert für einen niedrigen Funktionswert an der Stelle (α, β) . Im nächsten Schritt werden die Individuen zufällig ausgewählt, wobei die Auswahlwahrscheinlichkeit proportional zum Fitnesswert ist. Die ausgewählten Individuen werden dann entweder direkt in die nächste Generation übernommen oder sie „paaren“ sich mit einem zweiten Individuum. Bei der Paarung werden die Genome der Individuen untereinander gekreuzt, indem Teile des Genoms zufällig zwischen den Partnern getauscht werden. So entstehen zwei neue Individuen. Anschließend wird bei zufällig ausgewählten Individuen das Erbgut noch mutiert (es wird an zufälligen Stellen ein Rauschen addiert).

Über viele Generationen kann ein genetischer Algorithmus gute Ergebnisse für ein Minimierungsproblem liefern, selbst wenn deterministische numerische Methoden

versagen. Der große Vorteil des Verfahrens ist die vielfältige Anwendung des Zufalls. So ist es möglich, daß bei der Suche kurzfristig eine Verschlechterung des Ergebnisses in Kauf genommen wird, damit längerfristig ein besseres Ergebnis erreicht werden kann. Ein Beispiel dafür wäre, daß in der Funktionslandschaft ein Berg überwunden werden kann, damit man ins nächste, tieferliegende Tal gelangt (vergleiche dazu [16]).

Für das vorliegende Problem wird die Startpopulation rund um die Kandidatenmenge \mathbb{S} generiert, sodaß für jeden Kandidaten eine Anzahl an Individuen erzeugt wird. Für folgendes Beispiel

$$\mathbf{A} = \begin{bmatrix} -6 & 0 & 0 \\ 0 & -4 & 0 \\ 0 & 0 & -2 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (5.17)$$

zeigt Abbildung 5.6 die Verteilung der Individuen in der (α, β) -Ebene. Man erkennt, daß sich bereits nach zehn Generationen alle Individuen um das globale Minimum bei $(8, 12)$ versammeln und an den anderen Startpunkten keine Individuen überlebt haben. Abbildung 5.7 zeigt den Verlauf der minimalen Singulärwerte über die Generationen. Man sieht, daß der Algorithmus für dieses Beispiel das globale Minimum findet, allerdings beträgt der Funktionswert des besten Individuums der letzten Generation $3.57 \cdot 10^{-5}$. Berechnet man mit MATLAB die Eigenwerte und anschließend den Singulärwert, so erhält man 0.

Man erkennt, daß genetische Algorithmen unter vielen lokalen Minima das globale Minimum finden können. Allerdings haben sie Schwierigkeiten exakt die Stelle des Rangverlustes zu finden, sie finden eher eine Lösung in der Nähe der gesuchten Stelle.

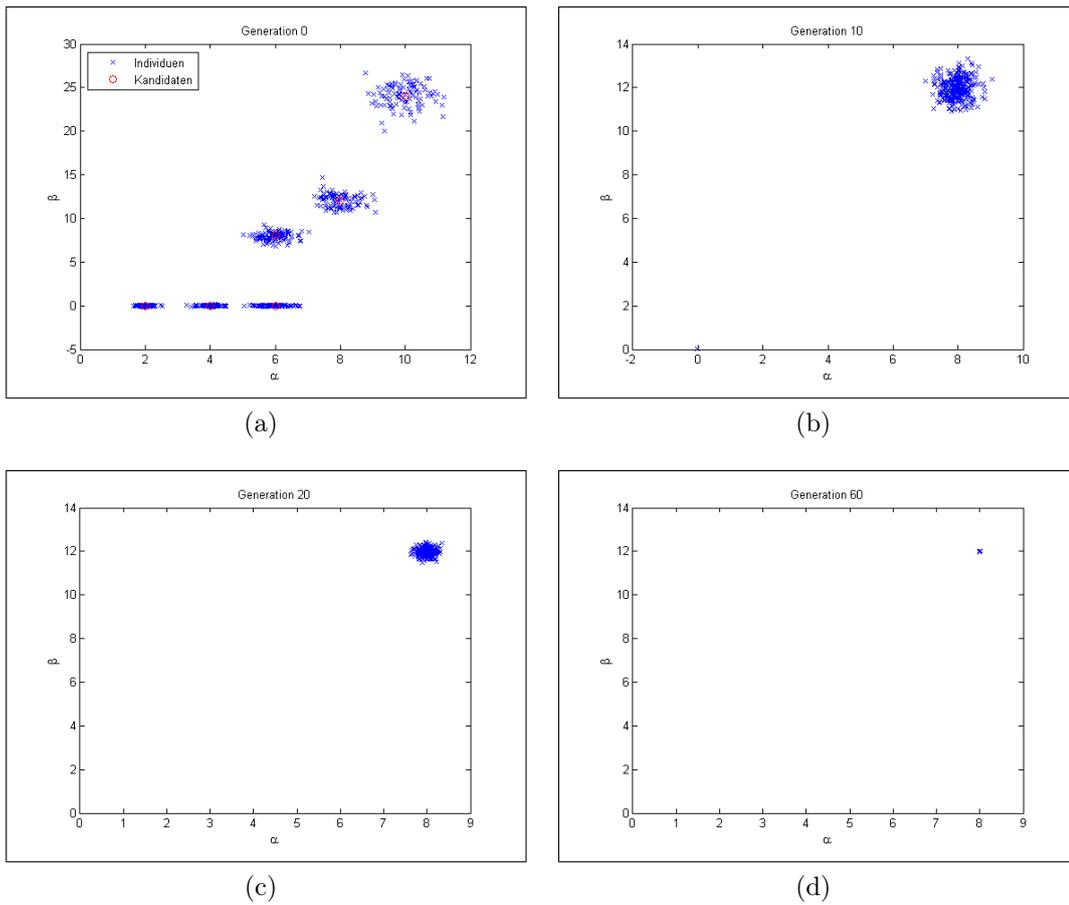


Abbildung 5.6: alle Individuen der Population (a) beim Start, (b) nach 10 Generationen, (c) nach 20 Generationen und (d) nach 60 Generationen

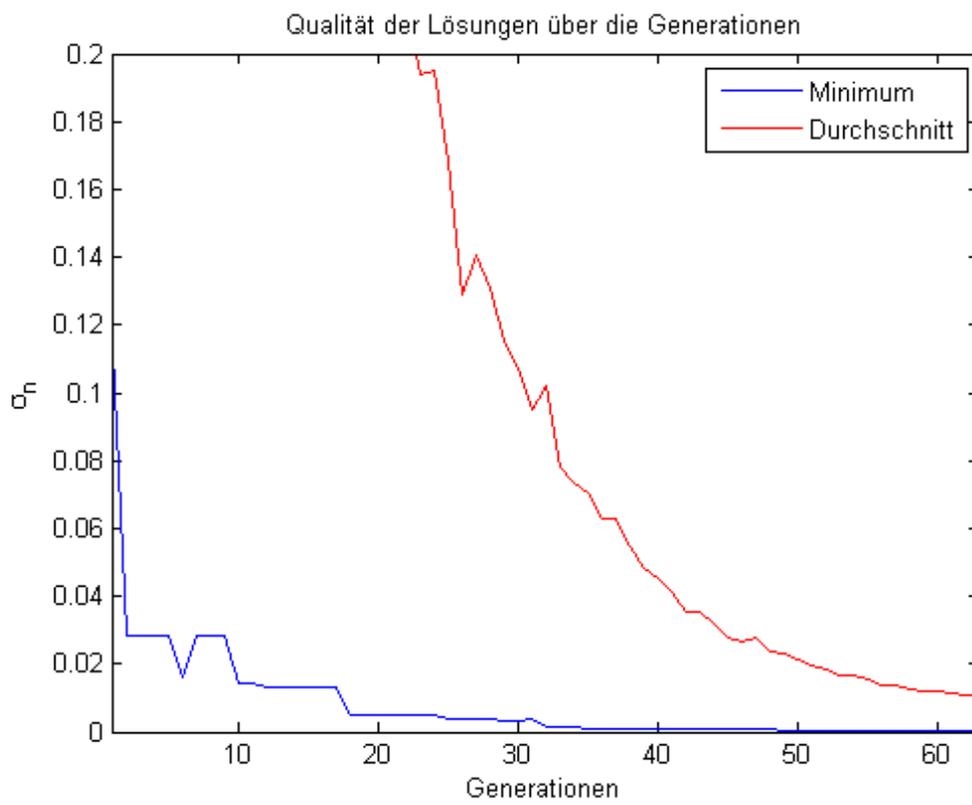


Abbildung 5.7: Verlauf des minimalen Singulärwertes über die Generationen

5.4.4 Simplex-Verfahren

Das Simplex-Verfahren ist ein numerisches Optimierungsverfahren 0. Ordnung. Das heißt, das Verfahren sucht das Minimum nur mit Hilfe der Zielfunktion, nicht aber mit den Ableitungen der Funktion. In MATLAB ist standardmäßig die Funktion `fminsearch` enthalten, die ein Simplex-Verfahren nach [17] ausführt.

Beim Simplex-Verfahren wird mit einem Simplex, das ist ein regelmäßiges Polygon mit $n + 1$ Kanten bei einem n -dimensionalen Problem, ein Minimum gesucht. Bei einem zweidimensionalen Problem ist ein Simplex ein gleichseitiges Dreieck. Das Verfahren basiert auf zwei verschiedenen Regeln. Normalerweise werden die Funktionswerte der Eckpunkte untereinander verglichen. Der Eckpunkt mit dem höchsten Funktionswert fällt aus dem Simplex hinaus. Statt dessen wird ein neuer Eckpunkt eingefügt, der symmetrisch zum alten ist. Die Symmetrieachse ist hierbei der restliche Teil des Simplex (siehe Abbildung 5.8a). Bei der iterativen Anwendung dieser Regel kann der Fall auftreten, daß ein bestimmter Eckpunkt x' wiederholt, innerhalb einer bestimmten Anzahl an Iterationen, in den Simplex aufgenommen wird. Tritt dieser Fall ein, so wird eine zweite Regel angewandt. Die Strecken des Eckpunktes x' werden halbiert und daraus entsteht der neue Simplex (siehe Abbildung 5.8b) (vergleiche dazu [18]).

Da der Algorithmus deterministisch ist, hängt von der Wahl des Startpunktes ab, welches Minimum gefunden wird. Besitzt man kein weiteres Wissen über das Minimierungsproblem treten die in Abschnitt 5.3 beschriebenen Schwierigkeiten auf.

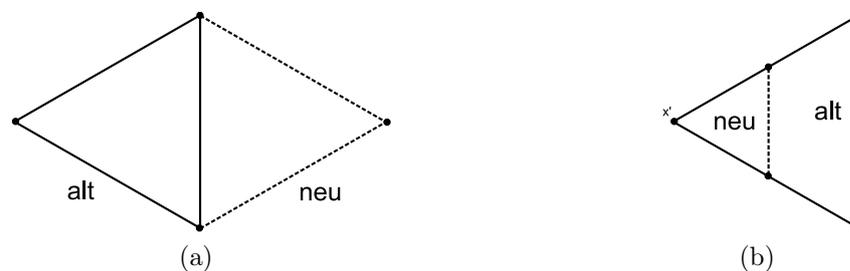


Abbildung 5.8: (a) Reflexion des Simplex, (b) Kontraktion des Simplex

5.4.5 Simplex-Verfahren mit „intelligenten“ Startpunkten

In Kapitel 3 wurde eine Menge von Kandidaten erarbeitet, bei denen der Rang von \mathbf{H}_{py} kleiner n sein könnte. Normalerweise sind die Eigenwerte der Systemmatrix nicht exakt bekannt, was in weiterer Folge bedeutet, daß man die Kandidatenmenge \mathbb{S} ebenfalls nicht exakt berechnen, sondern sie nur „schätzen“ (im Sinne von

fehlerbehaftet berechnen) kann. Die Menge der geschätzten Kandidaten wird im weiteren Verlauf mit $\hat{\mathcal{S}}$ bezeichnet.

Mit anderen Worten, man hat eine Liste von Punkten, in deren Nähe das gesuchte Minimum mit Funktionswert 0, falls es denn existiert, sein muß. Diese Punkte können als Startpunkte für das Simplex-Verfahren verwendet werden, um das gesuchte Minimum zu finden.

Betrachtet man noch einmal das Beispiel aus Abschnitt 5.1: Der entsprechende Kandidat $(-3 - 5.02 \cdot 10^{-13}, 2 + 1.05 \cdot 10^{-11})$ aus $\hat{\mathcal{S}}$ hatte den Funktionswert $3.31 \cdot 10^{-12}$. Dieser Punkt dient nun als Startpunkt für den Simplex-Algorithmus; nach 105 Iterationen liefert dieser einen Funktionswert von $9.25 \cdot 10^{-14}$ zurück. Dieser Wert ist zwar noch immer höher als der im Minimum $(-3, 2)$, aber mit MATLAB erkennt man trotzdem den Rangverlust der Matrix.

Die Eigenschaften dieser Methode kann man folgenden, theoretisch konstruierten, Beispiel zeigen. Das System sieht wie folgt aus:

$$\mathbf{A} = \begin{bmatrix} -1 & \epsilon & 0 & 0 \\ -\epsilon & -1 & 0 & 0 \\ 0 & 0 & -1 - \epsilon & 0 \\ 0 & 0 & 0 & -1 + \epsilon \end{bmatrix}, \quad (5.18)$$

$$\epsilon = 0.001$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 2 & 4 \\ 0 & 1 & 5 & 10 \end{bmatrix} \quad (5.19)$$

$$s = \{-1 + \epsilon, -1 - \epsilon, -1 \pm j\epsilon\}$$

Man erkennt die zwei „parallelen“ Spalten von \mathbf{C} , die das System nicht perspektivisch beobachtbar machen. Bei $\alpha = 2$ und $\beta = 1 - \epsilon^2$ tritt ein Rangverlust von \mathbf{H}_{py} ein.

Schätzt man nun die Eigenwerte des System fälschlicherweise auf einen vierfachen Eigenwert bei -1 , so ergibt sich die Menge $\hat{\mathcal{S}}$ als

$$\hat{\mathcal{S}} = \{(2, 1), (1, 0)\}. \quad (5.20)$$

Der Simplex-Algorithmus startet beim Punkt $(2, 1)$, der den Funktionswert $1 \cdot 10^{-6}$ besitzt. Nach 113 Iterationen findet er das globale Minimum bei $(2, 1 - \epsilon^2)$ mit dem Funktionswert $4.97 \cdot 10^{-17}$; der Rangverlust der Matrix \mathbf{H}_{py} wird von MATLAB erkannt.

Man ist aber trotzdem noch darauf angewiesen, daß man sich so verrechnet, das die Startpunkte „günstig“ liegen. Läßt man den Simplex-Algorithmus zum Beispiel beim (willkürlichen) Punkt $(1.8, 0.81)$ starten, so wird nur das lokale Minimum

bei $(2, 1 + \epsilon^2)$ mit dem Funktionswert $1.65 \cdot 10^{-7}$ gefunden; nicht aber das globale Minimum.

5.4.6 Simplex-Verfahren mit zufällig gewählten Startpunkten

Der Ansatz aus Abschnitt 5.4.5 kann verfeinert werden: Die Menge der Startpunkte $\hat{\mathbb{S}}$ wird mit zufällig gewählten Punkten erweitert. Es entsteht so die Menge $\tilde{\mathbb{S}}$, die zusätzlich für jedes Element aus $\hat{\mathbb{S}}$ eine bestimmte Anzahl an zufälligen Punkten in der Umgebung des Elementes besitzt:

$$\tilde{\mathbb{S}} = \hat{\mathbb{S}} \cup \left\{ (\mathcal{N}_i(\hat{\alpha}, 0.1 + 0.1\hat{\alpha}), \mathcal{N}_i(\hat{\beta}, 0.1 + 0.1\hat{\beta})) \mid (\hat{\alpha}, \hat{\beta}) \in \hat{\mathbb{S}}, i = 1, \dots, 10 \right\} \quad (5.21)$$

Wobei $\mathcal{N}(0, 1)$ für eine Normalverteilung mit Mittelwert 0 und Standardabweichung 1 steht und der Index i verdeutlichen soll, daß für jedes Element aus $\hat{\mathbb{S}}$ 10 Elemente zu $\tilde{\mathbb{S}}$ hinzugefügt werden. Die Addition der Konstante 0.1 bei der Standardabweichung beruht darauf, daß die Verteilung auch bei $\hat{\alpha} = 0$ (bzw. $\hat{\beta} = 0$) eine Standardabweichung größer null besitzen soll.

Betrachtet man nun das Beispiel (5.18), (5.19) nochmals, so erkennt man, daß sowohl das globale Minimum bei $(2, 1 - \epsilon^2)$ als auch das lokale bei $(2, 1 + \epsilon^2)$ gefunden werden, wenn man in der Umgebung des falschen Startpunkts $(1.8, 0.81)$ zufällig 10 Punkte generiert.

5.5 Algorithmus

Zusammenfassend sieht ein Algorithmus für eine numerisch stabile Überprüfung der Bedingung (2.49) folgendermaßen aus:

1. Berechnen der Eigenwerte der Systemmatrix \mathbf{A} (eventuell fehlerbehaftet)
2. Ermitteln der Menge $\hat{\mathbb{S}}$ als Startpunkte für die numerische Suche (siehe Gleichung (3.10))
3. optional: Ermitteln der Menge $\tilde{\mathbb{S}}$ als Startpunkte für die numerische Suche durch Hinzufügen von zufällig verteilten Punkten in einer Umgebung der Kandidaten aus $\hat{\mathbb{S}}$
4. Für jeden Startpunkt:
 - (a) numerische Suche nach einem Minimum mit dem Simplex-Verfahren

-
- (b) Überprüfung, ob bei dem gefundenen Minimum ein Rangverlust von \mathbf{H}_{py} eintritt
 - i. Wenn ja: System ist nicht perspektivisch beobachtbar.
 - ii. Wenn nein: weiter
 - 5. Wurde ein Rangverlust bei allen Startpunkten ausgeschlossen: Das System ist perspektivisch beobachtbar.

In Abschnitt A.1 findet man eine Implementierung dieses Algorithmus.

Kapitel 6

Maß für die perspektivische Beobachtbarkeit

6.1 Steuerbarkeitsmaß nach Paige

Wie bereits in Abschnitt 2.7 erwähnt, kann das Steuerbarkeitsmaß μ nach [8] über die Beziehung (2.55) ermittelt werden. Der Zusammenhang, der in [9] vorgestellt wurde, ist dabei wie folgt:

Für das nicht steuerbare System $(\mathbf{A} + \delta\mathbf{A}, \mathbf{B} + \delta\mathbf{B})$ ist das Rangkriterium nach Popov-Belevitch-Hautus nicht erfüllt, das heißt es

$$\begin{aligned} \exists \lambda \in \mathbb{C} : \text{rang}([\mathbf{A} + \delta\mathbf{A} - \lambda\mathbf{I}, \mathbf{B} + \delta\mathbf{B}]) = \\ \text{rang}(\underbrace{[\mathbf{A} - \lambda\mathbf{I}, \mathbf{B}]}_{\mathbf{H}_y} + \underbrace{[\delta\mathbf{A}, \delta\mathbf{B}]}_{\Delta\mathbf{H}_y}) < n \end{aligned} \quad (6.1)$$

Mit Hilfe der Singulärwertzerlegung gilt

$$[\mathbf{A} - \lambda\mathbf{I}, \mathbf{B}] = \sum_{i=1}^n u_i \sigma_i v_i^T \quad (6.2)$$

$$\text{wobei } \|u_i\| = \|v_i\| = 1$$

Damit der Rang des Ausdrucks (6.1) kleiner als n wird, muß mindestens ein Singulärwert gleich null sein. Damit die Norm von $[\delta\mathbf{A}, \delta\mathbf{B}]$ minimal wird, wählt man

$$\Delta\mathbf{H}_y = [\delta\mathbf{A}, \delta\mathbf{B}] = -u_n \sigma_n v_n^T \quad (6.3)$$

Damit ergibt sich

$$\|[\delta\mathbf{A}, \delta\mathbf{B}]\|_2 = \sigma_n \quad (6.4)$$

und so gilt Gleichung (2.55).

6.1.1 Übertragung auf die perspektivische Beobachtbarkeit

Überträgt man die Idee aus [8] auf das Konzept der perspektivischen Beobachtbarkeit, so erhält man folgende Maßdefinition:

$$\mu_p(\mathbf{A}, \mathbf{C}) = \min_{\delta\mathbf{A}, \delta\mathbf{C}} \left\{ \left\| \begin{bmatrix} \delta\mathbf{A} \\ \delta\mathbf{C} \end{bmatrix} \right\|_2 : \right. \\ \left. (\mathbf{A} + \delta\mathbf{A}, \mathbf{C} + \delta\mathbf{C}) \text{ nicht perspektivisch beobachtbar} \right\} \quad (6.5)$$

Der Zusammenhang aus [9] gilt in diesem Fall aber nicht mehr:

Die Matrix \mathbf{H}_{py} zur Überprüfung der perspektivischen Beobachtbarkeit für das System $(\mathbf{A} + \delta\mathbf{A}, \mathbf{C} + \delta\mathbf{C})$ sieht folgendermaßen aus:

$$\mathbf{H}_{py} = \mathbf{H}_{py}(\mathbf{A} + \delta\mathbf{A}, \mathbf{C} + \delta\mathbf{C}) \quad (6.6)$$

$$= \begin{bmatrix} (\mathbf{A} + \delta\mathbf{A})^2 + \alpha(\mathbf{A} + \delta\mathbf{A}) + \beta\mathbf{I} \\ \mathbf{C} + \delta\mathbf{C} \end{bmatrix} \quad (6.7)$$

$$= \begin{bmatrix} \mathbf{A}^2 + \mathbf{A}\delta\mathbf{A} + \delta\mathbf{A}\mathbf{A} + (\delta\mathbf{A})^2 + \alpha\mathbf{A} + \alpha\delta\mathbf{A} + \beta\mathbf{I} \\ \mathbf{C} + \delta\mathbf{C} \end{bmatrix} \quad (6.8)$$

$$= \underbrace{\begin{bmatrix} \mathbf{A}^2 + \alpha\mathbf{A} + \beta\mathbf{I} \\ \mathbf{C} \end{bmatrix}}_{\mathbf{H}_{py}(\mathbf{A}, \mathbf{C})} + \underbrace{\begin{bmatrix} \mathbf{A}\delta\mathbf{A} + \delta\mathbf{A}\mathbf{A} + (\delta\mathbf{A})^2 + \alpha\delta\mathbf{A} \\ \delta\mathbf{C} \end{bmatrix}}_{\Delta\mathbf{H}_{py}} \quad (6.9)$$

Es ist ersichtlich, daß die Matrix $\Delta\mathbf{H}_{py}$ nicht mehr der gesuchten Matrix $\begin{bmatrix} \delta\mathbf{A} \\ \delta\mathbf{C} \end{bmatrix}$ entspricht und daher

$$\mu_p(\mathbf{A}, \mathbf{C}) \neq \min_{\alpha, \beta \in \mathbb{R}} \sigma_n \left(\begin{bmatrix} \mathbf{A}^2 + \alpha\mathbf{A} + \beta\mathbf{I} \\ \mathbf{C} \end{bmatrix} \right) \quad (6.10)$$

gilt. Der Lösungsansatz, die Matrix $\begin{bmatrix} \delta\mathbf{A} \\ \delta\mathbf{C} \end{bmatrix}$ mit Hilfe der Kenntnis von $\Delta\mathbf{H}_{py}$ auszurechnen, erscheint nicht zielführend, weil eine Lösung für $\delta\mathbf{A}$ nicht eindeutig wäre und daher

$$\min_{\Delta\mathbf{H}_y} \|\Delta\mathbf{H}_y\|_2 \neq \min_{\delta\mathbf{A}, \delta\mathbf{C}} \left\| \begin{bmatrix} \delta\mathbf{A} \\ \delta\mathbf{C} \end{bmatrix} \right\|_2 \quad (6.11)$$

gilt.

6.2 Steuerbarkeitsmaß nach Eising

In [10] wird ein zu [8] leicht verschiedener Vorschlag für ein Steuerbarkeitsmaß vorgestellt. Dieser Vorschlag basiert auf der Tatsache, daß ein nicht steuerbares

System den Eigenvektor-Test nach Popov-Belevitch-Hautus (2.17) verletzt. Für einen gegebenen (Eigen-)Vektor x können so die Matrizen $\delta\mathbf{A}$ und $\delta\mathbf{B}$ bzw. deren Norm ausgerechnet werden. Für die Maßberechnung wird schließlich über alle x minimiert.

6.2.1 Übertragung auf die perspektivische Beobachtbarkeit

Die Idee aus [10] läßt sich auf das Konzept der perspektivischen Beobachtbarkeit übertragen:

$$\Delta_p(\mathbf{A}, \mathbf{C}) = \min_{\delta\mathbf{A}, \delta\mathbf{C}} \{d_A \|\delta\mathbf{A}\|_2 + d_C \|\delta\mathbf{C}\|_2 : (\mathbf{A} + \delta\mathbf{A}, \mathbf{C} + \delta\mathbf{C}) \text{ nicht perspektivisch beobachtbar}\} \quad (6.12)$$

Ein nicht perspektivisch beobachtbares System verletzt das Kriterium (3.45). Das heißt, es existiert entweder ein reeller Eigenvektor von \mathbf{A} , der normal auf die Ausgangsmatrix \mathbf{C} steht, oder es existieren zwei Eigenvektoren deren Produkte mit \mathbf{C} linear abhängig sind und deren Summe reell ist.

Zur Berechnung von $\delta\mathbf{A}$ geht man von der Gleichung

$$\mathbf{A}x = \mathbf{A}x \quad (6.13)$$

aus. Nun wird auf der rechten Seite ein Term addiert, gleich wieder subtrahiert und anschließend ein wenig umgeformt:

$$\mathbf{A}x = \mathbf{A}x + x \frac{x^H \mathbf{A}x}{x^H x} - x \frac{x^H \mathbf{A}x}{x^H x} \quad (6.14)$$

$$= x \underbrace{\frac{x^H \mathbf{A}x}{x^H x}}_{\lambda_x} + \underbrace{\left(\mathbf{A}x - x \frac{x^H \mathbf{A}x}{x^H x} \right)}_{x_0} \quad (6.15)$$

$$= x\lambda_x + x_0 \quad (6.16)$$

Für die Matrix $\mathbf{A} + \delta\mathbf{A}$ ergibt sich die Eigenvektorgleichung einmal als

$$(\mathbf{A} + \delta\mathbf{A})x = \lambda x \quad \text{für ein } \lambda \in \mathbb{C} \quad (6.17)$$

und durch Einsetzen der Beziehung (6.16) als:

$$(\mathbf{A} + \delta\mathbf{A})x = \lambda_x x + x_0 + \delta\mathbf{A}x \quad (6.18)$$

Setzt man die beiden linken Seiten der Gleichungen (6.17) und (6.18) gleich und drückt $\delta\mathbf{A}$ aus, so erhält man

$$\delta\mathbf{A}x = -x_0 + \xi x \quad \text{für ein } \xi \in \mathbb{C} \quad (6.19)$$

Die Lösung für $\delta\mathbf{A}$ mit $\|\delta\mathbf{A}\|$ minimal, ergibt sich aus (nach [19])

$$\delta\mathbf{A} = \frac{-x_0 + \xi x}{x^H x} x^H \quad \text{für ein } \xi \in \mathbb{C} \quad (6.20)$$

Damit $\delta\mathbf{A}$ minimal wird, wird $\xi = 0$ gewählt und man erhält somit durch Rückeinsetzen von (6.16)

$$\delta\mathbf{A} = \left(-\mathbf{A}x + x \frac{x^H \mathbf{A}x}{x^H x} \right) x^H / x^H x \quad (6.21)$$

Man hat nun eine Berechnungsvorschrift für $\delta\mathbf{A}$ mit minimaler Norm in Abhängigkeit von x und \mathbf{A} erhalten.

Für die Berechnung von $\delta\mathbf{C}$ gibt es zwei verschiedene Fälle:

Im ersten Fall steht der reelle Vektor x normal auf die Ausgangsmatrix:

$$(\mathbf{C} + \delta\mathbf{C})x = 0 \quad (6.22)$$

$$\delta\mathbf{C}x = -\mathbf{C}x \quad (6.23)$$

Die Lösung für $\delta\mathbf{C}$ mit minimaler Norm lautet dann

$$\delta\mathbf{C} = -\frac{\mathbf{C}x}{x^H x} x^H \quad (6.24)$$

Im zweiten Fall existiert ein zweiter Eigenvektor x_j mit den Eigenschaften

$$(\mathbf{A} + \delta\mathbf{A})x_j = \lambda_j x_j \quad (6.25)$$

$$\lambda \neq \lambda_j \quad (6.26)$$

$$x + x_j \in \mathbb{R}^n \quad (6.27)$$

und

$$(\mathbf{C} + \delta\mathbf{C})x + r(\mathbf{C} + \delta\mathbf{C})x_j = 0 \quad (6.28)$$

$$\delta\mathbf{C}(x + rx_j) = -\mathbf{C}(x + rx_j) \quad (6.29)$$

für ein $r \in \mathbb{C}$. Hier lautet die Lösung für $\delta\mathbf{C}$ mit minimaler Norm

$$\delta\mathbf{C} = -\frac{\mathbf{C}(x + rx_j)}{(x + rx_j)^H (x + rx_j)} (x + rx_j)^H \quad (6.30)$$

Fügt man nun alle Erkenntnisse zusammen, erhält man folgende Aussage \textcircled{M} :
 Man betrachtet ein System (\mathbf{A}, \mathbf{C}) , einen Vektor $x \in \mathbb{C}^n \neq 0$ und die Koeffizienten $d_A, d_C > 0$. Mit der Wahl von $\delta\mathbf{A}$ und $\delta\mathbf{C}$ als

$$\delta\mathbf{A} = \left(-\mathbf{A}x + x \frac{x^H \mathbf{A}x}{x^H x}\right) x^H / x^H x \quad (6.31)$$

$$\delta\mathbf{C}_0 = \begin{cases} -\frac{\mathbf{C}xx^H}{x^H x} & \text{für } x \in \mathbb{R}^n \\ \infty & \text{sonst} \end{cases} \quad (6.32)$$

$$\min_{r \in \mathbb{C}} \delta\mathbf{C}_j = \begin{cases} -\frac{\mathbf{C}(x + rx_j)(x + rx_j)^H}{(x + rx_j)^H(x + rx_j)} & \text{für } x + x_j \in \mathbb{R}^n \text{ und } \lambda \neq \lambda_j \\ \infty & \text{sonst} \end{cases} \quad (6.33)$$

$$\delta\mathbf{C} = \min_{j=0, \dots, n-1} \delta\mathbf{C}_j \quad (6.34)$$

ist das System $(\mathbf{A} + \delta\mathbf{A}, \mathbf{C} + \delta\mathbf{C})$ nicht perspektivisch beobachtbar und $d_A \|\delta\mathbf{A}\|_2 + d_C \|\delta\mathbf{C}\|_2$ minimal.

Das Maß für die perspektivische Beobachtbarkeit (6.12) ergibt sich somit durch die Minimierung des Ausdruckes $d_A \|\delta\mathbf{A}\|_2 + d_C \|\delta\mathbf{C}\|_2$ über alle $x \in \mathbb{C}^n$ mit der Eigenschaft $\|x\| = 1$. Symbolisch kann das als

$$\Delta_p(\mathbf{A}, \mathbf{C}) = \min_{x \in \mathbb{C}^n, \|x\|=1} \textcircled{M} \quad (6.35)$$

dargestellt werden.

6.2.2 Berechnung des Maßes für die perspektivische Beobachtbarkeit

Um das Maß (6.12) zu berechnen, muß zuerst das das Minimierungsproblem (6.35) gelöst werden. Dazu wird wieder der, in Abschnitt 5.4.4 vorgestellte, Simplex-Algorithmus verwendet. Es ist zu erwarten, daß die Eigenvektoren des nicht perspektivisch beobachtbaren Systems $(\mathbf{A} + \delta\mathbf{A}, \mathbf{C} + \delta\mathbf{C})$ in der Nähe der Eigenvektoren von \mathbf{A} liegen, weil $\|\delta\mathbf{A}\|$ minimal werden soll. Aus diesem Grund werden die Eigenvektoren von \mathbf{A} als Startpunkte für den Simplex-Algorithmus verwendet.

Die Bedingung $\|x\| = 1$ ist eine nichtlineare Einschränkung, die im Simplex-Verfahren nicht berücksichtigt werden kann. Um hier algorithmische Schwierigkeiten zu vermeiden, werden die Vektoren in der zu minimierende Aussage \textcircled{M} einfach normiert.

Um die Störmatrizen $\delta\mathbf{C}_j$ ($j = 1, \dots, n - 1$) zu berechnen, muß ein weiteres Minimierungsproblem gelöst werden. Auch für dieses Problem wird der Simplex-Algorithmus verwendet. Da hier das Skalar r sowohl negativ als auch positiv sein kann, startet die Minimierung bei $r = 0$.

In Abschnitt A.3 findet man eine Implementierung dieses Algorithmus.

6.2.3 Beispiel

In Abbildung 6.1 ist der Verlauf von $\Delta_p(\mathbf{A}, \mathbf{C})$ des Beispiels

$$\mathbf{A} = \begin{bmatrix} -2 & 0 & 0 \\ 0 & -4 & 0 \\ 0 & 0 & -6 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 & 2 & 0 \\ 0 & \alpha & 1 \end{bmatrix} \quad (6.36)$$

in Abhängigkeit von α dargestellt. Bei $\alpha = 0$ geht die perspektivische Beobachtbarkeit verloren; die ersten beiden Spalten von \mathbf{C} sind linear abhängig. Dies wird durch das Maß wiedergeben. Man sieht auch, daß das Maß stetig in α ist.

6.2.4 Probleme

Bei der Berechnung des Maßes kann, vor allem bei komplexen Eigenvektoren bzw. Eigenwerten der Systemmatrix, der Fall auftreten, daß die Störungsmatrizen $\delta\mathbf{A}, \delta\mathbf{C}$ komplex sind. Somit sind auch die Matrizen des Systems $(\mathbf{A} + \delta\mathbf{A}, \mathbf{C} + \delta\mathbf{C})$ komplex und es ist nicht mehr klar, ob das System die Eigenschaft perspektivische Beobachtbarkeit verliert.

Eine weitere Schwierigkeit ist, daß für perspektivische Beobachtbarkeit die Systemmatrix als stark regulär vorausgesetzt wird und die Irrationalitätsbedingung gelten muß. Dies ist für die entstehende Systemmatrix $\mathbf{A} + \delta\mathbf{A}$ nicht garantiert.

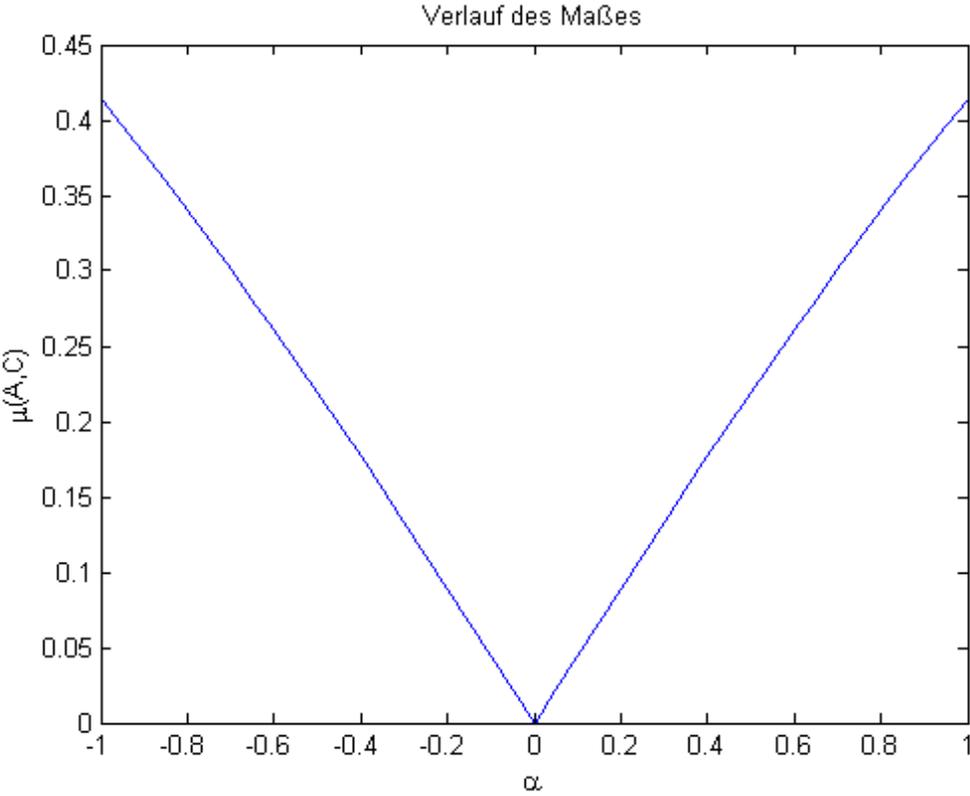


Abbildung 6.1: Verlauf des Maßes über α

Kapitel 7

Zusammenfassung und Ausblick

Zusammenfassend kann hervorgehoben werden, daß das Resultat aus [7] verbessert werden konnte. Durch die erfolgte Einschränkung auf eine endliche Parametermenge ist nun die Überprüfung der Eigenschaft perspektivischer Beobachtbarkeit praktisch möglich. Da die Genauigkeit dieser Überprüfung maßgeblich von der Kenntnis der Eigenwerte der Systemmatrix abhängt, wurde ein Algorithmus vorgestellt, der auch ohne exakte Kenntnis der Eigenwerte verlässliche Aussagen liefert.

Weiters wurde ein alternatives Kriterium zur Überprüfung perspektivischer Beobachtbarkeit gezeigt.

Das in dieser Arbeit vorgestellte Verfahren beseitigt somit die Probleme bei der Überprüfung der Rangbedingung (2.47). Bei der Überprüfung der Voraussetzungen existieren allerdings noch Schwierigkeiten. So ist die Irrationalitätsbedingung numerisch, mit endlicher Genauigkeit, nicht überprüfbar.

Eine mögliche zukünftige Fragestellung wäre, ob weitere Kriterien zur Überprüfung perspektivischer Beobachtbarkeit existieren, nachdem es für die „ähnlichen“ Eigenschaften Steuerbarkeit und Beobachtbarkeit eine Vielzahl an Kriterien gibt. Weiters gibt es in diesem Zusammenhang noch einige Begriffe, wie etwa die Steuerbarkeitsnormalform, die man gerne auf die perspektivische Beobachtbarkeit übertragen möchte um eine „perspektivische Beobachtbarkeitsnormalform“ aufstellen zu können.

In dieser Arbeit wird erstmals ein Maß für die perspektivische Beobachtbarkeit vorgestellt. Aus einer bereits existierenden Definition für ein Steuerbarkeitsmaß kann ein Maß für die perspektivische Beobachtbarkeit abgeleitet und berechnet werden. Aufgrund der einfacheren Berechnung werden bei der Maßdefinition komplexe Störungsmatrizen zugelassen. Durch dieses Zugeständnis ist allerdings nicht klar, ob bei einem komplexen Ergebnis das gefundene System tatsächlich nicht perspektivisch beobachtbar ist und ob in Folge das Maß korrekt ermittelt wurde. Auch bei der Berechnung, konkret beim Lösen der Minimierungsprobleme, ist noch Verbesserungspotential vorhanden.

Anhang A

Implementierung der Algorithmen

A.1 Überprüfung perspektivischer Beobachtbarkeit

Algorithmus A.1: Routine zur stabilen Überprüfung der perspektivischen Beobachtbarkeit

```
1 function [ja, alpha, beta] = pruefePerspektivischeBeobachtbarkeit(A,  
    C, zufall)  
2 % [ja, alpha, beta] = pruefePerspektivischeBeobachtbarkeit(A, C,  
    zufall)  
3 % überprüft die perspektivische Beobachtbarkeit des Systems (A, C)  
4 % Parameter:  
5 % A ... Systemmatrix  
6 % C ... Ausgangsmatrix  
7 % zufall ... bei zufall = 1 werden zufällige gewählte Punkte als  
8 %           Startpunkte für die Minimierung hinzugefügt; empfohlen,  
    wenn  
9 %           die Berechnung der Eigenwerte schwierig ist (Standard-  
    Vorgabe: 0)  
10 % Rückgabewerte:  
11 % ja ... -1 ... keine Entscheidung  
12 %       0 ... System ist nicht perspektivisch beobachtbar  
13 %       1 ... System ist perspektivisch beobachtbar  
14 % alpha, beta ... die Stelle, an der Verlust des Ranges auftritt  
15 % yes = 0 ... (A, C) persp. beobachtbar  
16  
17 absmin = inf;  
18  
19 % Standard-Vorgabe  
20 if nargin < 3  
21     zufall = 0;  
22 end  
23  
24 ja = -1; alpha = 0; beta = 0;
```

```

25
26 % Eigenwerte von A schätzen
27 eigs = eig(A);
28
29 % Kandidaten-Menge S ermitteln
30 S = ermittleKandidaten(eigs);
31 if zufall
32     for j = 1:size(S, 1)
33         for k = 1:10
34             S = [S; S(j, :) + (0.1 + 0.1.*S(j,:)) .* randn(1, 2)];
35         end
36     end
37 end
38
39 % Optionen für die Simplex-Suche
40 n = size(A, 2);
41 optionen = optimset('Display','off', ...
42                    'TolFun', n * eps(norm(A)), ...
43                    'MaxFunEvals', 10000, ...
44                    'MaxIter', 10000, ...
45                    'TolX', eps);
46
47 % jedes Kandidatenpaar ist Startpunkt für die Simplexsuche
48 for j = 1:size(S, 1)
49
50     disp(sprintf('Kandidat %g: alpha = %g, beta = %g', j, S(j, 1), S(
51                 j, 2)));
52
53     [x,fval,exitflag,output] = fminsearch(@(x) hpy(x, A, C), S(j, :),
54         optionen);
55     if exitflag ~= 1
56         disp('kein Minimum gefunden');
57     end
58     disp(sprintf('Minimum gefunden bei alpha = %g, beta = %g, f = %
59                 g, iterations = %g', x(1), x(2), fval, output.iterations));
60     if absmin > fval
61         absmin = fval;
62     end
63     if rank([A^2 + x(1)*A + x(2)*eye(n); C]) < n
64         disp(sprintf('Rangverlust bei alpha = %g, beta = %g', x(1),
65                 x(2)));
66         alpha = x(1);
67         beta = x(2);
68         ja = 0;
69         return;
70     end
71 end
72
73 ja = 1;

```

```
70 disp(sprintf('gefundener minimaler Singulärwert = %g', absmin));
```

Algorithmus A.2: Zielfunktion des Minimierungsproblems

```
1 function sigma_min = hpy(x, A, C)
2
3 alpha = x(1);
4 beta = x(2);
5 sigma_min = min(svd([A^2 + alpha * A + beta * eye(size(A, 2)); C]));
```

Algorithmus A.3: Routine zur Ermittlung der Kandidatenmenge S

```
1 function S = ermittleKandidaten(eigs)
2 % S = ermittleKandidaten(eigs)
3 % liefert Kandidaten (alpha, beta) für den Verlust des Ranges von  $H_{\{$ 
4 % Parameter:
5 % eigs ... Schätzung für die Eigenwerte
6 % Rückgabewert:
7 % S ... enthält die Kandidaten zeilenweise
8
9 n = length(eigs);
10 S = [];
11
12 % alle Paare von Eigenwerten, deren Summe reell ist
13 for i = 1:n
14     for j = i+1:n
15         kandidat = conv([1 -eigs(i)], [1 -eigs(j)]);
16         if isreal(kandidat)
17             S = [S; kandidat(2:3)];
18         end
19     end
20 end
21
22 % alpha = reeller Eigenwert; beta = 0
23 for i = 1:n
24     kandidat = conv([1 -eigs(i)], [1 0]);
25     if isreal(kandidat)
26         S = [S; kandidat(2:3)];
27     end
28 end
```

A.2 Überprüfung der Eigenvektor-Bedingung

Algorithmus A.4: Routine zur Überprüfung der Eigenvektor-Bedingung

```
1 function ja = eigenvektorTest(A, C)
2 % ja = eigenvektorTest(A, C)
3 % überprüft die Eigenschaft perspektivischer Beobachtbarkeit
```

```

4 % anhand von Rechts-Eigenvektoren von A und deren "Verhältnis" zu C
5 % Parameter:
6 % A ... Systemmatrix
7 % C ... Ausgangsmatrix
8 % Rückgabewert:
9 % ja ... -1 ... keine Entscheidung
10 %      0 ... System ist nicht perspektivisch beobachtbar
11 %      1 ... System ist perspektivisch beobachtbar
12
13 ja = -1;
14 [T, D] = eig(A);
15
16 n = size(A, 2);
17
18 % C*p ~ 0 für alle Eigenvektoren
19 for i = 1:n
20     if ~any(C*T(:, i))
21         disp(sprintf('C*p_%g = 0', i));
22         disp(T(:, i));
23         ja = 0;
24         return;
25     end
26 end
27
28 % teste, ob die Spalten von C*p_i paarweise linear unabhängig sind
29 % aber nur wenn die Summe der beiden Eigenvektoren reell ist
30 test = C*T;
31 for i = 1:n
32     for j = i+1:n
33         disp(sprintf('i = %g, j = %g', i, j));
34         if ~isreal(T(:, i)+T(:, j))
35             disp('nicht reell; weiter');
36             continue;
37         end
38         if rank([test(:, i), test(:, j)]) ~= 2
39             disp(sprintf('C*p_%g || C*p_%g', i, j));
40             disp(test(:, i));
41             disp(test(:, j));
42             ja = 0;
43             return;
44         end
45     end
46 end
47
48 ja = 1;
49 disp('beobachtbar!');

```

A.3 Ermittlung des perspektivischen Beobachtbarkeitsmaßes

Algorithmus A.5: Routine zur Ermittlung des Maßes für die perspektivische Beobachtbarkeit

```

1 function [mu, deltaA, deltaC] = perspektivischesBeobachtbarkeitsMass(
    A, C)
2 % [mu, deltaA, deltaC] = perspektivischesBeobachtbarkeitsMass(A, C)
3 % ermittelt ein Maß für die perspektivische Beobachtbarkeit des
    Systems (A, C)
4 % Parameter:
5 % A ... Systemmatrix
6 % C ... Ausgangsmatrix
7 % Rückgabewerte:
8 % mu ... das Maß
9 % deltaA ... die Störung der Matrix A, damit (A + deltaA, C + deltaC)
    nicht
10 %                perspektivisch beobachtbar ist
11 % deltaC ... die Störung der Matrix C, damit (A + deltaA, C + deltaC)
    nicht
12 %                perspektivisch beobachtbar ist
13
14 % Eigenvektoren ermitteln (schätzen)
15 [T, D] = eig(A);
16
17 n = size(A, 2);
18 mu = inf; deltaA = []; deltaC = [];
19
20 % Minimierungsoptionen
21 options = optimset('Display','off', ...
22                   'TolFun', 1e-10);%, ...
23 flag = 0;
24
25 % für jeden Eigenvektor als Startpunkt
26 for run = 1:n
27
28     disp([sprintf('Start bei ') num2str(T(:, run)')]);
29
30     [x, fval, exitflag, output] = fminsearch(@(x) aussageM(x, A, C, 0),
        T(:, run), options);
31     if (exitflag ~= 1) && (fval == inf)
32         disp(' kein Minimum gefunden');
33         disp(sprintf(' %s', output.message));
34     else
35         if exitflag ~= 1
36             flag = 1;
37         end

```

```

38     if exitflag ~= 1
39         disp([' uneindeutiges Minimum gefunden: x = ' num2str(x
           ') sprintf(', mu = %g', fval)]);
40     else
41         disp([' Minimum gefunden: x = ' num2str(x') sprintf(',
           mu = %g', fval)]);
42     end
43     if fval < mu
44         [mu, deltaA, deltaC] = aussageM(x, A, C, 0);
45     end
46 end
47
48 end
49
50 if (flag == 1) && (mu ~= 0)
51     warning('ACHTUNG: gefundenes Minimum uneindeutig!, evt.
           MaxFunEvals Parameter erhöhen');
52 end
53 if any(imag(deltaA))
54     warning('ACHTUNG: Störungsmatrix deltaA ist komplex!');
55 end
56 if any(imag(deltaC))
57     warning('ACHTUNG: Störungsmatrix deltaC ist komplex!');
58 end

```

Algorithmus A.6: Zielfunktion der Aussage \textcircled{M}

```

1 function [mu, deltaA, deltaC] = aussageM(x, A, C, debug)
2 % [mu, deltaA, deltaC] = aussageM(x, A, C, debug)
3 % ermittelt das Maß für die perspektivische Beobachtbarkeit des
   Systems (A, C)
4 % für einen gegebenen Vektor x
5 % Parameter:
6 % A ... Systemmatrix
7 % C ... Ausgangsmatrix
8 % x ... Vektor x
9 % debug ... 1 ... keine Zwischenergebnisse werden ausgegeben (
   Standard-Vorgabe)
10 %      0 ... keine Zwischenergebnisse werden ausgegeben
11 % Rückgabewerte:
12 % mu ... das Maß
13 % deltaA ... die Störung der Matrix A, damit (A + deltaA, C + deltaC)
   nicht
14 %      perspektivisch beobachtbar ist
15 % deltaC ... die Störung der Matrix C, damit (A + deltaA, C + deltaC)
   nicht
16 %      perspektivisch beobachtbar ist
17
18 % Standard-Parameter

```

```

19 if nargin < 4
20     debug = 1;
21 end
22
23 n = size(A, 2);
24
25 % x normieren
26 x = x/norm(x);
27
28 % deltaA ermitteln
29 deltaA = (-A*x + x*x'*A*x/(x'*x))*x'/(x'*x);
30
31 % finden der EV(A + deltaA)  $\tilde{=} x$ 
32 [T, tmp] = eig(A + deltaA);
33 t = [];
34 for run = 1:n
35     if rank([x T(:, run)]) == 2
36         t = [t T(:, run)];
37     end
38 end
39
40 % wenn x reell: deltaC so, daß (C + deltaC)x null
41 if any(imag(x))
42     kandidat(:, :, 1) = -C*x*x'/(x'*x);
43     norms(1) = norm(kandidat(:, :, 1));
44     if debug
45         disp([ 'x = ' num2str(x') ', normal zu 0' ]);
46         disp(sprintf(' mu = %g', norms(1)));
47     end
48 else
49     if debug
50         disp([ 'x ' num2str(x') 'ist nicht reell' ]);
51     end
52     kandidat(:, :, 1) = inf(size(B));
53     norms(1) = inf;
54 end
55
56 % Minimierungsoptionen
57 options = optimset('Display','off', ...
58                 'TolFun', eps, ...
59                 'TolX', eps);
60
61 % deltaC so, daß (C + deltaC)x, (C + deltaC)xj linear abhängig
62 found = 2; r0 = -1;
63 for run = 1:size(t, 2)
64
65     xj = t(:, run);
66     if debug
67         disp([ 'x = ' num2str(x') ', xj = ' num2str(xj') ]);

```

```

68     end
69
70     if ~any(imag(x + xj))
71         [r, fval, exitflag, output] = fminsearch(@(r) minDeltaC(r, x, xj
72             , C), r0, options);
73         if exitflag ~= 1
74             if debug
75                 disp(' kein Minimum gefunden');
76             end
77         else
78             if debug
79                 disp([sprintf(' Minimum gefunden, r = %g, mu = %g ',
80                     r, fval) 'x = ' num2str(x) ' xj = ' num2str(xj)
81                     ]);
82             end
83             [norms(found), kandidat(:, :, found)] = minDeltaC(r, x,
84                 xj, C);
85             found = found + 1;
86         end
87     else
88         if debug
89             disp(['x + xj ' num2str(x + xj) ' ist nicht reell']);
90         end
91     end
92 end
93
94 [min_C, ind] = min(norms);
95 deltaC = kandidat(:, :, ind);
96 mu = norm(deltaA) + min_C;
97 if debug
98     disp(sprintf('mu = %g, normA = %g, normB = %g', mu, norm(deltaA),
99         min_C));
100 end

```

Algorithmus A.7: Zielfunktion des Minimierungsproblems für δC_j

```

1 function [mu, deltaC] = minDeltaC(r, x, xj, C)
2 % [mu, deltaC] = minDeltaC(r, x, xj, C)
3 % ermittelt die minimale Norm von deltaC für die gegebenen zwei
4   Eigenvektoren und r
5 % Parameter:
6 % r ... skalarer Parameter, damit C*x, C*xj linear abhängig werden
7 % x ... Eigenvektor von A
8 % xj ... anderer Eigenvektor von A
9 % C ... Ausgangsmatrix
10 % Rückgabewerte:
11 % mu ... die Norm von deltaC
12 % deltaC ... die Störung der Matrix C, damit (A + deltaA, C + deltaC)
13   nicht

```

```

12 %           perspektivisch beobachtbar ist
13
14 % x, xj reell
15 if ~any(imag(x)) && ~any(imag(xj))
16     deltaC = -C*(x + r*xj)*(x + r*xj)' / ((x + r*xj)'*(x + r*xj));
17 %x, xj konj. kompl.
18 else
19     disp('konj. komp. ');
20     if rank([C*x C*xj]) == 1
21         deltaC = zeros(size(C)); % C*x, C*xj bereits l.a.
22     else
23         deltaC = -C*(x + r*xj)*(x + r*xj)' / ((x + r*xj)'*(x + r*xj));
24     end
25 end
26
27 mu = norm(deltaC);

```

A.4 Testumgebung

Im Laufe der vorliegenden Arbeit wurden immer wieder Beispiele konstruiert bzw. von existierenden Arbeiten übernommen. Um die entwickelten Routinen zu testen, wurde folgende Testroutine verwendet.

Algorithmus A.8: Routine zur Überprüfung der Eigenvektor-Bedingung \mathbb{S}

```

1 % Testfälle für perspektivische Beobachtbarkeit
2
3 clear all;
4 close all;
5
6 % Struktur: A ... Systemmatrix
7 %           eigs ... Eigenwerte von A
8 %           C ... Ausgangsmatrix
9 %           pB ... 0 für persp.beob.; 1 für nicht persp.beob.;
10 %           Test ... Testergebnis
11
12 % Testprogramm-Test
13 test = struct('A', [-1, 2; 0, -3], ...
14             'eigs', [-1 -3], ...
15             'C', [1 2], ...
16             'pB', 1, ...
17             'Test', -1);
18
19 % Artikel Ghosh, Rosenthal [6], Beispiel 4
20 test(2) = struct('A', [0 1; 0 0], ...
21               'eigs', [0 0], ...
22               'C', [1 0], ...
23               'pB', 1, ...
24               'Test', -1);

```

```

25
26 % Artikel Ghosh, Rosenthal [6], Beispiel 6
27 test(3) = struct('A', [-81, -56, 57, -11;
28                     146, 102, -106, 20;
29                     62, 43, -46, 9;
30                     203, 138, -149, 31], ...
31                 'eigs', [0 1 2 3], ...
32                 'C', [1 0 0 0;
33                     0 1 0 0;
34                     0 0 1 0], ...
35                 'pB', 0, ...
36                 'Test', -1);
37
38 % W.P. Dayawansa, B.K. Ghosh, C. Martin, X.Wang [4]
39 % verletzt die Irrationalitätsbedingung
40 test(4) = struct('A', [ 1, 1, 0, 0;
41                     -1, 1, 0, 0;
42                     0, 0, 2, 2;
43                     0, 0, -2, 2], ...
44                 'eigs', [1+1i, 1-1i, 2+2i, 2-2i], ...
45                 'C', [1, -1, 1, 0;
46                     1, 1, 0, 1], ...
47                 'pB', 0, ...
48                 'Test', -1);
49
50 % Buch DHJ [7], Beispiel 3
51 test(5) = struct('A', [0, -1, 0; 1, 0, 0; 0, 0, 1], ...
52                 'eigs', [+i, -i, 1], ...
53                 'C', [1, 0, 0; 0, 1, 0], ...
54                 'pB', 1, ...
55                 'Test', -1);
56
57 % mein Beispiel zu dem gleichen Thema
58 test(6) = struct('A', [0, -1, 0; 1, 0, 0; 0, 0, 4.5], ...
59                 'eigs', [+i, -i, 4.5], ...
60                 'C', [1, 0, 0; 0, 1, 0], ...
61                 'pB', 1, ...
62                 'Test', -1);
63
64 % Beispiel c_1 parallel zu c_3
65 test(7) = struct('A', diag([-6 -4 -2]), ...
66                 'eigs', [-6 -4 -2], ...
67                 'C', [1 0 1; 0 1 0], ...
68                 'pB', 1, ...
69                 'Test', -1);
70
71 % Beispiel wie test(7), aber transformiert
72 test(8) = struct('A', [-8 -3 -3;
73                     2 -3 3];

```

```

74             2  1 -1], ...
75         'eigs', [-6 -4 -2], ...
76         'C', [1 0 1; 0 1 0]*[1 1 0;
77             0 2 -2;
78             1 0 3], ...
79         'pB', 1, ...
80         'Test', -1);
81
82 % Buch DHJ [7], Beispiel 2
83 % verletzt die Irrationalitätsbedingung
84 test(9) = struct('A', [0 -1 0 0;
85                     1 0 0 0;
86                     0 0 2 -1;
87                     0 0 1 2], ...
88         'eigs', [-i, +i, 2+i, 2-i], ...
89         'C', [1 0 1 0;
90             0 1 0 1], ...
91 ...%         'pB', 1, ...
92         'pB', 0, ...
93         'Test', -1);
94
95 % wie test(9), aber die Minus ein bißchen anders
96 % verletzt die Irrationalitätsbedingung
97 test(10) = struct('A', [ 0 1 0 0;
98                       -1 0 0 0;
99                       0 0 2 1;
100                      0 0 -1 2], ...
101         'eigs', [-i, +i, 2+i, 2-i], ...
102         'C', [1 0 1 0;
103             0 1 0 1], ...
104 ...%         'pB', 1, ...
105         'pB', 0, ...
106         'Test', -1);
107
108 % 2 komplexe Polpaare, zueinander irrational
109 test(11) = struct('A', [ 0 1      0      0;
110                       -1 0      0      0;
111                       0 0      2  sqrt(2);
112                       0 0 -sqrt(2)  2], ...
113         'eigs', [-i, +i, 2+i*sqrt(2), 2-i*sqrt(2)], ...
114         'C', [1 0 1 0;
115             0 1 0 1], ...
116         'pB', 0, ...
117         'Test', -1);
118
119 % 3 reelle Pole
120 test(12) = struct('A', diag([-6 -4 -2]), ...
121         'eigs', [-6 -4 -2], ...
122         'C', [1 2 1; 2 1 1], ...

```

```

123         'pB', 0, ...
124         'Test', -1);
125
126 % Beispiel Dourdoumas, numerische Ungenauigkeit
127 test(13) = struct('A', [-1  0.001  0  0;
128                       -0.001 -1  0  0;
129                       0  0  -0.999  0;
130                       0  0  0  -1.001], ...
131                 'eigs', [-0.999 -1.001 -1+0.001i -1-0.001i], ...
132                 'C', [1 0 2 4; 0 1 5 10], ...
133                 'pB', 1, ...
134                 'Test', -1);
135
136 % Beispiel Dourdoumas, numerische Ungenauigkeit, anderes C
137 test(14) = struct('A', [-1  0.001  0  0;
138                       -0.001 -1  0  0;
139                       0  0  -0.999  0;
140                       0  0  0  -1.001], ...
141                 'eigs', [-0.999 -1.001 -1+0.001i -1-0.001i], ...
142                 'C', [1 0 1 0; 0 1 0 1], ...
143                 'pB', 0, ...
144                 'Test', -1);
145
146 % soll numerischen Fehler provozieren
147 % T = [  -3   7  -23   3;
148 %       5  -12  40  -8;
149 %       2   -5  17   1;
150 %       7  -17  58  23];
151 test(15) = struct('A', [-81, -56, 57, -11;
152                       146, 102, -106, 20;
153                       62, 43, -46, 9;
154                       203, 138, -149, 31], ...
155                 'eigs', [0 1 2 3], ...
156                 'C', [1 1 -1 0;
157                       0 1 1 -1;
158                       3 0 1 1], ...
159                 'pB', 1, ...
160                 'Test', -1);
161
162 % numerischer Fehler beim Berechnen der Eigenwerte
163 test(16) = struct('A', gallery(3), ...
164                 'eigs', [1 2 3], ...
165                 'C', [1435, 475, 1470;
166                       858, 284, 879], ...
167                 'pB', 1, ...
168                 'Test', -1);
169
170 % Landschaft schaut interessant aus
171 test(17) = struct('A', [-2 -43.2 -68.4;

```

```

172             0  -79.4  -118.8;
173             0   39.6   59.2], ...
174         'eigs', [-0.2 -2 -20], ...
175         'C', [0     1     2;
176             1     2     5], ...
177         'pB', 1, ...
178         'Test', -1);
179
180 % muss bei Cp \neq 0 p reell sein?
181 test(18) = struct('A', [0, -1, 0; 1, 0, 0; 0, 0, 1], ...
182                 'eigs', [+i, -i, 1], ...
183                 'C', [0, 1, 0; 0, 0, 1], ...
184                 'pB', 1, ...
185                 'Test', -1);
186
187 % muss bei Cp \neq 0 p reell sein?
188 test(19) = struct('A', [0, -1, 0; 1, 0, 0; 0, 0, 1], ...
189                 'eigs', [+i, -i, 1], ...
190                 'C', [1, 1, 0; 1, 0, 1], ...
191                 'pB', 0, ...
192                 'Test', -1);
193
194 % Cpj | Cpi mit pi, pj konj. kompl.
195 test(20) = struct('A', [0, -1, 0; 1, 0, 0; 0, 0, 1], ...
196                 'eigs', [+i, -i, 1], ...
197                 'C', [1, 2, 0; 0, 0, 1], ...
198                 'pB', 1, ...
199                 'Test', -1);
200
201
202 % Vorlage
203 % test(i+1) = struct('A', [], ...
204 %                 'eigs', [], ...
205 %                 'C', [], ...
206 %                 'pB', 0, ...
207 %                 'Test', -1);
208
209 korrekt = 0;
210 inkorrekt = 0;
211 sum = length(test);
212
213 for i = 1 : sum
214
215     disp(sprintf('TEST %d:', i));
216
217     % getestete Funktion
218     test(i).Test = ~eigenvektorTest(test(i).A, test(i).C);
219     % [test(i).Test, alpha, beta] =
220         pruefePerspektivischeBeobachtbarkeit(test(i).A, test(i).C);

```

```
    test(i).Test = ~test(i).Test;
220 % [mu, deltaA, deltaC] = perspektivischesBeobachtbarkeitsMass(A,
    C); mu
221
222 if test(i).Test == test(i).pB
223     disp(sprintf('Test %d korrekt', i));
224     korrekt = korrekt + 1;
225 else
226     disp(sprintf('Test %d NICHT korrekt', i));
227     disp(sprintf('Test ist %d, richtig wäre %d', test(i).Test,
    test(i).pB));
228     inkorrekt = inkorrekt + 1;
229 end
230
231 if test(i).pB == 0
232     disp('System ist persp. beobachtbar');
233 else
234     disp('System ist NICHT persp. beobachtbar');
235 end
236
237 end
238
239 disp(sprintf('%d korrekt, %d inkorrekt, in Summe %d Tests', korrekt,
    inkorrekt, sum));
```

Literaturverzeichnis

- [1] M. Horn, N. Dourdoumas: Regelungstechnik. Pearson Studium, München, 2004, S. 11-84.
- [2] B.K. Ghosh, E.P. Loucks: An Introduction to Perspective Observability and Recursive Identification Problems in Machine Vision. Proceedings of the 33rd IEEE Conference on Decision and Control, Vol. 4, Dec 1994, S. 3229-3234.
- [3] B.K. Ghosh, M. Jankovic, Y.T. Wu: Some Problems in Perspective System Theory and its Application to Machine Vision. Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems, July 1992, S.139-146.
- [4] W.P. Dayawansa, B.K. Ghosh, C. Martin, X. Wang: A necessary and sufficient condition for the perspective observability problem. Systems & Control Letters, Vol. 25, 1995, S. 159-166.
- [5] B.K. Ghosh, C.F. Martin: Homogeneous Dynamical Systems Theory. IEEE Transactions on Automatic Control, Vol. 47, No. 3, March 2002, S. 462-472.
- [6] B.K. Ghosh, J. Rosenthal: A Generalized Popov-Belevitch-Hautus Test of Observability. IEEE Transactions on Automatic Control, Vol. 40, No. 1, Jan 1995, S. 176-180.
- [7] G. Dirr, U. Helmke, J. Jordan: Control and Observation of the Matrix Riccati Differential Equation. Kapitel aus: B.K. Ghosh: Emergent problems in nonlinear systems and control. Springer, Berlin, 2009, aus der Reihe Lecture Notes in Control and Information Sciences.
- [8] C.C. Paige: Properties of Numerical Algorithms Related to Computing Controllability. IEEE Transactions on Automatic Control, Vol. 26, No. 1, Feb. 1981, S. 130-138.
- [9] R. Eising: Between Controllable and Uncontrollable. Systems & Control Letters, Vol. 4, No. 5, 1984, S. 263-264.
- [10] R. Eising: The distance between a system and the set of uncontrollable systems. Kapitel aus: P.A.Fuhrmann: Mathematical Theory of Networks and Systems. Proceedings of the MTNS-83 International Symposium Beer Sheva, Springer, Berlin, 1984, aus der Reihe Lecture Notes in Control and Information Sciences.

-
- [11] C.B. Moler: Numerical computing with MATLAB . Society for Industrial and Applied Mathematics, Philadelphia, 2004, S. 269ff.
 - [12] R. Eising: Between Controllable and Uncontrollable. Systems & Control Letters, Vol. 4, No. 5, 1984, S. 263-264.
 - [13] R. Byers: Detecting nearly uncontrollable Pairs. Signal Processing, Scattering and Operator Theory, and Numerical Methods Proceedings of the International Symposium MTNS-89, Amsterdam, Vol. 3, S. 1990, 447-457.
 - [14] M. Gao, M. Neumann: A Global Minimum Search Algorithm for Estimating the Distance to Uncontrollability. Linear Algebra and its Applications 188, 189; 1993; S. 305-350.
 - [15] M. Gu: New Methods for Estimating the Distance to Uncontrollability. SIAM Journal on Matrix Analysis and Applications, Vol. 21, No. 3, 2000, S. 989-1003.
 - [16] J.H. Holland: Adaptation in Natural and Artificial Systems. MIT Press, Cambridge, 1992, S. 89ff.
 - [17] J.C. Lagarias, J. A. Reeds, M. H. Wright, P. E. Wright: Convergence Properties of the Nelder-Mead Simplex Method in Low Dimensions. SIAM Journal of Optimization, Vol. 9, No. 1, 1998, S. 112-147.
 - [18] R. Fletcher: Practical Methods of Optimization. John Wiley & Sons, Chichester, 1987, S. 16ff.
 - [19] C.R. Rao, S.K. Mitra: Generalized Inverse of Matrices and its Applications. John Wiley & Sons, New York, 1971, S. 44ff.