

Übungsaufgabe in Luft- und Raumfahrtlabor

Erweiterter Kalman Filter zur quaternionsbasierten Orientierung (QEKf)

Stefan Lindörfer

15. Februar 2021

In dieser Aufgabe soll das bereits verwendete Konzept des erweiterten Kalman Filters auf das mathematische Konstrukt der Quaternionen als Orientierungsmodell ausgedehnt werden. Quaternionen bieten unter anderem den Vorteil, dass bei ihrer Beschreibung keine Singularitäten (z.B. Gimbal Lock bei Euler-Winkel) auftreten können. Ein Nachteil ist jedoch die Lesbarkeit, weshalb sie hier zu Anschauungszwecken in Tait-Bryan-Winkel umgerechnet werden.

Die drei-dimensionale Lage eines Multikopters soll durch ein QEKf geschätzt werden. Es stehen drei Sensoren zur Verfügung: Beschleunigungssensor, Gyroskop und Magnetometer.

Das Zustandsraummodell setzt sich aus dem zehn dimensional Vektor

$$\vec{x}(k) = (q_{NB}(k) \quad \omega_B(k) \quad x_g(k))^T \quad (1)$$

zusammen. Dabei ist $q_{NB}(k)$ das vier-dimensionale Lagequaternion, $\omega_B(k)$ die drei-dimensionale Drehgeschwindigkeit und $x_g(k)$ der ebenfalls drei Dimensionen beinhaltene Gyroskopbias, der aufgrund seines potenziell starken Einflusses mitgeschätzt werden soll. Das Gyroskop wird hierbei als Systemeingang verwendet. Weiterhin wird ein North-East-Down Koordinatensystem genutzt. Die Umrechnung von Navigationsframe N in Bodyframe B (und umgekehrt) erfolgt durch entsprechende Rotationsmatrizen. Ein Quaternion q wird dargestellt als

$$\mathbf{q} = q_0 + q_1 \cdot i + q_2 \cdot j + q_3 \cdot k \triangleq (q_0 \quad q_1 \quad q_2 \quad q_3)^T \quad (2)$$

Die weiteren Rechenregeln mit Quaternionen werden hierbei nicht detaillierter erläutert, können jedoch leicht auf entsprechenden Quellen nachvollzogen werden.

1 Systemmodell

Zunächst ist ein vollständiges Systemmodell aufzustellen. Dabei wird nach einem Kochrezept aus der Übung vorgegangen. Es werden alle Informationen über das Zustandsraummodell, die Sensoren und erforderlichen Transformationen zusammengetragen und anschließend für das Filter entsprechend aufbereitet, sodass die erforderlichen Gleichungen des Kalman Filters aufgestellt werden können.

1.1 Sensormodell

Die folgenden Gleichungen modellieren die Sensormessungen von Gyroskop (Index g), Magnetometer (m) und Accelerometer (a) im Bodyframe:

$$\vec{y}_g = \vec{\omega}_B + \vec{x}_g + \vec{v}_g \quad (3)$$

$$\vec{y}_m = \vec{m}_B + \vec{x}_m + \vec{v}_m \quad (4)$$

$$\vec{y}_a = \vec{a}_B - \vec{g}_B + \vec{x}_a + \vec{v}_a \quad (5)$$

Dabei ist ω_B die Rotationsgeschwindigkeit, m_B das Magnetfeld, a_B die Beschleunigung, g_B die Erdbeschleunigung (hier: Offset), x_i die Sensorbiaswerte für jeden Sensor und v_i das Rauschen jeden Sensors.

Zudem existiert für jeden Sensor die entsprechende Varianz: σ_g^2 , σ_m^2 und σ_a^2 . Berechnet werden können diese über das Quadrat der Stichprobenstandardabweichung eines geeigneten Zeitintervalls (vorzugsweise mit Messwerten während des Fluges).

1.2 Systemzustand x

Der eingangs bereits kompakt dargestellte Systemzustandsvektor (Gl. 1) ergibt ausführlich somit:

$$\vec{x}(k) = \left(\begin{array}{l} \left. \begin{array}{l} q_0(k) \\ q_1(k) \\ q_2(k) \\ q_3(k) \end{array} \right\} =: q_{NB}(k) \\ \left. \begin{array}{l} \omega_{B_x}(k) \\ \omega_{B_y}(k) \\ \omega_{B_z}(k) \end{array} \right\} =: \omega_B(k) \\ \left. \begin{array}{l} x_{g_x}(k) \\ x_{g_y}(k) \\ x_{g_z}(k) \end{array} \right\} =: x_g(k) \end{array} \right) \quad (6)$$

1.3 Systemeingang u & Messvektoren y

Das Gyroskop soll – wie bereits erwähnt – als Systemeingang verwendet werden, während das Accelerometer und das Magnetometer als Sensoreingabe einfließen. Alle drei Sensoren liefern je einen Messwert für jede der drei Achsen (x , y und z).

Damit folgt aus den Gleichungen 3-5, dass die Eingangsvektoren der Sensoren aus jeweils drei Zeilen bestehen.

1.4 Systemmodell

Die Zustandsänderungen für die nächste Filteriteration erfolgt bei Quaternionen über eine Integration der Änderungen mittels der Euler-Methode. Für die Änderung eines Quaternion q gilt nach der Definition des Hamilton-Produktes für Quaternionen (Index L):

$$\dot{q} = \frac{1}{2} \cdot (\mathbf{q})_L \cdot \begin{pmatrix} 0 \\ \omega_B \end{pmatrix} \quad (7)$$

Damit folgt nach Gleichung 7 und nach Ersetzung von $(\mathbf{q})_L$ durch die entsprechende Matrixdarstellung des Quaternions für diesen Teil des Zustandsraummodells:

$$q_{NB}(k+1) = q_{NB}(k) + \frac{1}{2} \cdot \dot{q}_{NB}(k) \cdot \Delta t = \frac{1}{2} \cdot \begin{pmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ \omega_{B_x} \\ \omega_{B_y} \\ \omega_{B_z} \end{pmatrix} \cdot \Delta t \quad (8)$$

Ausmultipliziert und auf die einzelnen Komponenten aufgeteilt ergibt sich damit:

$$q_0(k+1) = q_0(k) + \frac{1}{2} \cdot \left(- (q_1 \omega_{B_x} + q_2 \omega_{B_y} + q_3 \omega_{B_z}) \right) \cdot \Delta t \quad (9)$$

$$q_1(k+1) = q_1(k) + \frac{1}{2} \cdot \left(q_0 \omega_{B_x} - q_3 \omega_{B_y} + q_2 \omega_{B_z} \right) \cdot \Delta t \quad (10)$$

$$q_2(k+1) = q_2(k) + \frac{1}{2} \cdot \left(q_3 \omega_{B_x} + q_0 \omega_{B_y} - q_1 \omega_{B_z} \right) \cdot \Delta t \quad (11)$$

$$q_3(k+1) = q_3(k) + \frac{1}{2} \cdot \left(- q_2 \omega_{B_x} + q_1 \omega_{B_y} + q_0 \omega_{B_z} \right) \cdot \Delta t \quad (12)$$

Die Zustandsgleichungen für die Drehgeschwindigkeit ω_B lassen sich leichter berechnen und ergeben sich für die jeweilige Achse aus der Differenz von Gyroskopmessung und entsprechendem Bias (Gl. 14). Hier muss also zwischen den Messwerten des Sensors und der letztendlich berechneten Drehrate unterschieden werden. Die Größe $\vec{\omega}_B$ ist Teil des Zustandsraummodells, während \vec{y}_g , also die Rohwerte des Gyroskops, Teil des Systemeingangs u ist.

$$\vec{\omega}_B(k+1) = \vec{u}(k) - \vec{x}_g(k) \quad (13)$$

$$= \vec{y}_g(k) - \vec{x}_g(k) \quad (14)$$

Für den Gyroskopbias wird außerdem die Annahme getroffen, dass dieser in etwa konstant bleibt. Tatsächlich ist der Bias aber temperaturabhängig, was hier jedoch vernachlässigt werden soll. Damit gilt für den Bias:

$$\vec{x}_g(k+1) = \vec{x}_g(k) \quad (15)$$

Das vollständige nicht-lineare Zustandsraummodell ergibt sich somit durch Einsetzen der Gleichungen 9-15 in Gleichung 6.

1.5 Linearisierung des Systemmodells f

Damit das Filter die Zustandsgleichungen f verwenden kann, ist – wie bereits in der vorherigen Übung zum EKF gezeigt wurde – eine Linearisierung des Zustandsraummodells nötig. Dies wird mittels der Jacobi-Matrix bewerkstelligt:

$$f_x = \left. \frac{\partial f[k, \vec{x}(k)]}{\partial \vec{x}(k)} \right|_{\vec{x}=\hat{\vec{x}}_{pos}(k)} \quad (16)$$

Dabei wird in die 10-dimensionale, quadratische Zustandsmatrix F bzw. f_x und die Systemeingangsmatrix f_u mit dem Gyroskop als Eingabe unterschieden. Es gilt für f_x :

$$f_x = \begin{pmatrix} \frac{\partial q_0}{\partial q_0} & \frac{\partial q_0}{\partial q_1} & \cdots & \frac{\partial q_0}{\partial x_{gy}} & \frac{\partial q_0}{\partial x_{gz}} \\ \frac{\partial q_1}{\partial q_0} & \ddots & \ddots & \ddots & \frac{\partial q_1}{\partial x_{gz}} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \frac{\partial x_{gy}}{\partial q_0} & \ddots & \ddots & \ddots & \frac{\partial x_{gy}}{\partial x_{gz}} \\ \frac{\partial x_{gz}}{\partial q_0} & \frac{\partial x_{gz}}{\partial q_1} & \cdots & \frac{\partial x_{gz}}{\partial x_{gy}} & \frac{\partial x_{gz}}{\partial x_{gz}} \end{pmatrix} \quad (17)$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 & -\frac{1}{2}q_1\Delta t & -\frac{1}{2}q_2\Delta t & -\frac{1}{2}q_3\Delta t & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & \frac{1}{2}q_0\Delta t & -\frac{1}{2}q_3\Delta t & \frac{1}{2}q_2\Delta t & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \frac{1}{2}q_3\Delta t & \frac{1}{2}q_0\Delta t & -\frac{1}{2}q_1\Delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -\frac{1}{2}q_2\Delta t & \frac{1}{2}q_1\Delta t & \frac{1}{2}q_0\Delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (18)$$

Die Komponenten des Quaternions aus der vorherigen Iteration werden übernommen, sowie deren Änderungen integriert. Der als konstant angenommene Bias wird unverändert aus dem Initialzustandswert (z.B. aus einer Kalibrierung) für den gesamten Filterprozess übernommen. Zur Berechnung der resultierenden Drehrate wird der Bias mit negativem Vorzeichen auf den Rohwert des Gyroskops addiert, der nachfolgend durch den Systemeingang hinzukommt.

Um die entsprechende Matrix für den Systemeingang aufstellen zu können, muss zunächst der Systemeingangsvektor \vec{u} bestimmt werden. Dieser enthält die Rohwerte des Gyroskops, die eben im weiteren Verlauf noch um den Bias bereinigt werden müssen, bevor sie als Drehrate ω_B verwendet werden können. Dementsprechend folgt für \vec{u} :

$$\vec{u} = (y_{gx} \quad y_{gy} \quad y_{gz})^T \quad (19)$$

Demnach gilt analog für f_u

$$f_u = \left. \frac{\partial f[k, \vec{u}(k)]}{\partial \vec{u}(k)} \right|_{\vec{u}=\hat{\vec{u}}(k)} \quad (20)$$

und damit für die entsprechende (10×3) -Matrix:

$$\begin{aligned}
 f_u &= \begin{pmatrix} \frac{\partial q_0}{\partial \omega_{B_x}} & \cdots & \frac{\partial q_0}{\partial \omega_{B_z}} \\ \frac{\partial q_1}{\partial \omega_{B_x}} & \ddots & \frac{\partial q_1}{\partial \omega_{B_z}} \\ \vdots & \ddots & \vdots \\ \frac{\partial x_{gz}}{\partial \omega_{B_x}} & \cdots & \frac{\partial x_{gz}}{\partial \omega_{B_z}} \end{pmatrix} \\
 &= \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \tag{21}
 \end{aligned}$$

Damit sind beide Matrizen bestimmt um die Zustandsvorhersage im Filter durchführen zu können:

$$\vec{x}(k+1) = f[k, \vec{x}(k), \vec{u}(k)] + \vec{v}(k) = \underbrace{f[k, \vec{x}(k)]}_{f_x \cdot \vec{x}(k) :=} + \underbrace{f[k, \vec{u}(k)]}_{f_u \cdot \vec{u}(k) :=} + \vec{v}(k) \tag{22}$$

Die Zustandsvorhersage setzt sich also aus der Multiplikation von f_x mit $x(k)$, plus der Multiplikation von f_u mit $u(k)$ zusammen. Zusätzlich enthalten die Messungen des Systemeingangs einen Rauschanteil $\vec{v}(k)$.

1.6 Modell zur Messvorhersage h & Messfunktionen H

Als nächstes werden die Gleichungen zur Messvorhersage aufgestellt. Den Anfang macht der Beschleunigungssensor. Dessen gemessener Beschleunigungsvektor im Navigationsframe muss in das Bodyframe transformiert werden. Dies geschieht mittels folgender Gleichung:

$$\vec{h}_a[k, x(k)] = \begin{pmatrix} -2 \cdot (q_1 q_3 - q_0 q_2) \\ -2 \cdot (q_2 q_3 + q_0 q_1) \\ -(q_0^2 - q_1^2 - q_2^2 + q_3^2) \end{pmatrix} \tag{23}$$

Die entsprechende (3×10) Jacobi-Matrix lautet somit:

$$\begin{aligned}
H_a(k) &= \frac{\partial h_a[k, \vec{x}(k)]}{\partial \vec{x}(k)} \bigg|_{\vec{x}=\hat{\vec{x}}_{pos}(k)} \\
&= \begin{pmatrix} \frac{\partial h_{ax}}{\partial q_0} & \frac{\partial h_{ax}}{\partial q_1} & \cdots & \frac{\partial h_{ax}}{\partial x_{gz}} \\ \frac{\partial h_{ay}}{\partial q_0} & \frac{\partial h_{ay}}{\partial q_1} & \cdots & \frac{\partial h_{ay}}{\partial x_{gz}} \\ \frac{\partial h_{az}}{\partial q_0} & \frac{\partial h_{az}}{\partial q_1} & \cdots & \frac{\partial h_{az}}{\partial x_{gz}} \end{pmatrix} \\
&= 2 \cdot \begin{pmatrix} q_2 & -q_3 & q_0 & -q_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -q_1 & -q_0 & -q_3 & -q_2 & 0 & 0 & 0 & 0 & 0 & 0 \\ -q_0 & q_1 & q_2 & -q_3 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}
\end{aligned} \tag{24}$$

Das Magnetometer wird für die Bestimmung des Yaw-Winkels verwendet. Dazu existiert folgende Transformation:

$$h_\psi[k, x(k)] = \text{atan2}(2 \cdot (q_0 q_3 + q_1 q_2), 1 - 2 \cdot (q_2^2 + q_3^2)) \tag{25}$$

Die entsprechende (1×10) Jacobi-Matrix setzt sich demnach wie folgt zusammen:

$$\begin{aligned}
H_\psi(k) &= \frac{\partial h_\psi[k, \vec{x}(k)]}{\partial \vec{x}(k)} \bigg|_{\vec{x}=\hat{\vec{x}}_{pos}(k)} \\
&= \begin{pmatrix} \frac{\partial h_\psi}{\partial q_0} & \frac{\partial h_\psi}{\partial q_1} & \cdots & \frac{\partial h_\psi}{\partial x_{gz}} \end{pmatrix} \\
&= \begin{pmatrix} \frac{\partial h_\psi}{\partial q_0} & \frac{\partial h_\psi}{\partial q_1} & \frac{\partial h_\psi}{\partial q_2} & \frac{\partial h_\psi}{\partial q_3} & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}
\end{aligned} \tag{26}$$

Die vier Ableitungen von h_ψ , die erkennbar ungleich Null sind, lauten:

$$\frac{\partial h_\psi}{\partial q_0} = \frac{2 \cdot q_3 \cdot (1 - 2 \cdot (q_2^2 + q_3^2))}{4 \cdot (q_0 q_3 + q_1 q_2)^2 + (1 - 2 \cdot (q_2^2 + q_3^2))^2} \tag{27}$$

$$\frac{\partial h_\psi}{\partial q_1} = \frac{2 \cdot q_2 \cdot (1 - 2 \cdot (q_2^2 + q_3^2))}{4 \cdot (q_0 q_3 + q_1 q_2)^2 + (1 - 2 \cdot (q_2^2 + q_3^2))^2} \tag{28}$$

$$\frac{\partial h_\psi}{\partial q_2} = \frac{8 \cdot q_2 \cdot (q_0 q_3 + q_1 q_2) + 2 \cdot q_1 \cdot (1 - 2 \cdot (q_2^2 + q_3^2))}{4 \cdot (q_0 q_3 + q_1 q_2)^2 + (1 - 2 \cdot (q_2^2 + q_3^2))^2} \tag{29}$$

$$\frac{\partial h_\psi}{\partial q_3} = \frac{8 \cdot q_3 \cdot (q_0 q_3 + q_1 q_2) + 2 \cdot q_0 \cdot (1 - 2 \cdot (q_2^2 + q_3^2))}{4 \cdot (q_0 q_3 + q_1 q_2)^2 + (1 - 2 \cdot (q_2^2 + q_3^2))^2} \tag{30}$$

1.7 Prozessrauschkovarianzmatrix Q

Die Prozessrauschkovarianzmatrix Q , die Rauschen und Störungen zwischen zwei Filteriterationen beschreibt, lässt sich erneut mit dem bereits bekannten Zusammenhang $Q = F_U \cdot U \cdot F_U^T$ berechnen. Dazu müssen also noch die beiden Matrizen F_U und U bestimmt werden.

Das Systemeingangsrauschen U setzt sich – wie bereits bekannt – aus den Varianzen der Sensoren zusammen. Diese können leicht mit den Sensordaten berechnet werden. Die Varianz des Bias (näherungsweise konstant) wird sehr klein gewählt und auf

$$\sigma_{x_g}^2 = 1.0 \cdot 10^{-11} \text{ (rad/s)}^2 \tag{31}$$

festgesetzt. Damit ergibt sich für die Matrix des Systemeingangsrauschens:

$$U = \begin{pmatrix} \sigma_g^2 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_g^2 & 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_g^2 & 0 & 0 & 0 \\ 0 & 0 & 0 & \sigma_{x_g}^2 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_{x_g}^2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_{x_g}^2 \end{pmatrix} \quad (32)$$

Die Matrix F_U beschreibt das Rauschen nach den Systemeingängen und wird ebenfalls durch eine entsprechende (10×6) Jacobi-Matrix gebildet. Die zugrunde gelegten Gleichungen sind in diesem Fall 3 und 4, da nur diese zum Systemeingang gehören. Damit werden alle Gleichungen des Zustandsraummodells (siehe 1.2) nach den Rauschanteilen abgeleitet. Speziell die Gleichungen 13 und 15, die das Rauschen des Gyroskops an sich (v_g) und das Rauschen des Bias (v_{x_g}) beschreiben.

$$F_U = \frac{\partial f[k, \vec{x}(k), u(k)]}{\partial v} \quad (33)$$

$$= \begin{pmatrix} \frac{\partial q_0}{\partial v_{gx}} & \frac{\partial q_0}{\partial v_{gy}} & \cdots & \frac{\partial q_0}{\partial v_{xgy}} & \frac{\partial q_0}{\partial v_{xgz}} \\ \frac{\partial q_1}{\partial v_{gx}} & \ddots & \ddots & \ddots & \frac{\partial q_1}{\partial v_{xgz}} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \frac{\partial x_{gy}}{\partial v_{gx}} & \ddots & \ddots & \ddots & \frac{\partial x_{gy}}{\partial v_{xgz}} \\ \frac{\partial x_{gz}}{\partial v_{gx}} & \frac{\partial x_{gz}}{\partial v_{gy}} & \cdots & \frac{\partial x_{gz}}{\partial v_{xgy}} & \frac{\partial x_{gz}}{\partial v_{xgz}} \end{pmatrix} \quad (34)$$

$$= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad (35)$$

Mit der im Kapitelanfang erwähnten Gleichung kann damit die Prozessrauschkovarianzmatrix Q berechnet werden:

$$Q = F_U \cdot U \cdot F_U^T$$

$$= \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sigma_g^4 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \sigma_g^4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \sigma_g^4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{x_g}^4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{x_g}^4 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_{x_g}^4 \end{pmatrix} \quad (36)$$

1.8 Messrauschkovarianzmatrix R

Zuletzt ist das Messrauschen zu bestimmen, welches eine Aussage über die Güte bzw. die Streuung von Messwerten eines Sensors (und damit auch deren Zuverlässigkeit) gibt. Die Dimensionierung der jeweiligen quadratischen Matrizen ist entsprechend der Anzahl der Komponenten des Sensors: So fließt der Beschleunigungssensor mit drei Achsen ein, während das Magnetometer indirekt über die Yaw-Winkel-Berechnung als Skalar eingebracht wird.

Zusammengesetzt wird die Matrix – wie bereits auch in den vorherigen Übungen – aus den Varianzen der Sensoren für die jeweilige Achse bzw. der Varianz der Yaw-Winkel Berechnung, die separat bestimmt werden muss: Mit mehreren Werten des Magnetometers wird der Yaw-Winkel berechnet. Aus diesen Daten kann dann die Varianz bestimmt werden.

Sofern kein korreliertes Rauschen vorliegt, ist die entsprechende Messrauschkovarianzmatrix stets eine Diagonalmatrix.

$$R_a = \begin{pmatrix} \sigma_a^2 & 0 & 0 \\ 0 & \sigma_a^2 & 0 \\ 0 & 0 & \sigma_a^2 \end{pmatrix} \quad R_\psi = (\sigma_\psi^2) \quad (37)$$

1.9 Gemeinsamer Korrekturschritt

Für den Fall, dass die Samplerraten (Frequenzen) aller Sensoren gleich sind, zu jeder Filteriteration also je ein aktueller Messwert aller Sensoren zur Verfügung steht, können die vorher aufgestellten Matrizen auch zusammengeführt werden, um nur einen einzigen Korrekturschritt im Filter durchführen zu müssen. Andernfalls müssen die Korrekturschritte aufgeteilt werden, wie dies in vorherigen Übungen der Fall war.

Für den Fall eines einzigen Korrekturschrittes, setzen sich die finalen Matrizen wie folgt zusammen:

$$y(k) = \begin{pmatrix} [\vec{y}_a(k)] \\ [y_{psi}(k)] \end{pmatrix} \quad (38)$$

$$h[k, x(k)] = \begin{pmatrix} [\vec{h}_a[k, x(k)]] \\ [h_{psi}[k, x(k)]] \end{pmatrix} \quad (39)$$

$$H[k, x(k)] = \begin{pmatrix} [H_a[k, x(k)]] \\ [H_{psi}[k, x(k)]] \end{pmatrix} \quad (40)$$

$$R = \begin{pmatrix} [R_a] & 0_{3 \times 1} \\ 0_{1 \times 3} & [R_\psi] \end{pmatrix} \quad (41)$$

1.10 Anfangszustände P_0 und x_0

Da nun alle erforderlichen Informationen und Gleichungen vorliegen, können die Initialwerte für die beiden Anfangszustände der Kovarianzmatrix P und des Systemzustands bestimmt werden.

Aus den Sensormessdaten kann über einen bestimmten Zeitraum (z.B. 1 s) ein Mittelwert der jeweiligen Sensoren berechnet werden, der für die weitere Berechnung der Startzustände (dies beinhaltet die Berechnung der YPR-Winkel) hergenommen werden kann.

Für den Startzustand wird eine ruhende Position angenommen, womit sich die Drehraten für alle Komponenten auf Null setzen lassen sowie der Bias des Gyroskops bestimmt werden kann. Damit sind alle Felder des Startzustandes befüllt. Beispielhaft seien hier die Startwerte der durchgeführten Simulation angegeben:

$$\phi_{Roll,0} = -0,0570 \text{ rad} \quad (42)$$

$$\theta_{Pitch,0} = 0,1320 \text{ rad} \quad (43)$$

$$\psi_{Yaw,0} = -0,3956 \text{ rad} \quad (44)$$

$$\sigma_a^2 = 0,2612 \text{ g}^2 \quad (45)$$

$$\sigma_g^2 = 0,0003 \text{ (rad/s)}^2 \quad (46)$$

$$\sigma_{yaw}^2 = 0,2075 \text{ rad}^2 \quad (47)$$

Aus den drei Winkeln (Roll, Pitch und Yaw) ergibt sich darüber hinaus das folgende Quaternion als Startzustand:

$$\vec{q}_{Start} = (0,9783 \quad -0,0149 \quad 0,0702 \quad -0,1942)^T \quad (48)$$

Für den Bias des Gyroskops gilt:

$$\vec{x}_g = (0,0353 \quad 0,0399 \quad 0,0775)^T \text{ rad} \quad (49)$$

Zusammenfassend ergibt sich somit der folgende Startzustand x_0 :

$$x_0 = ([\vec{q}_{Start}] \quad 0_{3 \times 1} \quad [\vec{x}_g])^T \quad (50)$$

Für das Prozessrauschen zu Beginn, also wie sehr sich das Filter auf die Initialzustandswerte verlassen kann, kann vereinfacht

$$P_0 = 10 \cdot I_{10} \quad (51)$$

angenommen werden, wobei I_n die $(n \times n)$ -Einheitsmatrix darstellt.

2 Auswertung

Mit den unter 1 erarbeiteten Gleichungen und Matrizen kann nun das QEKF implementiert werden. Dazu liegt eine Sammlung von Messwerten vor, um eine Simulation durchführen zu können. Abbildung 1 zeigt den Verlauf der Daten aller drei Sensoren, aufgeteilt auf die jeweils drei verschiedene Komponenten der Achsen. Die Implemen-

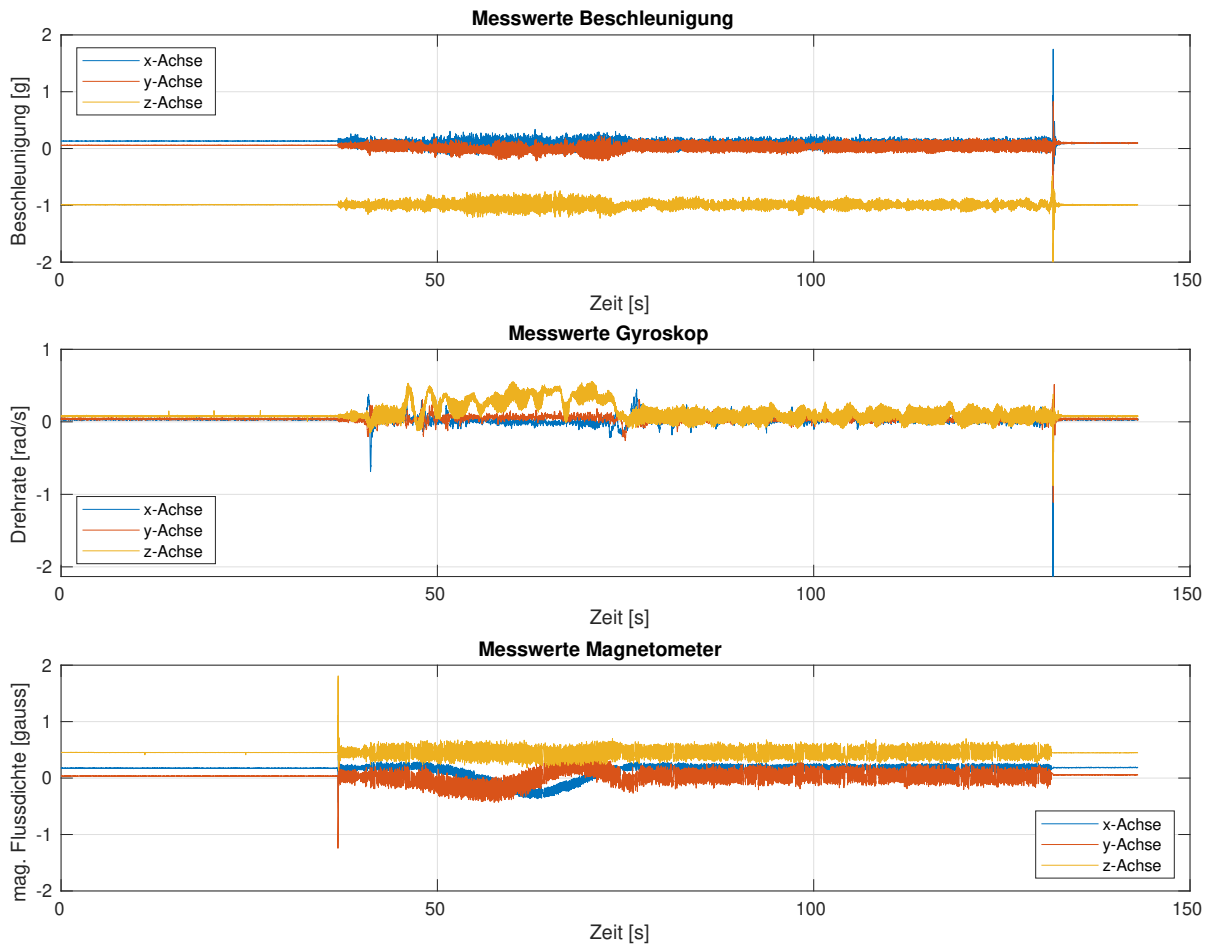


Abbildung 1: Grafische Aufbereitung der Messdaten des Beschleunigungssensors, Gyroskops und Magnetometers

tierung erfolgt wie gehabt in MATLAB, unterscheidet sich jedoch nur gering von bisherigen Implementierungen und wird hier nicht näher erläutert.

Die erste durchgeführte Simulation mit den unter 1.10 bestimmten Initialwerten ergibt die unter Abbildung 2 durchgeführten Ergebnisse. Das Filter zeigt in Abbildung 2 schon von Beginn an eine recht gute Näherung an den wahren Wert und folgt dem weiteren Verlauf mit minimalen Abweichungen. Der Fehlerausschlag im mittleren Teil beweist eine nicht optimale Dynamik in Bezug auf Änderungen: Das Filter ist also stets dem wahren Verlauf etwas hinterher. Im folgenden soll durch eine Variation verschiedener Werte in Q und R das Filterverhalten untersucht und gegebenenfalls auch verbessert werden.

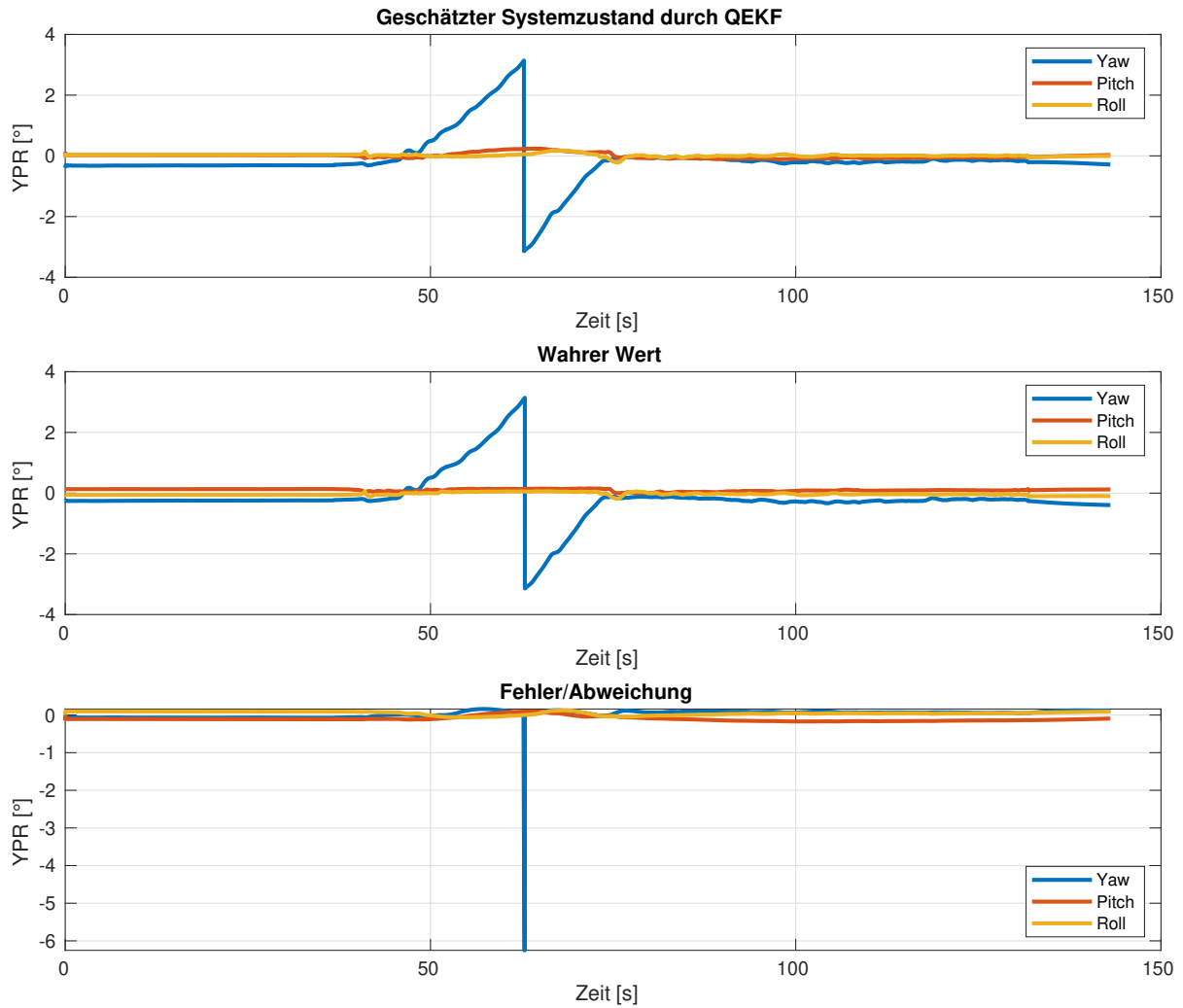


Abbildung 2: Ergebnisse der Simulation des QEFK mit Initialwerten in Q und R

Vorweg gilt im Allgemeinen zur Wahl von Q und R , dass große Werte in der Prozessrauschkovarianzmatrix und kleine Werte im Messrauschen ein dynamischeres Filterverhalten erzeugen: Das Filter reagiert also schneller auf Änderungen. Dagegen bewirkt die umgekehrte Einstellung – kleine Werte in Q und große in R –, dass Rauschen besser unterdrückt wird, der Verlauf also geglättet wird und weniger Ausreißer aufweist – auf Kosten der Dynamik.

Werden die Werte der Prozessrauschkovarianzmatrix erhöht ($10^6 \cdot Q$), wirkt sich dies im Filterprozess dahingehend aus, dass das Filter zwischen zwei aufeinanderfolgenden Berechnungsiterationen ein erhöhtes Rauschen des Systemzustands vermutet, das Zustandsraummodell betreffend der Systemeingänge (die von Q eingeordnet werden) also unzuverlässig oder nicht vollständig ist. Dies ist im Hinblick auf die Verknüpfung mit dem Systemeingang dadurch bedingt, dass höheren Werten in Q eine größere und niedrigeren eine geringere Unsicherheit in \vec{u} attestiert wird.

Die mit dieser erhöhten Prozessunsicherheit behafteten Größen fließen schließlich auch in die Berechnung des Lagequaternions mit ein. Abbildung 3 zeigt dieses Verhalten.

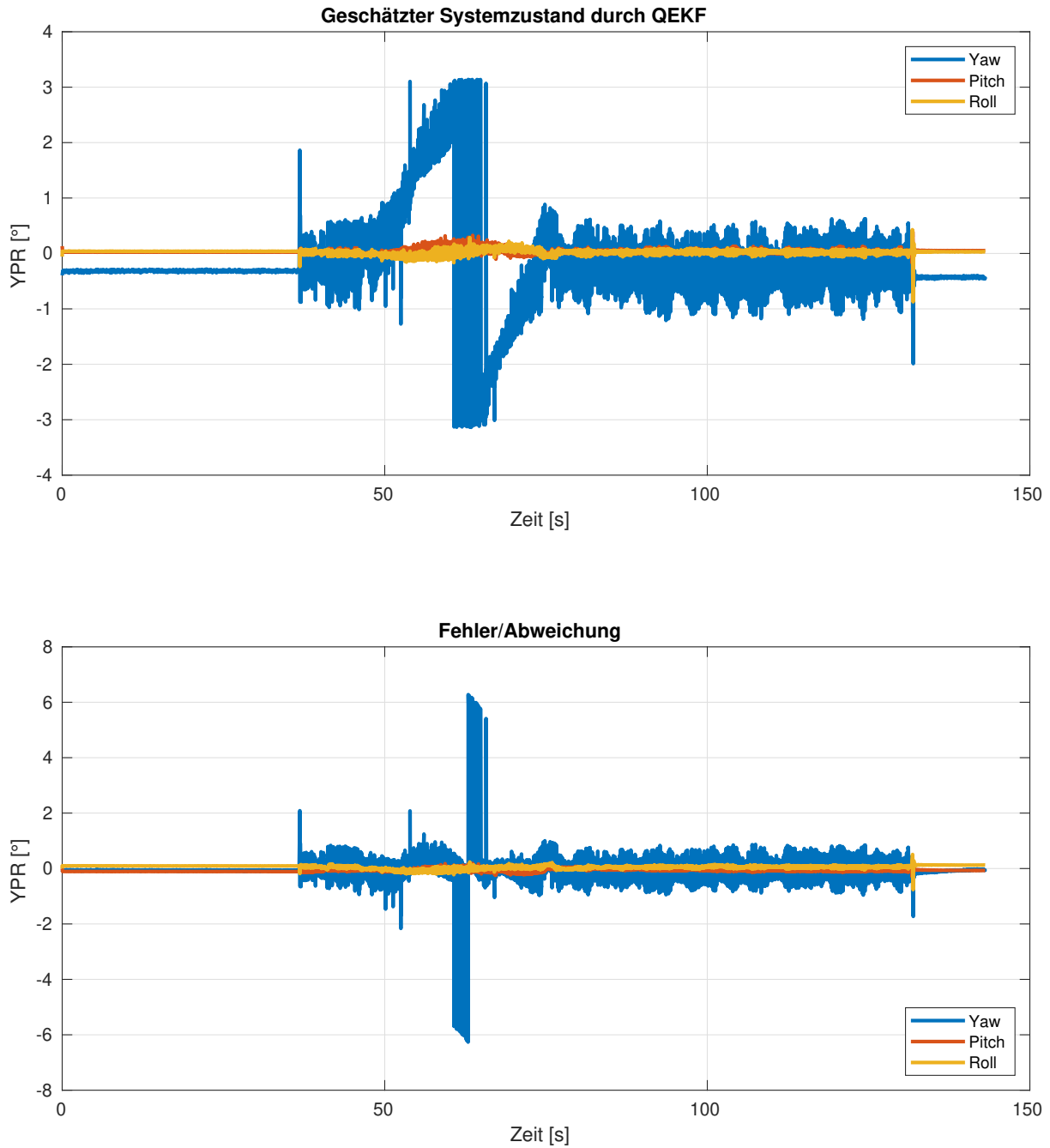


Abbildung 3: Filterverhalten und Abweichung bei hohen Werten in Q

Wie vermutet ist dabei erhöhtes Rauschen zu beobachten. Der Yaw-Winkel streut aus den o.g. Gründen und wegen der höheren Volatilität der Messwerte (siehe Abbildung 1, mittleres Bild, z-Achse) deutlich stärker, als Roll und Pitch.

Ähnlich sieht der Verlauf aus, wenn R niedrig gewählt wird ($10^{-6} \cdot R$). Abbildung 4 zeigt dieses Szenario. In diesem Fall wird die Güte der Messungen aus Beschleunigungssensor und Magnetometer vom Filter überschätzt, während das Prozessrauschen

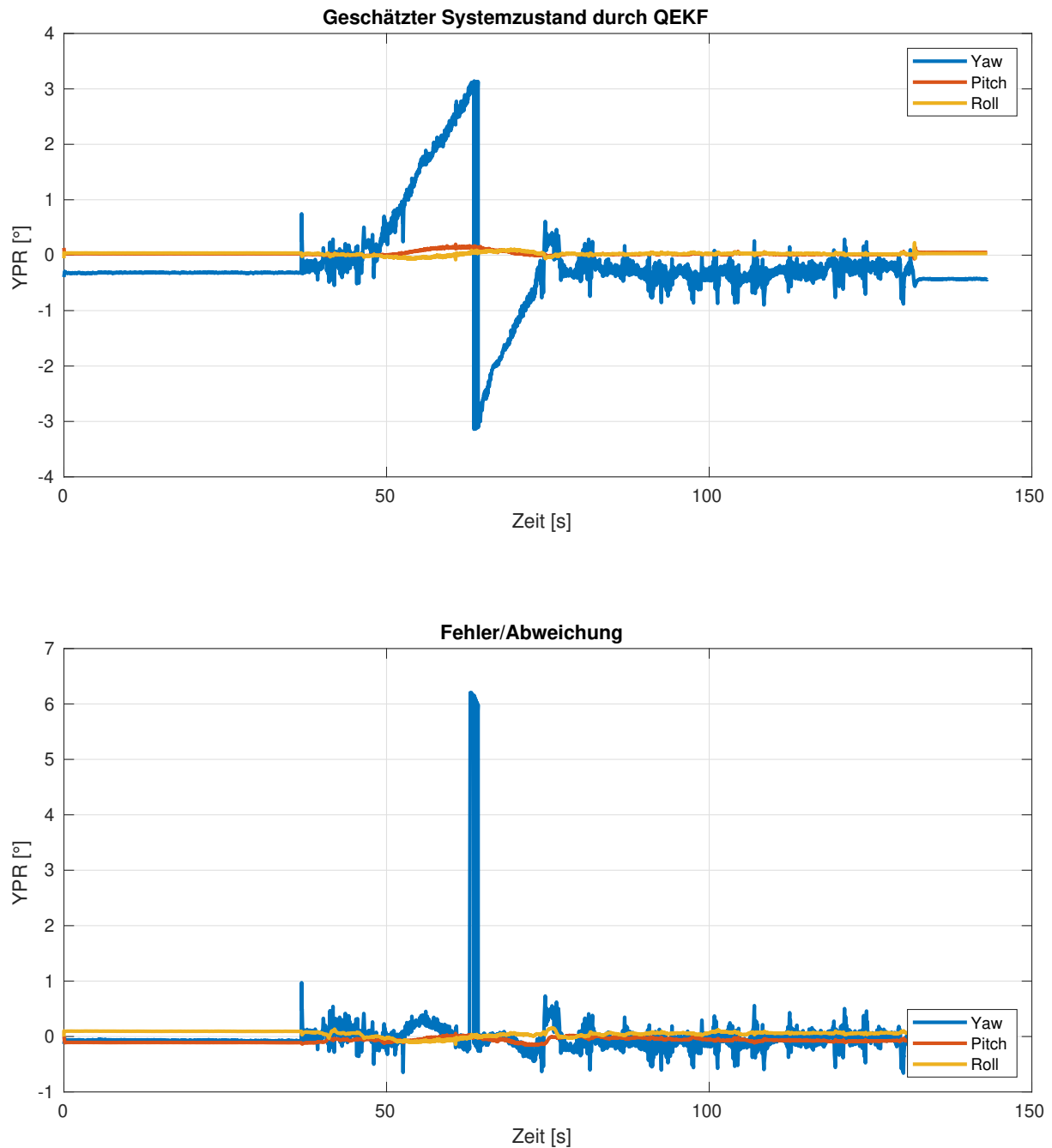


Abbildung 4: Filterverhalten und Abweichung bei niedrigen Werten in R

normal bleibt. Der normalwirkende Einfluss des Gyroskops mindert vermutlich bei gleicher Verstärkung von R das Rauschen im Vergleich zu Abbildung 3 etwas ab.

Abbildung 5 zeigt das Verhalten des Filters für hohe Werte in R ($10^6 \cdot R$). Damit wird dem Filter mitgeteilt, dass die Messwerte der beiden Sensoren Accelerometer und Magnetometer insofern unzuverlässig sind, als dass sie sehr breit streuen. Dementsprechend wird sich das Filter mehr auf das Zustandsraummodell mit Systemeingang

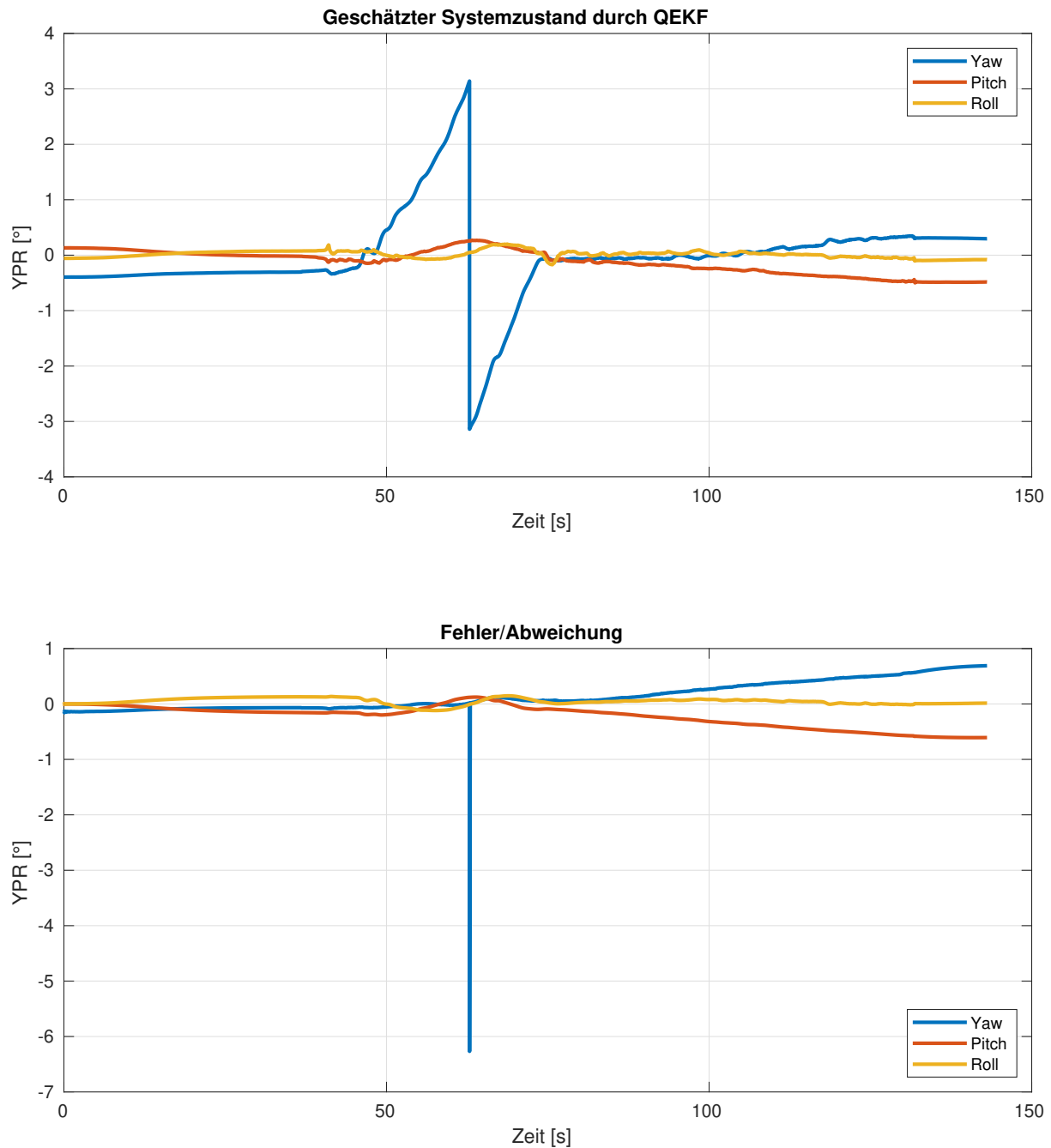


Abbildung 5: Filterverhalten und Abweichung bei hohen Werten in R

Gyroskop verlassen – welches von dieser Änderung nicht angesprochen wird. Auffällig ist in diesem Fall, dass im letzten Drittel der Simulation bei zwei Schätzkomponenten ein immer größer werdender Fehler aufsummiert wird. Die Ursache dafür könnte sein, dass mit dem Gyroskop lediglich eine Sensoreingabe mit annehmbarer Güte für den Filter zur Verfügung steht, während es die beiden anderen Sensoren mit sehr niedrigerer Gewichtung einfließen lässt, woraus eine Korrekturmöglichkeit entfällt, die früheren Implementierung zur Verfügung steht um Fehler zu korrigieren.