

## **Aerobotics-Seminar Moonshot-Aufgabe**

# **Systemidentifikation: Schätzung der Parameter flugmechanischer Modelle aus Flugmessdaten**

**Autoren: Gruppe 02**  
**Calvin Ebert**  
**Adam Ghribi**  
**Florian Gschwandtner**  
**Fabrizio Turco**

**Datum: 06.08.2021**

# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>2</b>
<b>2. Modell</b>	<b>3</b>
<b>3. Vorbereitung der Daten</b>	<b>4</b>
3.1. Ermittlung der relevanten Signale . . . . .	4
3.2. Trimpunkt . . . . .	5
3.3. Interpolation . . . . .	6
3.4. Filterung . . . . .	6
3.4.1. Ablauf . . . . .	7
3.4.2. Wahl der Filterübertragungsfunktion . . . . .	7
3.5. Simulator . . . . .	8
<b>4. Systemidentifikation im Zeitbereich</b>	<b>9</b>
4.1. <i>Least Squares</i> -Methode . . . . .	9
4.1.1. Schätzung der Parameter mit dem LSQ-Verfahren . . . . .	10
4.2. Matrix-LSQ-Methode . . . . .	10
4.2.1. Modellgleichung . . . . .	10
4.2.2. Schätzgleichung . . . . .	11
4.3. Ergebnisse . . . . .	11
4.4. Interpretation . . . . .	12
<b>5. Systemidentifikation im Frequenzbereich</b>	<b>14</b>
5.1. Fourier-Transformation . . . . .	14
5.2. <i>Output Error</i> -Methode . . . . .	14
5.3. Vor- und Nachteile . . . . .	16
5.4. Ergebnisse . . . . .	17
<b>6. Zusammenfassung und Ausblick</b>	<b>19</b>
<b>Literaturverzeichnis</b>	<b>19</b>
<b>A. Test des Algorithmus im Frequenzbereich</b>	<b>21</b>

# 1. Einleitung

Die vorliegende Arbeit wurde im Rahmen des Aerobotics-Seminars im Sommersemester 2021 an der Universität Stuttgart verfasst. Ziel des Projekts war die Identifikation der Parameter eines linearisierten Modells der Längsbewegung aus vorliegenden Flugmessdaten eines 1:3-Modells des Flugzeugs *e-Genius*. Für die Seitenbewegung wurde ebenfalls eine Systemidentifikation durchgeführt, aus Gründen der Übersichtlichkeit und Kürze wird diese aber in der Ausarbeitung ausgelassen.

Es wurden Verfahren im Zeit- und im Frequenzbereich verwendet, um die Ergebnisse abschließend vergleichen zu können. Genauer handelt es sich dabei im Zeitbereich um die *Least Squares*-Methode sowie die Matrix-LSQ-Methode. Im Frequenzbereich wurde die *Output Error*-Methode in Verbindung mit einem *Newton-Raphson*-Algorithmus verwendet.

Linearisierte Modelle spielen eine wichtige Rolle in der Regelungstechnik: Mit ihnen lässt sich eine lineare Regelung um einen stationären Arbeits- bzw. Trimpunkt entwickeln. Eine Schwierigkeit besteht dabei aber immer wieder darin, die Regelstrecke mit dem Modell ausreichend genau zu beschreiben. Hier kommt die Systemidentifikation ins Spiel: Sie hat zum Ziel, ausgehend von gegebenen Flugmessdaten die Parameter des definierten Modells bestmöglich abzuschätzen.

## 2. Modell

Im folgenden Abschnitt wird das der durchgeführten Systemidentifikationen zugrundeliegende Modell beschrieben, dessen Beiwerte zu bestimmen sind. Es handelt sich dabei um das bekannte linearisierte Modell der Längsbewegung mit den folgenden Annahmen [2]:

- Linearisierung um den symmetrischen Horizontalflug ( $\gamma_0 = 0$ )
- kein Auftrieb durch Nickrate ( $Z_q = 0$ )
- keine Querkräfte durch Roll- oder Gierdrehrate ( $Y_p = Y_r = 0$ )
- keine Querkraft durch Querruder ( $Y_\xi$ )
- kein Wind ( $\Delta\gamma = \Delta\theta - \Delta\alpha$ )
- horizontal eingebautes Triebwerk ( $i_F = 0$ )

Die Dynamiken können deshalb entkoppelt behandelt werden, was eine getrennte Identifikation der Längs- bzw. Seitenbewegung ermöglicht.

Der Zustand  $\mathbf{x}$  der Längsbewegung setzt sich zusammen aus dem Anstellwinkel  $\alpha$ , der Nickrate  $q$ , der Anströmgeschwindigkeit  $V_A$  und dem Bahnwinkel  $\gamma$ . Die zugehörige Steuerung  $\mathbf{u}$  umfasst den Höhenruderausschlag  $\eta$  und den Schubdrosselgrad  $\delta_F$ . Bis auf die Nickrate werden alle Größen als Abweichungen (Delta-Größen) vom jeweiligen Trimpunkt (gekennzeichnet durch den Index "0") beschrieben. Es ergibt sich folgendes Modell:

$$\underbrace{\begin{pmatrix} \Delta\dot{\alpha} \\ \dot{q} \\ \Delta\dot{V}_A \\ \Delta\dot{\gamma} \end{pmatrix}}_{\dot{\mathbf{x}}} = \underbrace{\begin{pmatrix} \frac{Z_\alpha}{V_0} & 1 & \frac{Z_V}{V_0} & 0 \\ M_\alpha & M_q & M_V & 0 \\ X_\alpha & 0 & X_V & -g \\ -\frac{Z_\alpha}{V_0} & 0 & -\frac{Z_V}{V_0} & 0 \end{pmatrix}}_{\mathbf{A}} \cdot \underbrace{\begin{pmatrix} \Delta\alpha \\ q \\ \Delta V_A \\ \Delta\gamma \end{pmatrix}}_{\mathbf{x}} + \underbrace{\begin{pmatrix} \frac{Z_\eta}{V_0} & -\frac{X_{\delta F}}{V_0} \sin(\alpha_0) \\ M_\eta & M_{\delta F} \\ X_\eta & X_{\delta F} \cos(\alpha_0) \\ -\frac{Z_\eta}{V_0} & \frac{X_{\delta F}}{V_0} \sin(\alpha_0) \end{pmatrix}}_{\mathbf{B}} \cdot \underbrace{\begin{pmatrix} \Delta\eta \\ \Delta\delta_F \end{pmatrix}}_{\mathbf{u}} \quad (2.1)$$

## 3. Vorbereitung der Daten

Aus den Flugversuchen des e-Genius 1:3 liegt eine Fülle von Messdaten in verschiedenen .csv-Dateien vor. Im folgenden Kapitel wird erklärt, wie diese vorbereitet werden, um sie für die anschließende Systemidentifikation zu verwenden.

### 3.1. Ermittlung der relevanten Signale

Für die weitere Verarbeitung ist es zunächst nötig, aus den gegebenen Messdaten die relevanten Signalverläufe auszuwählen bzw. zu berechnen. Die meisten Zustandsgrößen können direkt aus den Messdaten verwendet werden, einzig der Bahnwinkel  $\gamma$  muss explizit berechnet werden. Die Bestimmung über die Beziehung  $\gamma = \theta - \alpha$  liefert dabei aufgrund eines unplausiblen Verlaufs des Nickwinkels  $\theta$  kein sinnvolles Ergebnis.<sup>1</sup> Stattdessen wird der Bahnwinkel mit Gleichung (3.1) über die geodätischen Geschwindigkeiten in  $z$ - und  $x$ -Richtung berechnet.

$$\gamma = \arcsin\left(\frac{V_z}{V_x}\right) \quad (3.1)$$

Tabelle 3.1 zeigt eine Übersicht über die verwendeten Messsignale aus den Originaldateien. Zu beachten ist, dass in den Originaldaten für das Höhenruder jeweils ein eigenes Signal für das linke sowie das rechte Ruder vorliegt, im Falle es Querruders sogar zwei je Seite. Im betrachteten Bereich sind diese Signale jedoch zu jedem Zeitpunkt gleich, weshalb die Verwendung eines Verlaufs je Stellruder ausreicht.

**Tabelle 3.1.** – Übersicht über die verwendeten Messgrößen

Längsbewegung			Seitenbewegung		
Größe	Messsignal	Datei	Größe	Messsignal	Datei
$\alpha$	Alpha	vectoflow_airstream	$r$	yawspeed	vehicle_attitude
$q$	pitchspeed	vehicle_attitude	$\beta$	Beta	vectoflow_airstream
$V_A$	VMag	vectoflow_airstream	$p$	rollspeed	vehicle_attitude
$\gamma$	vx, vz	vehicle_local_position	$\phi$	Phi	vectoflow_airstream
$\eta$	elevator_l	actuator_controls	$\xi$	aileron_inner_l	actuator_controls
$\delta_F$	thrust	actuator_controls	$\zeta$	rudder	actuator_controls

Die vorliegenden Messdaten umfassen einen großen Zeitbereich von Start bis Landung. Für die Systemidentifikation wurde nur der Abschnitt berücksichtigt, in dem das Flugzeug Platzrunden fliegt. Dies entspricht im originalen Datensatz in etwa der Zeit zwischen 910 s und 2225 s. Abb. 3.1 zeigt den Flug und den entsprechenden Ausschnitt.

<sup>1</sup>Berechnet man den Bahnwinkel auf diese Weise, nimmt er genau wie der Nickwinkel nie Werte kleiner 0 an. Lediglich der Anstellwinkel bewegt sich sowohl im positiven als auch im negativen Bereich.

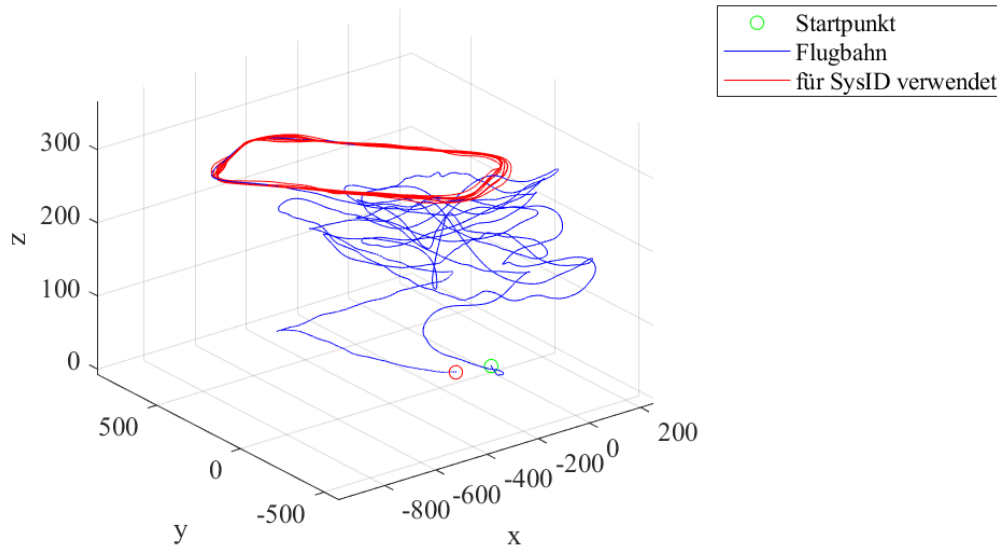


Abbildung 3.1. – Visualisierung der Flugbahn

## 3.2. Trimpunkt

In Abb. 3.2 ist beispielhaft der zeitliche Verlauf der Anströmgeschwindigkeit dargestellt. Es zeigen sich insgesamt acht stationären Bereiche, die als Trimpunkt für eine Modellierung dienen könnten. In den nachfolgenden Systemidentifikationen soll jeweils der erste Punkt (TP 1) als Grundlage dienen. Dazu werden alle Zustands- und Steuergrößen über den Bereich TP 1 gemittelt und diese Mittelwerte als Trimmwerte  $x_0$  und  $u_0$  verwendet. Damit lassen sich die Abweichungen vom Trimpunkt berechnen:

$$\begin{aligned}\Delta x(t) &= x(t) - x_0 \\ \Delta u(t) &= u(t) - u_0\end{aligned}\tag{3.2}$$

Eine Ausnahme bilden hier die Drehraten  $p$ ,  $q$  und  $r$ , bei denen keine Differenz zum Trimmwert gebildet werden muss.

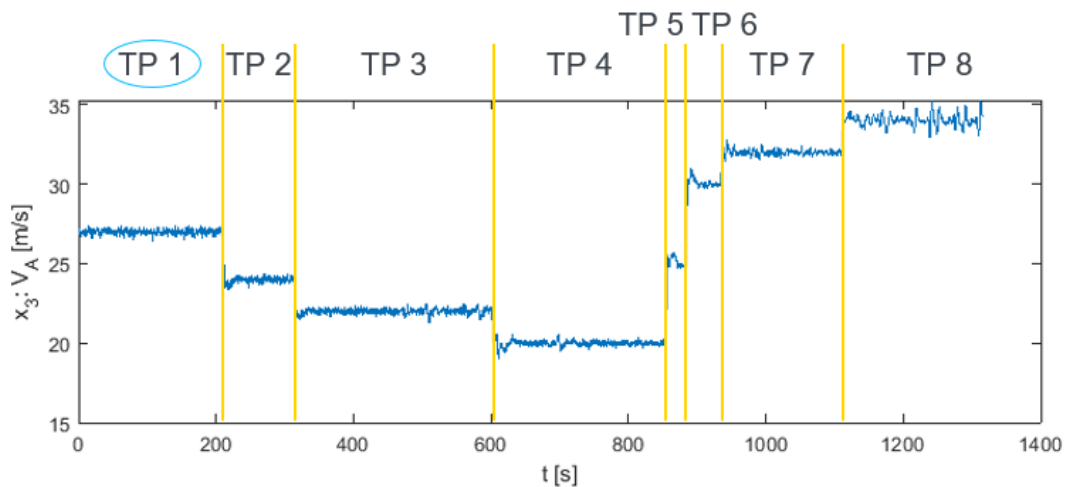


Abbildung 3.2. – zeitlicher Verlauf der Anströmgeschwindigkeit mit den stationären Bereichen

### 3.3. Interpolation

Die Rohdaten werden von mehreren Sensoren mit unterschiedlichen Abtastraten geliefert. Da die Identifikationsalgorithmen die Werte zu diskreten Zeitpunkten benötigen ist es notwendig einen einheitlichen Zeitvektor mit zugehörigen Eingangs- und Zustandsvektoren zu generieren. Außerdem wird eine konstante Schrittweite gefordert.

Die Abtastrate dieses Zeitvektors ist wichtig, da die Dimension des Optimierungsproblems und damit der Rechenaufwand mit feinerer Diskretisierung steigt. In dieser Arbeit wird der Zeitvektor des *vectoflow.airdata* Sensors verwendet, da dieser die geringste Abtastrate hat.

In Abb. 3.3 ist deutlich zu sehen, dass die wesentliche Form des Signals erhalten bleibt. Die Frequenzen beider Sensoren sind so hoch, dass auch im interpolierten Verlauf alle relevanten Signalanteile erhalten bleiben und nur Rauschen verloren geht.

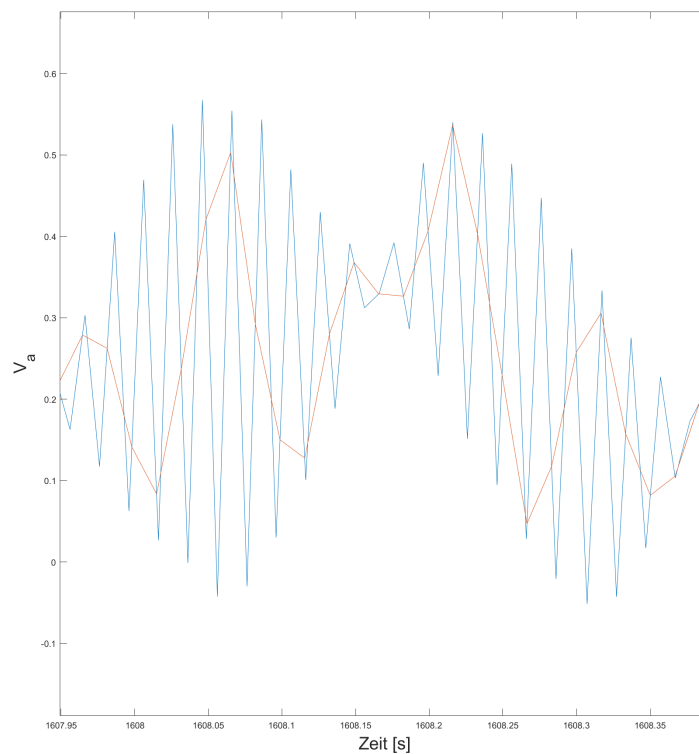


Abbildung 3.3. – Beispiel Interpolation

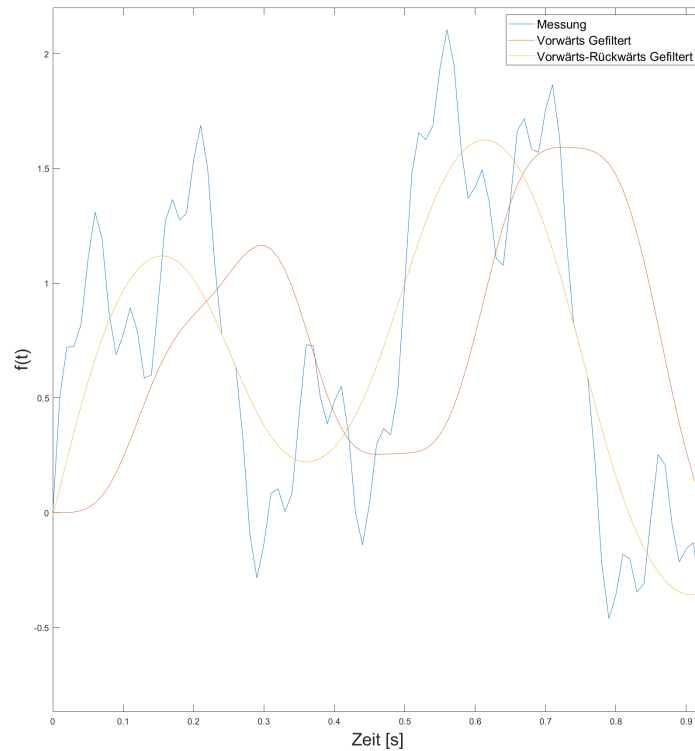
### 3.4. Filterung

Verrauschte Messdaten stellen für die Systemidentifikation eine Herausforderung dar. Numerische Ableitungen aus verrauschten Daten liefern in vielen Fällen keine sinnvolle Aussage. Neben aufwändigeren Ableitungsregeln bietet sich eine vorangehende Filterung der Daten an.

Das Vorwärts-Rückwärtsfilter bietet den Vorteil, dass keine Phasenverschiebung auftritt. Gerade wenn nur einzelne Signale gefiltert werden, beispielsweise nur der Eingang, ist diese Eigenschaft unerlässlich. Der Nachteil ist, dass das Filter nicht in Echtzeit verwendet werden kann, da immer die vollständige Datenreihe vorliegen muss. Für eine Systemidentifikation ist

dies jedoch keine praktische Einschränkung. [4]

In Abb. 3.4 sind die Auswirkungen einer reinen Vorwärts- und einer Vorwärts-Rückwärtsfilterung auf die Phase gut zu sehen.



**Abbildung 3.4.** – Beispiel Filterung

### 3.4.1. Ablauf

Für das Filter wird eine Übertragungsfunktion  $f(s)$  auf die Messdaten vorwärts angewandt, die Messdaten umgekehrt und die selbe Übertragungsfunktion noch einmal verwendet. In Matlab ist diese Vorgehensweise in der Funktion `filtfilt()` bereits implementiert.

### 3.4.2. Wahl der Filterübertragungsfunktion

Es wurde ein quadriertes PT2-Glied gewählt, da so die Eckfrequenz direkt eingestellt werden kann. Die Eckfrequenz wurde zu 20 Hz gewählt, damit das Rauschen unterdrückt wird, aber keine Information verloren geht.

Mit

$$\omega_{filt} = 2 \cdot \pi \cdot f_{eck} = 2 \cdot \pi \cdot 20 \text{ Hz} \quad (3.3)$$

und der Dämpfung

$$\zeta_{filt} = \frac{1}{\sqrt{2}} \quad (3.4)$$



ergibt sich die Übertragungsfunktion zu:

$$f(s) = \left( \frac{\omega_{filt}^2}{s^2 + 2 \cdot \zeta_{filt} \cdot \omega_{filt} s + \omega_{filt}^2} \right)^2 \quad (3.5)$$

Im Bodediagramm, Abb. 3.5, sind Amplituden- und Frequenzgang der Filterfunktion zu sehen.

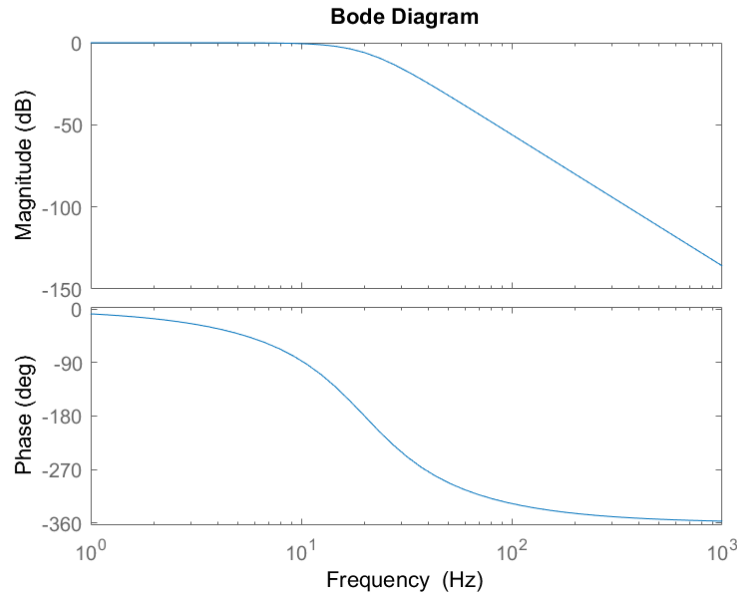


Abbildung 3.5. – Bodediagramm der Filterfunktion aus Gleichung (3.5)

### 3.5. Simulator

Zur Beurteilung und Überprüfung des Modells wird ein einfacher Simulator im Zustandsraum verwendet. Die System- und Eingangsmatrix werden aus der Schätzung übernommen. Weiterhin gilt  $\mathbf{y} = \mathbf{x}$  und  $\mathbf{D} = 0$ .

Als Eingang  $\mathbf{u}$  werden die gefilterten Messdaten der Steuerkommandos verwendet. Der Zeitvektor für die Simulation ergibt sich aus dem langsamst gemessenen Zustand, analog zu Abschnitt 3.3. Das Ergebnis wird zur Weiterverarbeitung gespeichert. Die Simulink-Implementierung ist in Abb. 3.6 dargestellt.

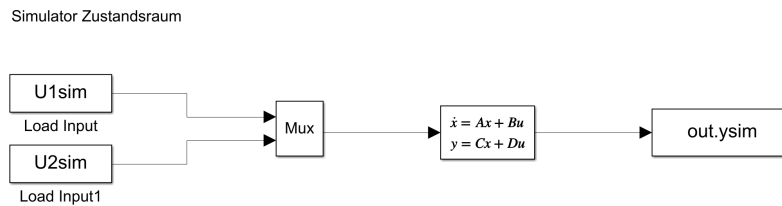


Abbildung 3.6. – Simulator in Simulink

## 4. Systemidentifikation im Zeitbereich

Systemidentifikation im Zeitbereich heißt, ein Modell und den zugehörigen Parametersatz zu finden, welches den zeitlichen Verlauf möglichst gut approximiert. Grundlegend ist hier zwischen der Systemidentifikation und der Parameteridentifikation zu unterscheiden. Systemidentifikation bedeutet, dass auch das Modell als unbekannt angenommen wird und die gesamte Struktur aus den Messdaten geschätzt wird, wie hier mit der Matrix-LSQ. Bei der Parameteridentifikation hingegen ist ein flugmechanisches Modell bekannt, in dem Parameter gesucht werden (vgl. LSQ).

### 4.1. Least Squares-Methode

In diesem Kapitel wird die Methode der kleinsten Fehlerquadrate (*Least Squares*, LSQ) vorgestellt. Eine der Kernaufgaben vieler Ingenieure besteht darin, einen belastbaren Zusammenhang zwischen gemessenen Werten zu finden. Diese Zusammenhänge können näherungsweise durch folgendes lineares Modell beschrieben werden:

$$\mathbf{z} = \mathbf{H} \cdot \mathbf{x} + \mathbf{v} \quad (4.1)$$

mit

$$\text{Messvektor: } \dim(\mathbf{z}) = m \times 1$$

$$\text{bekannte Matrix: } \dim(\mathbf{H}) = m \times n$$

$$\text{unbekannter Rauschterm: } \dim(\mathbf{v}) = m \times 1$$

$$\text{unbekannter Parametervektor: } \dim(\mathbf{x}) = n \times 1$$

Die LSQ-Methode kann beispielsweise verwendet werden, um die Schätzung von Modellparametern im Rahmen der Systemidentifikation durchzuführen. Gesucht ist dabei ein Schätzwert  $\hat{\mathbf{x}}$  für den Parametervektor. Der Restfehlervektor  $\mathbf{e}$  lautet:

$$\mathbf{e} = \mathbf{z} - \mathbf{H} \cdot \hat{\mathbf{x}} \quad (4.2)$$

Die Idee besteht darin, einen Wert für  $\hat{\mathbf{x}}$  zu finden, der die Norm des quadrierten Restfehlervektors minimiert. Das quadratische Zielfunktional  $J$  ist gegeben durch:

$$\begin{aligned} J &= \frac{1}{2} \cdot \mathbf{e}^T \cdot \mathbf{e} \\ &= \frac{1}{2} \cdot (\mathbf{z} - \mathbf{H}\hat{\mathbf{x}})^T \cdot (\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}) \\ &= \frac{1}{2} \cdot (\mathbf{z}^T - \hat{\mathbf{x}}^T \mathbf{H}^T) \cdot (\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}) \\ &= \frac{1}{2} \cdot \mathbf{z}^T \mathbf{z} - \mathbf{z}^T \mathbf{H} \hat{\mathbf{x}} + \frac{1}{2} \cdot \hat{\mathbf{x}}^T \mathbf{A} \hat{\mathbf{x}} \end{aligned} \quad (4.3)$$

mit  $\mathbf{A} = \mathbf{H}^T \mathbf{H}$ . Die Lösung des Minimierungsproblems lautet dann:

$$\hat{\mathbf{x}} = (\mathbf{H}^T \mathbf{H})^{-1} \cdot \mathbf{H}^T \mathbf{z} \quad (4.4)$$

#### 4.1.1. Schätzung der Parameter mit dem LSQ-Verfahren

Die Zustandsraumdarstellung (2.1) liefert folgende Gleichungen:

$$\Delta\dot{\alpha} - q = \frac{Z_\alpha \Delta\alpha}{V_0} + \frac{Z_\nu \Delta V_A}{V_0} + \frac{Z_\eta \Delta\eta}{V_0} - \frac{X_{\delta_F} \sin(\alpha_0) \cdot \Delta\delta_F}{V_0} \quad (4.5)$$

$$\dot{q} = M_\alpha \Delta\alpha + M_q q + M_\nu \Delta V_A + M_\eta \Delta\eta + M_{\delta_F} \Delta\delta_F \quad (4.6)$$

$$\Delta\dot{V}_A = X_\alpha \Delta\alpha + X_\nu \Delta V_A + X_\eta \Delta\eta + X_{\delta_F} \cos(\alpha_0) \cdot \Delta\delta_F \quad (4.7)$$

$$\Delta\dot{\gamma} = \frac{-Z_\alpha \Delta\alpha}{V_0} + \frac{-Z_\nu \Delta V_A}{V_0} + \frac{-Z_\eta \Delta\eta}{V_0} + \frac{X_{\delta_F} \sin(\alpha_0) \cdot \Delta\delta_F}{V_0} \quad (4.8)$$

Diese können zu einem linearen Gleichungssystem umgeformt werden:

$$\mathbf{z}_L = \mathbf{H}_L \cdot \mathbf{x} + \mathbf{v} \quad (4.9)$$

Die einzelnen Vektoren und Matrizen lauten:

$$\mathbf{z}_L = (\Delta\dot{\alpha} - q \quad \dot{q} \quad \Delta\dot{V}_A)^T \quad (4.10)$$

$$\mathbf{H}_L = \begin{pmatrix} 0 & 0 & 0 & \frac{-\sin(\alpha_0) \cdot \Delta\delta_F}{V_0} & \frac{\Delta\alpha}{V_0} & \frac{\Delta V_A}{V_0} & \frac{\Delta\eta}{V_0} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \Delta\alpha & q & \Delta V_A & \Delta\eta & \Delta\delta_F \\ \Delta\alpha & \Delta V_A & \Delta\eta & \cos(\alpha_0) \cdot \Delta\delta_F & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (4.11)$$

$$\mathbf{x} = (X_\alpha \ X_\nu \ X_\eta \ X_{\delta_F} \ Z_\alpha \ Z_\nu \ Z_\eta \ M_\alpha \ M_q \ M_\nu \ M_\eta \ M_{\delta_F})^T \quad (4.12)$$

Wie schon in dem letzten Abschnitt erwähnt, wird es nach einem geschätzten Parametervektor  $\hat{\mathbf{x}}$  gesucht, der die Norm des quadrierten Restfehlervektors  $\mathbf{e} = \mathbf{z} - \mathbf{H} \cdot \hat{\mathbf{x}}$  minimiert. Die Lösung ist laut Gleichung (4.4)

$$\hat{\mathbf{x}} = (\mathbf{H}_L^T \mathbf{H}_L)^{-1} \cdot \mathbf{H}_L^T \mathbf{z}_L \quad (4.13)$$

## 4.2. Matrix-LSQ-Methode

Bei der Matrix-LSQ-Methode werden, im Gegensatz zur klassischen LSQ-Schätzung, sowohl die Systemmatrix  $\mathbf{A}$  als auch die Eingangsmatrix  $\mathbf{B}$  als unbekannt angesehen. Es handelt sich somit um eine Systemidentifikation im eigentlichen Sinn, da keine flugmechanischen Modelle verwendet werden.

### 4.2.1. Modellgleichung

Die Messdaten der Eingänge  $\mathbf{u}$ , des Zustands  $\mathbf{x}$  und der Zustandsableitung  $\dot{\mathbf{x}}$  werden in die Form von Gleichung (4.14) gebracht.

$$\underbrace{\begin{pmatrix} \dot{x}_1^T \\ \vdots \\ \dot{x}_n^T \end{pmatrix}}_{\mathbf{z}} = \underbrace{\begin{pmatrix} x_1^T & u_1^T \\ \vdots & \vdots \\ x_n^T & u_n^T \end{pmatrix}}_{\mathbf{H}} \underbrace{\begin{pmatrix} \mathbf{A}^T \\ \mathbf{B}^T \end{pmatrix}}_{\hat{\mathbf{x}}} + \underbrace{\begin{pmatrix} v_1^T \\ \vdots \\ v_n^T \end{pmatrix}}_{\mathbf{v}} \quad (4.14)$$

- $\mathbf{z}$  Messmatrix
- $\mathbf{H}$  Modellmatrix
- $\hat{\mathbf{x}}$  Schätzwertmatrix
- $\mathbf{v}$  Störgrößenmatrix
- $n$  Anzahl der Zeitschritte

Die Dimension der Systemmatrix und der Eingangsmatrix werden von den Dimensionen von  $\mathbf{u}$ ,  $\mathbf{x}$  und  $\dot{\mathbf{x}}$  vorgegeben. Hier liegt auch der Nachteil der Matrix-LSQ-Methode. Es ist nicht möglich die einzelnen Parameter innerhalb der Matrizen zu beeinflussen oder Wertebereiche vorzugeben.

#### 4.2.2. Schätzgleichung

Nach der Herleitung in [6] ergibt sich die Schätzgleichung zu:

$$\hat{\mathbf{x}} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{z}$$

Besonders hervorzuheben ist hier die Inversion der Matrix  $\mathbf{H}^T \mathbf{H}$ . Es handelt sich dabei um eine Matrix der Dimension  $(\dim(\mathbf{x}) + \dim(\mathbf{u})) \times (\dim(\mathbf{x}) + \dim(\mathbf{u}))$ . In dieser Arbeit bedeutet das konkret eine Dimension  $6 \times 6$ , womit der Rechenaufwand für die - für große Matrizen aufwändige - Inversion klein bleibt.

#### 4.3. Ergebnisse

Anschließend werden die Ergebnisse in diesem Kapitel kurz vorgestellt.

Nach der Schätzung der Einträge der Matrizen  $\mathbf{A}$  und  $\mathbf{B}$  wird das Anfangswertproblem (2.1) im gesamten Zeitintervall gelöst. Bei allen Simulationen sind die Filterparametern wie in Abschnitt 3.4 gewählt.

In Abb. 4.1 wird die Lösung anhand des Matrix-LSQ-Verfahrens dargestellt. Die geschätzten Matrizen  $\mathbf{A}_L$  und  $\mathbf{B}_L$  lauten:

$$\mathbf{A}_L = \begin{pmatrix} -0.1940 & 0.005 & -0.0002 & -0.0540 \\ -38.9899 & -12.8888 & 0.3632 & -4.2686 \\ -6.1506 & -0.1501 & -0.0589 & -5.0107 \\ 0.4433 & 0.0427 & 0.0023 & 0.0818 \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} \frac{Z_\alpha}{V_0} & 1 & \frac{Z_V}{V_0} & 0 \\ M_\alpha & M_q & M_V & 0 \\ X_\alpha & 0 & X_V & -g \\ -\frac{Z_\alpha}{V_0} & 0 & -\frac{Z_V}{V_0} & 0 \end{pmatrix}$$

$$\mathbf{B}_L = \begin{pmatrix} -0.0267 & -0.0164 \\ -26.1377 & -12.9651 \\ -0.7414 & 4.6092 \\ 0.0477 & -0.1328 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} \frac{Z_\eta}{V_0} & -\frac{X_{\delta F}}{V_0} \sin(\alpha_0) \\ M_\eta & M_{\delta F} \\ X_\eta & X_{\delta F} \cos(\alpha_0) \\ -\frac{Z_\eta}{V_0} & \frac{X_{\delta F}}{V_0} \sin(\alpha_0) \end{pmatrix}$$

Die Ergebnisse in Abb. 4.1 zeigen eine relativ präzise Approximation, das Modell liefert vor allem für Anstellwinkel und Geschwindigkeit gute Ergebnisse. Auch die Ergebnisse für  $q$  und den Bahnwinkel sind sinnvoll, durch die wenig aussagekräftigen Messdaten aber schwer zu vergleichen. Allerdings weist das Ergebnis in flugmechanischer Hinsicht Abweichungen in System- und Eingangsmatrix auf. Die Beträge der Einträge der ersten und vierten Zeile von  $\mathbf{A}_L$  unterscheiden sich stark voneinander (0.1940 und 0.4433) und (0.0002 und 0.0023), obwohl

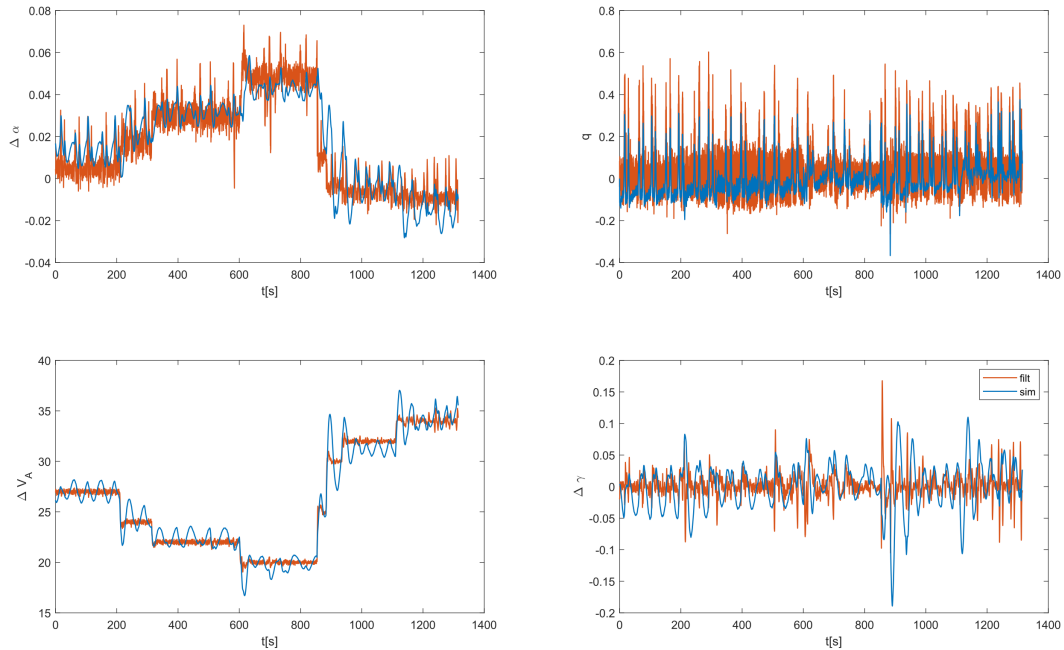


Abbildung 4.1. – Ergebnisse anhand des Matrix-LSQ-Verfahrens

diese beim ersten, dritten und vierten Eintrag aufgrund  $\Delta\gamma = \Delta\theta - \Delta\alpha$  gleich sein sollten. Eine starke Abweichung ist auch zwischen den vierten Spalten der Matrizen  $A_L$  und  $A$  zu sehen.

Abb. 4.2 zeigt die Lösung anhand des LSQ-Verfahrens. Die Methode liefert kein sinnvolles Ergebnisse. Die Simulation ist nicht stabil.

## 4.4. Interpretation

Die wesentliche Erkenntnis der Analyse im Zeitbereich ist, dass es mit der LSQ-Methode nicht möglich ist, die Parameter des linearen Modells für die Längsbewegung so zu bestimmen, dass eine stabile Simulation möglich ist. Mit der Matrix-LSQ-Methode lassen sich Matrizen  $A_L$  und  $B_L$  finden, mit denen die Messdaten angenähert werden können. Jedoch treten hier Abweichungen von dem ursprünglichen Modell auf.

Eine mögliche Erklärung dafür ist, dass die Einträge der mit Matrix-LSQ identifizierten Systemmatrix ungleich 0 eine Übertragung von Effekten ermöglichen, zu der das ursprüngliche Modell nicht in der Lage ist. Die in Kapitel 2 beschriebenen Vereinfachungen sind hier ein guter Ansatz. Besonders ist es möglich, dass der Wind und der Auftrieb durch die Nickrate eine relevante Rolle spielen.

Die Unterschiede in der ersten und vierten Zeile der Systemmatrix sprechen für einen Einfluss des Windes, weil somit  $\Delta\gamma = \Delta\theta - \Delta\alpha$  nicht mehr gilt.<sup>1</sup> Eine weitere Untersuchung und eine Betrachtung der Bedingungen beim Flugversuch sind hier notwendig, um eine fundierte Aussage treffen zu können.

<sup>1</sup>Der Bahnwinkel hat somit einen Einfluss auf den Anstellwinkel.

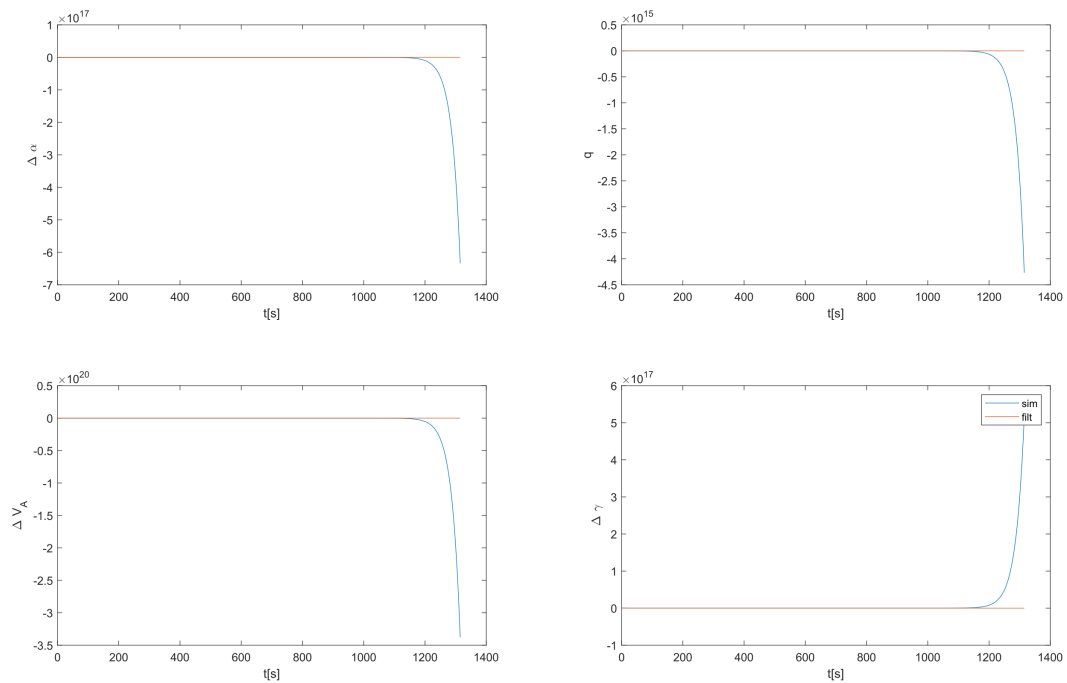


Abbildung 4.2. – Ergebnisse anhand des LSQ-Verfahrens

## 5. Systemidentifikation im Frequenzbereich

Neben der Systemidentifikation im Zeitbereich ist es auch möglich, die Parameterschätzung im Frequenzbereich durchzuführen. Die Analyse im Frequenzbereich bringt einige Vorteile mit sich, die im weiteren Verlauf dieser Dokumentation genauer beleuchtet werden. Ausgehend von den Fourier-transformierten Messwerten wird im Folgenden genauer auf die Idee der *Output Error*-Methode und auf einen Algorithmus zur Lösung des Schätzproblems eingegangen. Zum Schluss werden die Ergebnisse bewertet und der Blick auf mögliche Verbesserungen und andere Methoden gerichtet.

### 5.1. Fourier-Transformation

Die Grundlage für die Arbeit im Frequenzbereich bildet die Fourier-Transformation [1], mit der ein Verlauf im Zeitbereich in den Frequenzbereich transformiert werden kann. Dafür wird der diskrete *Fast Fourier Transform*-Algorithmus verwendet, welcher in Matlab bereits als `fft()` integriert ist. Das Fourier-transformierte Zustandsraummodell ergibt sich wie folgt [5]:

$$\begin{aligned} j\omega_k \tilde{\mathbf{x}}(k) &= \mathbf{A} \tilde{\mathbf{x}}(k) + \mathbf{B} \tilde{\mathbf{u}}(k) \\ \tilde{\mathbf{y}}(k) &= \mathbf{C} \tilde{\mathbf{x}}(k) + \mathbf{D} \tilde{\mathbf{u}}(k) = \mathbf{G}(k, \theta) \tilde{\mathbf{u}}(k) \\ \text{mit } \mathbf{G}(k, \theta) &= \mathbf{C}(j\omega_k \mathbf{I} - \mathbf{A})^{-1} \mathbf{B} + \mathbf{D} \end{aligned} \quad (5.1)$$

Hier bezeichnen  $\tilde{\mathbf{x}}$ ,  $\tilde{\mathbf{u}}$  und  $\tilde{\mathbf{y}}$  die Fourier-Transformation des Zustands-, Steuer bzw. Ausgangsvektors.  $\omega_k = 2\pi f_k = 2\pi k/T$  ist die Kreisfrequenz und  $\mathbf{G}(k, \theta)$  die Übertragungsfunktion einer Frequenz  $f_k$  abhängig vom Parametervektor  $\theta$ .

### 5.2. Output Error-Methode

Die Idee der *Output Error*-Methode (OEM) liegt in der Art der Fehlerbetrachtung. Es wird dabei der Fehler, der durch die mathematische Modellbildung entsteht, vernachlässigt. Das dynamische System wird als deterministisch angenommen. Unsicherheiten entstehen nur durch fehlerhafte und verrauschte Messungen. Ausgangspunkt der OEM ist das transformierte Zustandsraummodell (5.1). Im vorliegenden Fall entspricht der Ausgangsvektor  $\tilde{\mathbf{y}}$  dem Zustandsvektor  $\tilde{\mathbf{x}}$ . Die Matrizen  $\mathbf{C}$  und  $\mathbf{D}$  nehmen also folgende Gestalt an:

$$\begin{aligned} \mathbf{C} &= \mathbf{I} \\ \mathbf{D} &= \mathbf{0} \end{aligned} \quad (5.2)$$

Da der Zustandsvektor gleichzeitig der gemessene Zustand ist, kann der *Output Error* folgendermaßen formuliert werden:

$$\tilde{\mathbf{v}}(k, \theta) = \tilde{\mathbf{z}}(k) - \tilde{\mathbf{y}}(k, \theta) = \tilde{\mathbf{z}}(k) - \mathbf{G}(k, \theta) \tilde{\mathbf{u}}(k) \quad (5.3)$$

$\tilde{\mathbf{z}}$  entspricht dabei dem gemessenen Zustand, der zusammen mit dem Messrauschen aus dem übertragenen Zustand  $\tilde{\mathbf{y}}$  entsteht. Der übertragene Zustand kann aus den Steuergrößen  $\tilde{\mathbf{u}}$  durch

Multiplikation mit der Übertragungsmatrix  $\mathbf{G}(\theta)$  berechnet werden. Die Übertragungsmatrix ist, wie in Gleichung (5.1) gezeigt, abhängig vom zugrunde liegenden Modell und von den zu schätzenden Parametern  $\theta$ . Diese sind die Einträge der Matrizen des Zustandsraummodells. Das Ziel der Methode ist nun, den Ausgangsvektor, der aus dem Modell und seinen Parametern folgt, dem gemessenen Zustand möglichst gut anzunähern. Mathematisch bedeutet das, dass das Minimum einer Kostenfunktion gefunden werden soll. Die Kostenfunktion der OEM ist nach *Klein* und *Morelli* [5] die negative Log-Likelihood-Funktion mithilfe des Ansatzes des Ausgangsfehlers (5.3):

$$J(\theta) = N \sum_{k=0}^{N-1} \tilde{\mathbf{v}}^H(k, \theta) \mathbf{S}_{\nu\nu}^{-1} \tilde{\mathbf{v}}(k, \theta) + N \ln |\mathbf{S}_{\nu\nu}| \quad (5.4)$$

Die Matrix  $\mathbf{S}_{\nu\nu}$  gewichtet die Messfehler der einzelnen Steuergrößen auf die Zustandsgrößen. Wie bereits beschrieben, geht es nun darum, das Minimum dieser Kostenfunktion zu finden. Wir suchen also die Nullstelle ihrer Ableitung. Einer der wichtigsten Algorithmen zur Bestimmung von Nullstellen (nicht)-linearer Funktionen sowohl bei Eingrößenproblemen als auch bei Mehrgrößenproblemen ist der iterative *Newton-Raphson*-Algorithmus. Dabei wird die Nullstelle mithilfe der Richtung der Tangentensteigung iterativ angenähert. Das Problem lässt sich folgendermaßen beschreiben:

Gesucht ist die Nullstelle  $s$  einer Funktion  $f(x)$  mit  $f(s) = 0$  für  $x = s$ .  
 $\bar{x}$  sei ein Punkt in der Umgebung der Nullstelle.

Stellt man nun eine Taylor-Reihe um  $\bar{x}$  auf und vernachlässigt die Terme höherer Ordnung, erhält man die Iterationsvorschrift des *Newton-Raphson*-Verfahrens:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \text{ für } i = 0, 1, 2, \dots \quad (5.5)$$

Diese Vorschrift lässt sich umformulieren für einen mehrdimensionalen Funktionsvektor  $\mathbf{F}$ :

$$\underbrace{\begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}}_{\mathbf{J}^i} \underbrace{\begin{pmatrix} \Delta x_1 \\ \Delta x_2 \\ \vdots \\ \Delta x_n \end{pmatrix}}_{\Delta \mathbf{X}^{i+1}} = - \underbrace{\begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}}_{\mathbf{F}^i} \quad (5.6)$$

Da wir das Minimum der Kostenfunktion suchen ist unser Funktionsvektor die Ableitung der Kostenfunktion nach den einzelnen zu schätzenden Parametern.

$$\mathbf{F} = \frac{\partial J}{\partial \theta} \quad (5.7)$$

Wir suchen die Nullstelle dieser Funktion. Mithilfe von Gleichung (5.6) können wir die Vorschrift für unser Problem umschreiben:

$$\Delta \theta^{i+1} = - \left( \left[ \frac{\partial^2 J}{\partial \theta \partial \theta^T} \right]^{-1} \right)^i \cdot \left( \frac{\partial J}{\partial \theta} \right)^i \quad (5.8)$$

$$\theta^{i+1} = \theta^i + \Delta \theta^{i+1}$$



In jedem Iterationsschritt muss also ein lineares Gleichungssystem gelöst werden. Das größte Problem bei der Implementierung des Algorithmus ist die Berechnung der Ableitungen und der Hesse-Matrix in jedem Iterationsschritt. Diese können bei vielen zu schätzenden Parametern schnell sehr kompliziert werden. Da die Kostenfunktion als Summe über alle Frequenzen berechnet wird (vgl. Gleichung (5.4)), gilt dies auch für die Ableitungen. Dadurch wird zusätzliche Rechenzeit beansprucht. Der Algorithmus ist in Abb. 5.1 nochmals übersichtlich dargestellt.

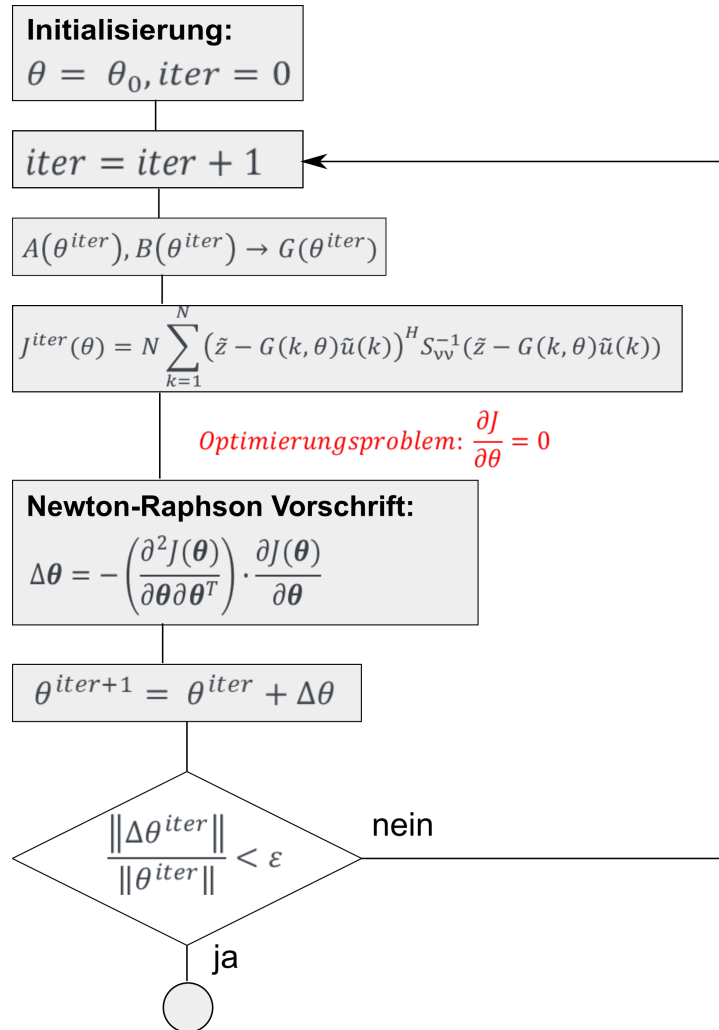


Abbildung 5.1. – Newton-Raphson-Algorithmus für die OEM

Der in Matlab implementierte Algorithmus wurde mit eigens dafür generierten Testdaten getestet. Er war dabei in der Lage, die gewählten Frequenzen zu treffen. Die Ergebnisse sind in Anhang A aufgeführt.

### 5.3. Vor- und Nachteile

Wie bereits erwähnt werden bei der Systemidentifikation im Frequenzbereich fouriertransformierte Signalverläufe betrachtet. Ein großer Vorteil besteht nun darin, dass die in der Identifikation berücksichtigten Frequenzpunkte frei gewählt werden können. Dies ist von Vorteil, wenn im Vorhinein bereits bekannt ist, welche Frequenzen für das Modell eine wichtige Rolle spielen. Auch wenn die Berechnungen im Algorithmus durchaus sehr aufwändig werden können, so ist

der Rechenaufwand unabhängig von der Länge des betrachteten Zeitraums bzw. der Anzahl der Datenpunkte im Zeitbereich.

Darüberhinaus lässt sich ein physikalisches Verständnis des Systems im Frequenzbereich gewinnen, was für viele Anwendungen sehr hilfreich sein kann. Im Reglerentwurf beispielsweise sind Frequenzverläufe von großer Wichtigkeit.

Ein schwerwiegender Nachteil ist die Notwendigkeit eines passenden Eingangssignals zum Erreichen einer guten Identifikation. Als Beispiel sei ein Frequency Sweep, auch Chirp-Signal genannt, bei welchem eine Schwingung mit steigender Frequenz auf eine Steuerung gegeben wird. Mit solchen passenden Steuerungen ist mit deutlich besseren Identifikationsergebnissen zu rechnen.

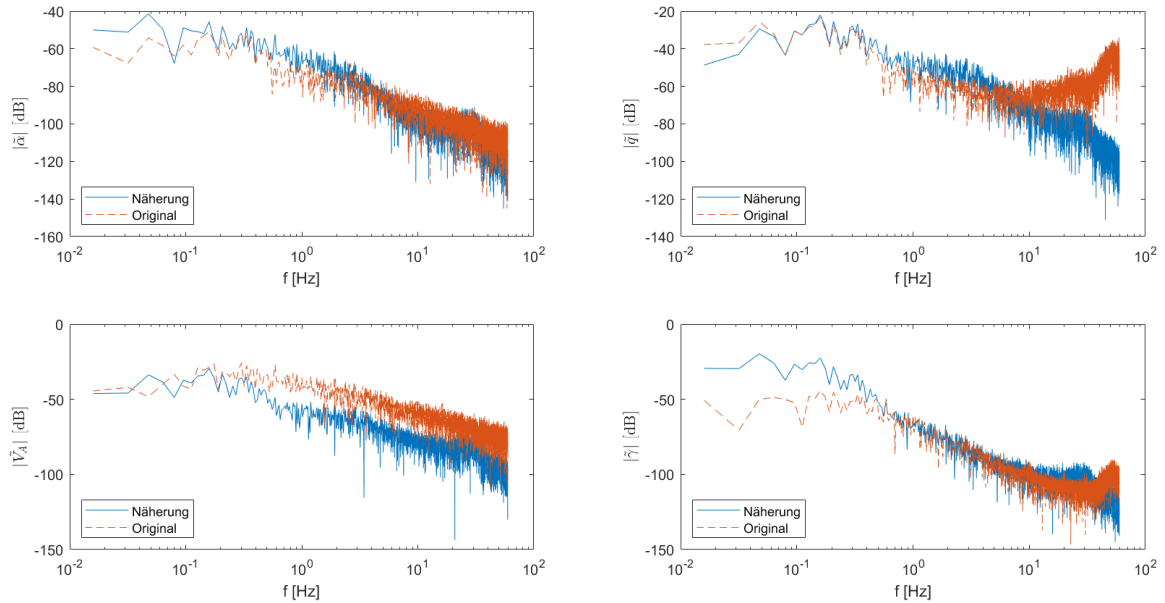
Die Bestimmung der einfachen und zweifachen Ableitungen für die Jacobi- und Hesse-Matrizen ist sehr aufwändig. In diesem Projekt wurde deshalb auf die Möglichkeit des symbolischen Rechnens in Matlab zurückgegriffen. Trotzdem bleibt dies ein Nachteil.

Schließlich bleibt noch die Berücksichtigung des Abtasttheorems nach Shannon ein wichtiger Punkt, um Frequenzverfälschungen zu vermeiden [3]. Demzufolge muss die Abtastfrequenz mehr als doppelt so groß sein wie die größte Frequenz im Signal. Dies stellte sich bei den vorliegenden Daten jedoch nicht als Problem heraus.

## 5.4. Ergebnisse

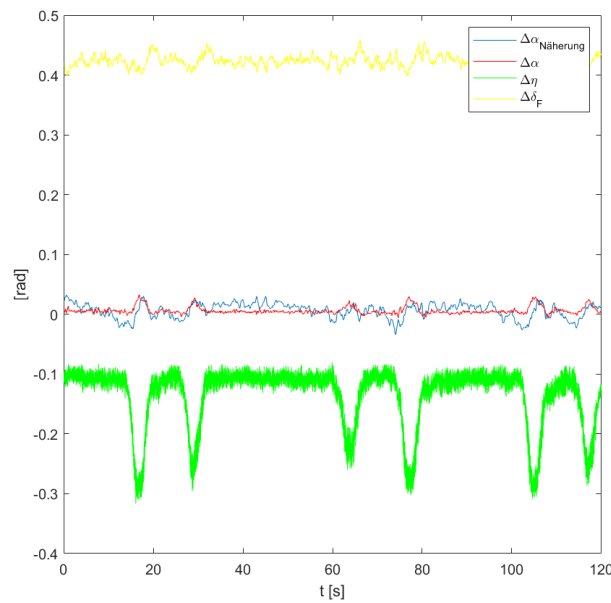
Abschließend soll der folgende Abschnitt noch einen kurzen Einblick in die Ergebnisse der Parameterschätzung im Frequenzbereich geben. Aus dem *Newton-Raphson*-Algorithmus erhält man eine Abschätzung des Parametervektors  $\theta$  und damit auch eine Abschätzung der Übertragungsmatrix  $G(\theta)$ . Mithilfe von  $G(\theta)$  und den gemessenen Steuergrößen lässt sich ein geschätzter Frequenzverlauf der Zustandsgrößen berechnen. In Abb. 5.2 sind sowohl die geschätzten Frequenzverläufe als auch die originalen Frequenzverläufe dargestellt. Beim Blick auf die originalen Daten (in rot dargestellt) fällt der erwartete Trend ins Auge. Die Übertragung findet hauptsächlich im Bereich niedriger Frequenzen statt. Die Verläufe knicken für große Frequenzen ab. Ausnahme ist die Nickrate  $q$ . Da wir allerdings nur die Bewegung während der Platzrunde betrachten, liegt  $q$  nahe bei 0 und wird hauptsächlich vom Messrauschen bestimmt. Dies könnte den Anstieg bei hohen Frequenzen erklären. Um ein besseres Bild vom Übertragungsverhalten von  $q$  zu bekommen, müsste beispielsweise im Bereich der Anstellwinkelschwingung geflogen werden. Allgemein zeigt sich, dass die Frequenzverläufe gut getroffen werden. Teilweise werden Ausschläge sogar exakt getroffen (vgl. Verlauf von  $q$  im niedrigen Frequenzbereich). Das häufigste Problem, das beobachtet werden konnte, war allerdings das Überschätzen bzw. Unterschätzen der Ausgangsverläufe. Deutlich zu sehen ist dies im Verlauf der Geschwindigkeit  $V_A$ . Ein Grund könnten die unterschiedlichen Größenordnungen der Zustandsgrößen sein. Dies würde erklären weshalb genau die Geschwindigkeit, mit der größten nominellen Abweichung vom Trimpunkt, unterschätzt wird. Eine Normalisierung der Zustandsgrößen wäre deshalb ein möglicher Ansatz zur Verbesserung der Ergebnisse (vgl. Kapitel 6).

Exemplarisch wurde der Frequenzverlauf des Anstellwinkels  $\alpha$  wieder in den Zeitbereich zurücktransformiert. Das Ergebnis ist in Abb. 5.3 dargestellt. Zusätzlich sind die Steuerverläufe von Elevator und Schub aufgetragen. Es ist gut zu erkennen, dass sogar die Ausschläge im Anstellwinkel abgeschätzt werden und der geschätzte Verlauf ein brauchbares Ergebnis liefert. Weniger realistisch sind allerdings die geschätzten Matrizen des Zustandsraummodells  $A$  und  $B$ . Auch die Verläufe der anderen Größen werden nicht ganz so gut angenähert. Das Problem könnte hier bereits vor der Anwendung des Algorithmus liegen. Die gegebenen Flugdaten sind speziell für ein Schätzverfahren im Frequenzbereich unzureichend. Es ist vor allem in-



**Abbildung 5.2.** – Frequenzverläufe der vier Zustandsgrößen der Längsbewegung: in rot sind die Frequenzverläufe der Messdaten und in blau die Frequenzverläufe des geschätzten Modells dargestellt

interessant, in welchem Frequenzbereich ein Steuereingang auf die Zustandsgrößen übertragen wird. Dies erkennt man dann, wenn die Frequenzen, beispielsweise in einem *Frequency Sweep*, abgeflogen werden. Dies wäre eine mögliche Erklärung für die weniger guten Ergebnisse einiger Abschätzungen.



**Abbildung 5.3.** – Zeitverlauf der Steuergrößen und der Zustandsgröße  $\alpha$

## 6. Zusammenfassung und Ausblick

Abschließend lässt sich sagen, dass die Datenvorbereitung ein sehr wichtiger und auch zeitintensiver Teil der gesamten Arbeit war. Die richtigen Messsignale aus der Fülle an Originaldaten zu finden und auf Plausibilität zu prüfen legt den Grundstein für alle weiteren Schritte. Schwierig ist außerdem die Auswahl eines geeigneten Zeitbereichs zur Identifikation.

Das LSQ-Verfahren im Zeitbereich lieferte keine sonderlich guten Ergebnisse. Deutlich bessere Approximationen der Messdaten ließen sich mit dem Matrix-LSQ-Verfahren erreichen. Dieses stellte sich als sehr schnell und vergleichsweise einfach in der Implementierung heraus. Aus flugmechanischer Sicht sind allerdings Einbußen in den bestimmten Parametern hinzunehmen, da sich keine Vorgaben bzgl. der bereits bekannten Einträge in der System- und Steuermatrix treffen lassen.

Es zeigte sich, dass der Flug in den Platzrunden weniger geeignet für eine Identifikation im Frequenzbereich ist. Der Verlauf des Steuersignals ist hier von großer Relevanz, bestimmte Manöver, wie ein *Frequency Sweep*, sollten hier zu deutlich besseren Ergebnissen führen. Nichtsdestotrotz konnten bestimmte Zustandsgrößen angenähert werden. Der Implementierungsaufwand für die *Output Error*-Methode ist deutlich höher verglichen mit den Zeitbereich-Verfahren; die symbolische Berechnung der sehr länglichen Ableitungen in Matlab ermöglichte überhaupt erst eine sinnvolle Anwendung.

An dieser Stelle sei erneut erwähnt, dass in der vorliegenden Arbeit lediglich der Platzrundenflug berücksichtigt wurde. In den restlichen Flugdaten steckt aber womöglich ebenso Potenzial zur Bestimmung des flugmechanischen Modells, vor allem, wenn sich noch passende Manöver (z.B. Anregung der Eigenschwingung des Flugzeugs) finden lassen. Als nächster Schritt bietet sich also eine genaue Untersuchung des bisher vernachlässigten Flugs an.

Möglichkeiten zur Verfeinerung und Optimierung der Identifikation selbst gibt es viele. An erster Stelle bietet sich eine Normalisierung der Signale an, d.h. eine Skalierung mit den Maximalwerten eines jeden Verlaufs, sodass die Werte danach in der gleichen Größenordnung liegen. Dies ist sonst nicht der Fall, die Geschwindigkeit erreicht weit höhere Absolutwerte als alle anderen Zustände, was in den verwendeten Algorithmen einer stärkeren Gewichtung des Geschwindigkeitssignalls entspricht. Eine Gewichtung ließe sich aber außerdem bewusst vornehmen und damit der Fokus speziell auf wichtige Größen legen. Weiterhin könnten die zugrundeliegenden Modelle vereinfacht werden. In der Längsbewegung würde das bedeuten, zwei reduzierte Modelle für die Anstellwinkelschwingung einerseits und die Phygoide andererseits zu erstellen und die Identifikationen getrennt durchzuführen.

Weit aufwändiger, aber auch sehr vielversprechend, ist eine Identifikation im Zeitbereich mit einem nichtlinearen Modell, welches aus verschiedenen zu identifizieren Untermodellen (z.B.: Giermodell, Aerodynamisches Modell und Propeller) besteht. Als Beispiel werden die aerodynamischen Kräfte separat aus den Beiwerten berechnet. Die Aufgabe der Systemidentifikation ist dann eine Abschätzung der Polaren aus den gemessenen Daten durchzuführen.

Im Frequenzbereich wurde mit der *Output Error*-Methode nur der Fehler im gemessenen Signal berücksichtigt. Alternativ wäre hier eine Betrachtung des Modellfehlers bei gleichzeitiger Vernachlässigung des Messfehlers möglich, wie es die *Equation Error*-Methode vorsieht. Zu guter Letzt lassen sich diese beiden Methoden auch kombinieren.

# Literaturverzeichnis

- [1] BENDAT, J. S. ; PIERSOL, A. G.: *Random Data: Analysis and Measurement Procedures*. John Wiley & Sons, Inc., 1986. – ISBN 0–471–04000–2
- [2] FICHTER, W. ; GRIMM, W. : *Flugmechanik*. 2009
- [3] GRIMM, W. : *Skriptum Digitale Regelung und Filterung*. 2017
- [4] GSCHWANDTNER, F. : *Bestimmung eines Pilotenmodells mit Flugmessdaten aus einem UL-Helikopter*. 2019
- [5] KLEIN, V. ; MORELLI, E. A.: *Aircraft System Identification*. American Institute of Aeronautics and Astronautics, Reston, VA, 2006. – ISBN 1–56347–832–3
- [6] MANDRY, N. : *Modul 5: Kleinste Fehlerquadrate für eine Matrixgleichung: Matrix-LSQ*. 2021

## A. Test des Algorithmus im Frequenzbereich

Zum Test der *Output Error*-Methode mit dem *Newton-Raphson*-Algorithmus wurde ein sinusförmiges Signal mit einer bestimmten Frequenz in der Höhenrudersteuerung sowie ein Sinus-Signal im Anstellwinkel erstellt und der Algorithmus verwendet, um die entsprechende Übertragung zu identifizieren. Mit der identifizierten Übertragungsfunktion wurde dann eine Näherung des Zustandsverlaufs berechnet. Wie sich in Abb. A.1 erkennen lässt, wird die ursprüngliche Frequenz gut angenähert. Der Anstellwinkelverlauf wird vom Algorithmus mit einem leichten Phasenversatz noch gut getroffen (vgl. Abb. A.2).

Die anderen Zustände wurden in diesem Test nicht betrachtet, der Schubdrosselgrad wurde konstant gehalten.

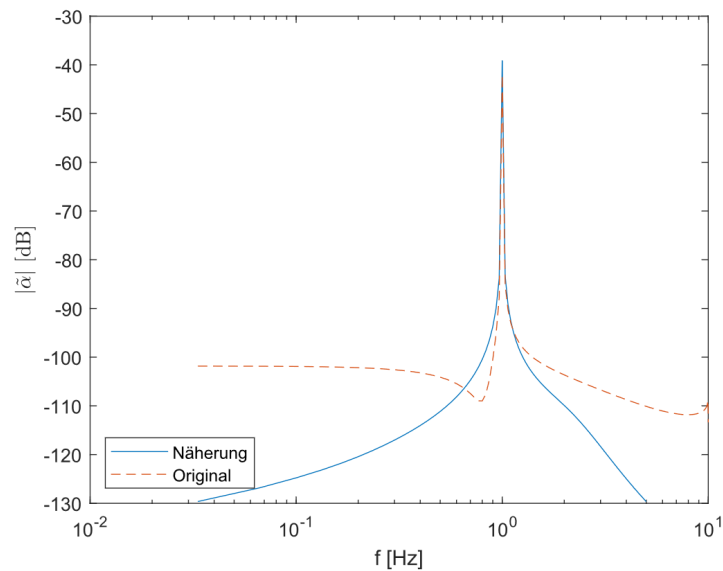


Abbildung A.1. – Betragskennlinie des Originalsignals sowie der Näherung

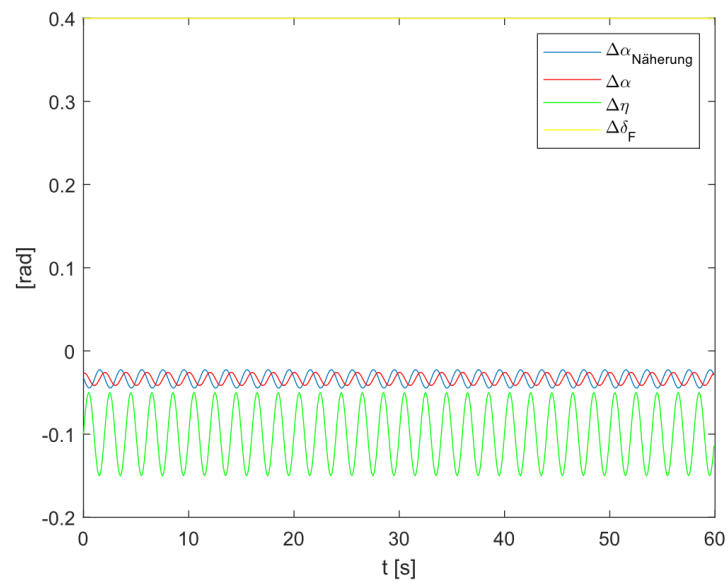


Abbildung A.2. – Zeitverläufe des Originalsignals sowie der Näherung und der Steuerungen