

Kamera-basierte Regelung und Sicherheitsüberwachung für den Quadrokopter „Parrot Mambo“

Als Masterarbeit
vorgelegt von

Lewin Buttazzo

Lehrstuhl für Regelungstechnik
Universität Erlangen-Nürnberg

Juni 2020

Kurzzusammenfassung

In dieser Arbeit wird eine kamerabasierte Flugregelung und eine Sicherheitsüberwachung für einen Quadrocopter entwickelt. Die Arbeit gliedert sich in drei maßgebliche Bestandteile. Alle Konzepte werden theoretisch erarbeitet und dann sowohl simulativ als auch physikalisch erprobt.

Im ersten Teil der Arbeit wird ein Zustandsmodell für die Dynamik des Quadrocopters kurz diskutiert und dann für den zeitdiskreten, linearen Reglerentwurf vorbereitet. Auf Basis dieses Zustandsmodells wird dann ein zeitinvarianter Linear Quadratic Regulator ausgelegt.

Im zweiten Teil der Arbeit wird eine Kameraauswertung entworfen. Diese soll mithilfe einer Bodenmarkierung zusätzliche Messwerte für die x - y -Position und den Gierwinkel des Quadrocopters liefern. Dann wird auf Basis der nun erweiterten Messwerte ein Zustandsbeobachter in Form eines Kalman-Filters ausgelegt.

Im dritten Teil der Arbeit wird eine Sicherheitsüberwachung für den Quadrocopter ausgelegt. Diese dient unter Anderem der sicheren Erprobung experimenteller Regler. Hierfür wird mithilfe einer Lyapunov-Funktion eine invariante Menge für den sicheren Zustandsraum der Quadrocopterdynamik bestimmt.

Masterarbeit

Kamera-basierte Regelung und Sicherheitsüberwachung für den Quadrocopter „Parrot Mambo“

Aufgabenstellung:

Am Lehrstuhl für Regelungstechnik wird ein Quadrocopter zu einem Demonstrator-system für Echtzeitregelungen weiterentwickelt. Hierzu sollen in dieser Arbeit zwei regelungstechnische Funktionen entworfen und auf dem kommerziell erhältlichen Quadrocopter „Parrot Mambo“ erprobt werden:

Erstens soll die integrierte Kamera dazu verwendet werden, die Position und die Lage des Quadrocopters zu bestimmen. Die gewonnenen Werte sind in die Regelung einzubeziehen, um ein Wegdriften bei Seitenwind zu verhindern. Hierzu ist, basierend auf der Modellbildung in der Vorarbeit von Runmu Yang, eine lineare Regelung in Zustandsraumstruktur zu entwerfen.

Zweitens soll ein virtueller Sicherheitszaun entwickelt werden, um die gefahrlose Erprobung neuer Regler zu ermöglichen. Rechtzeitig bevor der Quadrocopter den erlaubten Bereich verlässt, muss dies erkannt und auf die im ersten Schritt entworfene, konservativ ausgelegte Regelung zurückgeschaltet werden. Hierzu ist das auf elliptischen invarianten Mengen basierende Konzept aus der Vorarbeit von Quirin Apfel geeignet weiterzuentwickeln.

Die entwickelten Konzepte sollen simulativ und praktisch evaluiert werden. Bei der Umsetzung ist darauf zu achten, dass sie gut auf andere Quadrocopter übertragbar sind.

Es wird ausdrücklich auf die „Richtlinien für die Durchführung von Abschlussarbeiten“ hingewiesen.

(Prof. Dr.-Ing. habil. Günter Roppenecker)

Betreuung: Maximilian Gaukler, M.Sc.

Erklärung

Ich versichere, dass ich die vorliegende Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Erlangen, den 18.06.2020

Lewin Buttazzo

Inhaltsverzeichnis

1	Einleitung	1
2	Modellbildung	3
2.1	Modell auf Basis von Euler-Winkeln	3
2.2	Erweiterung des Zustandsraummodells	6
2.3	Zeitdiskretisierung des Zustandsraummodells	7
2.4	Berücksichtigung des Gierwinkels im linearisierten Modell	9
3	Zustandsreglerentwurf	11
3.1	Grundlagen zum Linear Quadratic Regulator	12
3.2	Auslegung des LQR	13
3.3	Regelkreisdynamik	14
3.4	Simulative Erprobung	18
3.5	Erprobung am Parrot Mambo	21
4	Kameraauswertung	27
4.1	Positionserkennung mittels Bodenmarkierung	27
4.2	Berechnung von ψ , x und y aus RGB-Daten	28
4.3	Arten der Ausreißerbehandlung	33
4.4	Präventive Ausreißerbehandlung	34
4.5	Berechnungsperiode	35
4.6	Erprobung des Algorithmus	36
5	Entwurf eines Zustandsbeobachters	39
5.1	Stochastische Grundlagen zum Kalman-Filter	40
5.2	Kalman-Filter für das lineare Zustandssystem	41
5.3	Auslegung des Kalman-Filters	44
5.4	Dynamik des über Beobachter geschlossenen Regelkreises	51
5.5	Simulative Erprobung	53

5.6	Erprobung am Parrot Mambo	54
6	Sicherheitsüberwachung	57
6.1	Dynamik des autonomen Systems	58
6.2	Definition eines zulässigen Zustandsraums	59
6.3	Lyapunov-Stabilitätskriterium im zeitdiskreten Fall	61
6.4	Berechnung des wiederherstellbaren Zustandsraums	63
6.5	Umschaltlogik	67
6.6	Simulative Erprobung	68
6.7	Erprobung am Parrot Mambo	71
7	Zusammenfassung und Ausblick	73
	Literaturverzeichnis	75
A	Verzeichnis der Formelzeichen	77
B	Simulationsmodell	81
C	Versuchsaufbau für den Flugtest mit dem Parrot Mambo	83
C.1	Modell für den Kopterflug	86
D	Videos Flugexperimente	87
E	Kameraauswertung	89
E.1	Offline-Farbanalyse	89
E.2	Kamerasimulation zur Erprobung des Auswertealgorithmus	89
E.3	Algorithmus zur Erkennung der Bodenmarkierung	90

Kapitel 1

Einleitung

Digitale Regelungssysteme werden mit wachsenden, modernen Anforderungen immer komplexer. Komponenten wie smarte Sensoren und verteilte Echtzeitsysteme führen zu nicht-deterministischem Timing der Reglersoftware. Dies kann Auswirkungen auf die Dynamik des Systems und somit den Reglerentwurf haben. Mit diesen Problemstellungen befasst sich das Projekt qronOS [KFG⁺16].

Um im Rahmen von qronOS entwickelte Konzepte an realen Systemen demonstrieren zu können, bedarf es eines physikalischen Versuchsaufbaus. Gefordert ist ein dynamisches System, das mittels eingebettetem Rechner geregelt werden soll. Der Quadrocopter ist dafür aus mehreren Gründen eine geeignete Wahl. Die Materialkosten eines Quadrocopter-Versuchsstands sind vergleichsweise niedrig und ein solcher Versuchsstand hat keinen großen Platzbedarf. Außerdem eignet sich die vereinfachte Zustandsraumdarstellung des Quadrocopters gut, um theoretische Konzepte zu übertragen, da es sich linearisieren lässt, eine moderate Ordnung aufweist und teilweise aus voneinander entkoppelbaren Teilsystemen besteht [Gre].

In dieser Arbeit wird also in Hinblick auf die Nutzung des Quadrocopters als Versuchsstand für weiterführende Konzepte eine Grundlage für dessen Flugregelung erarbeitet. Diese soll übersichtlich und leicht anpassbar sein. In Kapitel 2 wird zunächst ein vereinfachtes, lineares Zustandsraummodell der Ordnung 12 für die Flugkörperdynamik eingeführt und für den Reglerentwurf vorbereitet. Dann wird in Kapitel 3 ein linearer Zustandsregler entworfen, dessen Verhalten sich später einfach und gezielt abwandeln lässt, sodass sich die Konzepte leicht auf einen beliebigen anderen Quadrocopter übertragen lassen. Auch der Zustandsbeobachter wird in Kapitel 5 auf Basis eines

Kalman-Filterentwurfs ausgelegt, sodass er sich im Nachhinein einfach modifizieren lässt.

Um einen Flug in Innenbereichen zu vereinfachen, muss ein Positiondrift des Quadropters vermieden werden. Zu diesem Zweck wird in Kapitel 4 eine Kameraauswertung erarbeitet. Dabei soll ein Augenmerk auf die Recheneffizienz und die Nachvollziehbarkeit der zugrundeliegenden Algorithmen gelegt werden. Die Kameraauswertung soll mithilfe einer farbigen Bodenmarkierung dazu in der Lage sein, die Absolutposition und -verdrehung des Quadropters zu messen. Mithilfe dieser Messung kann der Quadropter eine stabile Position im Raum halten. Dies verringert den Platzbedarf des gesamten Versuchsaufbaus und das Risiko eines Absturzes.

Als weitere Maßnahme, um den Quadropter vor Sturzschäden zu schützen, wird in Kapitel 6 eine Sicherheitsüberwachung nach [Apf] und [SS99] ausgearbeitet. Diese dient dem Zweck, einen gegebenenfalls unerprobten Flugregler austesten zu können, ohne zu riskieren dass es zu einem Absturz kommt. Um dies zu erreichen, wird mithilfe einer Lyapunov-Funktion eine sichere Zustandsmenge ermittelt. Das Verbleiben innerhalb dieser invarianten Menge soll garantieren, dass der erprobte Regler unter Einhaltung definierter Schranken sowie der Stellsignalbegrenzung wieder in den stabilen Arbeitspunkt zurückkehren kann. Eine Umschaltung soll dann, rechtzeitig bevor die sichere Zustandsmenge verlassen werden würde, auf den erprobten Regler zurückschalten und den Quadropter vor einem Absturz bewahren.

Die geschilderten Konzepte werden zunächst theoretisch erarbeitet. Anschließend werden sie simulativ überprüft, und schließlich am physikalischen Quadroptersystem, dem kommerziell erhältlichen *Parrot Mambo* implementiert und optimiert.

Kapitel 2

Modellbildung

Im Folgenden wird ein im Schwebeflug linearisiertes Zustandsmodell des Quadropters (im Weiteren auch *Kopter*) eingeführt. Dieses wird um Integralzustände erweitert, die beim späteren Reglerentwurf stationäre Genauigkeit gewährleisten. Für den Reglerentwurf muss das bisher zeitkontinuierliche System anschließend diskretisiert werden. Zuletzt müssen noch Betrachtungen für die Gültigkeit des gewählten Arbeitspunkts in Bezug auf den späteren Reglerentwurf angestellt werden.

2.1 Modell auf Basis von Euler-Winkeln

Um das Flugverhalten des Kopters zu modellieren, wird eine Euler-Winkel-Darstellung nach [Cra] gewählt, bei der die Lage des Flugobjekts mithilfe von drei Raumwinkeln angegeben wird. Diese bezeichnen sich jeweils als Rollwinkel ϕ um die x -Achse, Nickwinkel θ um die y -Achse und Gierwinkel ψ um die z -Achse. Nick- und Rollwinkel, also diejenigen Winkel, die den Kopter aus seiner grenzstabilen Fluglage herausdrehen, werden im Weiteren auch als *Kippwinkel* bezeichnet. Bild 2.1 zeigt den Quadropters *Parrot Mambo* mit beschrifteten x -, y - und z -Achsen.

Ein Modell, das auf einer Eulerwinkel-Darstellung beruht, hat zum Nachteil, nicht im vollen Zustandsraum gültig zu sein. Es kommt bei der Wahl der Rotationsreihenfolge zu **Gier-Nick-Roll** für die Bestimmung der Rotationsmatrix für $\theta = \pm\pi$ bei der Division durch $\cos \theta$ zur Singularität in der Rotationsmatrix, auch als *Gimbal Lock* bezeichnet. Dies ist im Rahmen dieser Arbeit jedoch nicht von Bedeutung, da der Regler lediglich für einen im Arbeitspunkt linearisierten Bereich ausgelegt werden



Bild 2.1: Quadrokopter *Parrot Mambo* mit beschrifteten Achsen [Mat]

soll. Es ergibt sich während des Betriebs also aufgrund dieser Beschränkung ohnehin nur eine geringe Abweichung von der Schwebeposition. Nach [Yan] ergibt sich das Zustandsraummodell aus dem Zustandsvektor

$$X = [x \ y \ z \ \phi \ \theta \ \psi \ \dot{x} \ \dot{y} \ \dot{z} \ p \ q \ r]^T, \quad (2.1)$$

den Eingangsgrößen

$$U = [\tau_{tot} \ \tau_x \ \tau_y \ \tau_z]^T \quad (2.2)$$

und den nichtlinearen Zustandsgleichungen

$$\begin{aligned} \dot{x} &= \dot{x} \\ \dot{y} &= \dot{y} \\ \dot{z} &= \dot{z} \\ \dot{\phi} &= p + q \sin \phi \tan \theta + r \cos \phi \tan \theta \\ \dot{\theta} &= q \cos \phi - r \sin \phi \\ \dot{\psi} &= q \sin \phi / \cos \theta + r \cos \phi / \cos \theta \\ \ddot{x} &= -\frac{\tau_{tot}}{m} (\sin \psi \sin \phi + \cos \psi \sin \theta \cos \phi) \\ \ddot{y} &= -\frac{\tau_{tot}}{m} (-\cos \psi \sin \phi + \cos \phi \sin \theta \sin \phi) \\ \ddot{z} &= -\frac{\tau_{tot}}{m} (\cos \theta \cos \phi) + g \\ \dot{p} &= (\tau_x + J_y p q - J_z p q) / J_x \\ \dot{q} &= (\tau_y + J_x p q - J_z p q) / J_y \\ \dot{r} &= (\tau_z + J_x p q - J_y p q) / J_z \end{aligned} \quad (2.3)$$

Die Position $[x \ y \ z]^T$ und die Geschwindigkeit $[\dot{x} \ \dot{y} \ \dot{z}]^T$ sind im Weltkoordinatensystem definiert. Die Euler-Winkel $[\phi \ \theta \ \psi]^T$ sind ebenfalls im Weltkoordinatensystem definiert. Die Winkelgeschwindigkeiten $[p \ q \ r]^T$ sind im Körperkoordinatensystem definiert. Für kleine Werte von ϕ und θ gilt $[p \ q \ r]^T = [\dot{\phi} \ \dot{\theta} \ \dot{\psi}]^T \cdot \tau_{tot}$ bezeichnet den Gesamtschub des Motors, $[\tau_x \ \tau_y \ \tau_z]^T$ bezeichnet die Motormomente um jeweils x , y und z -Achse. m bezeichnet die Masse des Kopters, J_i bezeichnet das Massenträgheitsmoment um die entsprechende Hauptachse.

Dieses Zustandsmodell wird nun im Arbeitspunkt X_b linearisiert. Es gelten

$$X_b = [x_b \ y_b \ z_b \ 0 \ 0 \ \psi_b \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T \quad (2.4)$$

und

$$U_b = [\tau_{tot_b} \ \tau_{x_b} \ \tau_{y_b} \ \tau_{z_b}]^T. \quad (2.5)$$

Dieser Arbeitspunkt entspricht einer konstanten Schwebeposition des Kopters. Der Regler soll für beliebige Größen von x_b , y_b , z_b und ψ_b funktionieren. Das linearisierte Zustandsmodell ergibt sich in [Yan] mittels Taylor-Linearisierung zu

$$A = \begin{bmatrix} O_{6 \times 6} & I_{6 \times 6} \\ \Psi_b & O_{6 \times 6} \end{bmatrix}, B = \begin{bmatrix} O_{8 \times 4} \\ J^{-1} \end{bmatrix} \quad (2.6)$$

mit

$$\Psi_b = \begin{bmatrix} 0 & 0 & 0 & -g \sin \psi_b & -g \cos \psi_b & 0 \\ 0 & 0 & 0 & g \cos \psi_b & -g \sin \psi_b & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (2.7)$$

und

$$J = \begin{bmatrix} m & 0 & 0 & 0 \\ 0 & J_x & 0 & 0 \\ 0 & 0 & J_y & 0 \\ 0 & 0 & 0 & J_z \end{bmatrix} \quad (2.8)$$

und der Systemgleichung

$$\dot{X} = A(X - X_b) + B(U - U_b). \quad (2.9)$$

Es wird ersichtlich, dass das linearisierte Modell aus sechs Teilsystemen besteht, die bis auf eine Ausnahme voneinander entkoppelt sind. Lediglich die x - und y -Position sind jeweils über die Matrix Ψ_b von den Kippwinkeln abhängig.

Die üblichen Bezeichnungen mit Großbuchstaben X für den Systemzustand und U für den Systemeingang dienen zur Abgrenzung von der kartesischen Koordinate, die mit dem Kleinbuchstaben x gekennzeichnet ist.

2.2 Erweiterung des Zustandsraummodells

Würde man später einen *Linear Quadratic Regulator* (LQR) anhand des oben entwickelten Systems auslegen, so könnte dieser keine stationäre Genauigkeit gewährleisten. Grund dafür ist, dass ein LQR lediglich eine proportionale Zustandsrückführung umfasst.

Aus diesem Grund soll das System um vier Integralzustände erweitert werden, die jeweils die Zustandsfehler integrieren. Gleichung 2.10 beschreibt, wie die Integralzustände S gebildet werden. Diese Zustände sollen dann durch den LQR stabilisiert werden und sorgen entsprechend für stationäre Genauigkeit [Gre].

$$S = \int \begin{bmatrix} x - x_b \\ y - y_b \\ z - z_b \\ \psi - \psi_b \end{bmatrix} dt = \begin{bmatrix} s_x \\ s_y \\ s_z \\ s_\psi \end{bmatrix} \quad (2.10)$$

Für die Zustandsfehler der beiden übrigen Kippwinkel ϕ und θ werden keine Integrationszustände vorgesehen. Diese würden in linearer Abhängigkeit zu den Zuständen \dot{x} und \dot{y} stehen und somit die vollständige Steuerbarkeit kompromittieren. Dies lässt sich ebenfalls physikalisch motivieren: Sollte einer der Kippwinkel ungleich Null sein, so würde der Kopter in x - bzw. y -Richtung beschleunigen.

Die erweiterten Systemmatrizen ergeben sich wie folgt

$$A_{ext} = \begin{bmatrix} O_{6 \times 6} & I_{6 \times 6} & O_{6 \times 4} \\ \Psi_b & O_{6 \times 6} & O_{6 \times 4} \\ \tilde{A} & O_{4 \times 6} & O_{4 \times 4} \end{bmatrix}, B_{ext} = \begin{bmatrix} O_{8 \times 4} \\ J^{-1} \\ O_{4 \times 4} \end{bmatrix} \quad (2.11)$$

mit

$$\tilde{A} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.12)$$

Um zu verhindern, dass durch die Integralzustände in Zusammenspiel mit der Stellsignalbegrenzung des echten Systems ein Regler-Windup entsteht, wird in der Simulink-Implementierung eine *Anti-Regler-Windup-Maßnahme* gemäß [Gre] ergriffen. Sie unterbindet die Integration der Zustände, für den Fall, dass die Stellsignalbegrenzung über- bzw. unterschritten wird.

2.3 Zeitdiskretisierung des Zustandsraummodells

Der Reglerentwurf soll später im Zeitdiskreten stattfinden, sodass das Regelgesetz auf der Kopter-CPU berechnet werden kann. Um die Implementierung zu ermöglichen, wird daher das System zeitdiskretisiert und anschließend ein zeitdiskreter Regler bestimmt. Der Index d markiert im Weiteren eine zeitdiskrete Größe.

Als Abtastperiode wird $T_s = 5\text{ms}$ gewählt. Diese Wahl hat sich als schnell genug erwiesen, um die Lage des Kopters stabilisieren zu können ohne zu Laufzeitproblemen bei der Berechnung zu führen. Mit den Formeln für die Zeitdiskretisierung linearer

Zustandssysteme nach [Lun16]

$$A_d = e^{AT_s} \quad (2.13)$$

und

$$B_d = \int_0^{T_s} e^{A\tau} d\tau \cdot B \quad (2.14)$$

ergeben sich für die zeitdiskreten Systemmatrizen bei Vernachlässigung von Termen mit Ordnung höher als zwei:

$$A_d = I_{12 \times 12} + \begin{bmatrix} O_{6 \times 6} & T_s \cdot I_{6 \times 6} \\ O_{6 \times 6} & O_{6 \times 6} \end{bmatrix} + \begin{bmatrix} O_{2 \times 3} & \frac{1}{2}T_s^2\Psi_d & O_{2 \times 4} & O_{2 \times 2} & O_{2 \times 1} \\ O_{4 \times 3} & O_{4 \times 2} & O_{4 \times 4} & O_{4 \times 2} & O_{4 \times 1} \\ O_{2 \times 3} & T_s\Psi_d & O_{2 \times 4} & \frac{1}{2}T_s^2\Psi_d & O_{2 \times 1} \\ O_{4 \times 3} & O_{4 \times 2} & O_{4 \times 4} & O_{4 \times 2} & O_{4 \times 1} \end{bmatrix} \quad (2.15)$$

und

$$B_d = \begin{bmatrix} O_{2 \times 4} \\ \frac{1}{2}T_s^2 J^{-1} \\ O_{2 \times 4} \\ T_s J^{-1} \end{bmatrix} \quad (2.16)$$

mit

$$\Psi_d = \begin{bmatrix} -g \sin \psi_b & -g \cos \psi_b \\ g \cos \psi_b & -g \sin \psi_b \end{bmatrix}. \quad (2.17)$$

Für die zeitdiskrete Systemgleichung gilt

$$X_{k+1} = A_d X_k + B_d U_k, \quad (2.18)$$

wobei X_k den Zustandsvektor und U_k das Eingangssignal zum diskreten Zeitpunkt k beschreiben.

Man bemerke, dass der letzte Summand in Gleichung 2.15 die Verkopplung des x - y -Teilsystems mit dem Kippwinkel-Teilsystem beschreibt.

Die zeitdiskrete Form der Zustandserweiterung spielt für die Veranschaulichung im

Weiteren keine Rolle. Daher wurde hier übersichtshalber auf ihre Darstellung verzichtet.

2.4 Berücksichtigung des Gierwinkels im linearisierten Modell

In verwandten Arbeiten, wie [Yan] wird eine kaskadierte PID-Regelung angewendet. Dies ermöglicht, dass das Stellsignal, welches der Positionsregler an den Lageregler gibt, vorher in Abhängigkeit des Gierwinkels transformiert werden kann. Beispielsweise wird bei einem Gierwinkel von $\psi = \frac{\pi}{2}$ ein Stellsignal, das eine x -Abweichung in Weltkoordinaten vorgibt, zu einer y -Abweichung in Körperkoordinaten umgerechnet. Diese wird dann an den Lageregler weitergegeben. Damit kann gewährleistet werden, dass auch während des Betriebs der Gierwinkel frei wählbar ist.

Andere Werke, wie [Lan] verwenden eine LQR-Entwurfsmethode mit einer zeitinvarianten Systemmatrix $A(t)$. Dies ist in dieser Arbeit auf Grund von Rechenleistungsbeschränkung nicht möglich. Arbeiten wie [BNS04] linearisieren den Arbeitspunkt um mehrere Werte des Gierwinkels und wechseln dann zur Laufzeit zwischen einer diskreten Anzahl an möglichen Reglern.

Letzteres Konzept wird in dieser Arbeit erweitert. Es soll allerdings für jeden kontinuierlichen Arbeitspunkt ein angepasstes Regelgesetz realisiert werden. Für den Reglerentwurf wird dafür zunächst der Gierwinkel zu $\psi_b = 0$ fixiert. Damit ergibt sich mit

$$\Psi_{d,0} = \begin{bmatrix} 0 & -g \\ g & 0 \end{bmatrix} \quad (2.19)$$

eine somit konstante Systemmatrix $A_{d,ext,0}$, auf Basis derer der Reglerentwurf stattfinden soll. Es müssen dann die Fehler der Zustände im Weltkoordinatensystem $\begin{bmatrix} x_w & y_w \end{bmatrix}^T$ vor dem Regler mithilfe der Transformationsmatrix

$$T^{BW2} = \begin{bmatrix} \cos \psi & \sin \psi \\ -\sin \psi & \cos \psi \end{bmatrix} \quad (2.20)$$

in das um ψ verdrehte Körperkoordinatensystem transformiert werden. Dies stellt sicher, dass ein von 0 abweichendes ψ_b durch den Regler entsprechend respektiert wird. Selbige Transformation muss auch auf den entsprechenden Geschwindigkeitsvektor $\begin{bmatrix} \dot{x}_w & \dot{y}_w \end{bmatrix}^T$ sowie den Vektor der im Weltkoordinatensystem integrierten Zustandsfehler $\begin{bmatrix} s_{x,w} & s_{y,w} \end{bmatrix}^T$ angewendet werden. Es ergibt sich insgesamt eine Transformation des gesamten Zustandsfehlers der Form

$$E_b = T^{BW16} E_w, \quad (2.21)$$

wobei E_b den Vektor des Zustandsfehlers in Körperkoordinaten und E_w den Vektor des Zustandsfehlers in Weltkoordinaten bezeichnet. T^{BW16} ist die Matrix, die den gesamten Vektor transformiert. Da die sonstigen Zustände belassen werden sollen, ergibt sich diese Matrix zu

$$T^{BW16} = \begin{bmatrix} T^{BW2} & 0 & 0 & 0 & 0 & 0 \\ 0 & I_{4 \times 4} & 0 & 0 & 0 & 0 \\ 0 & 0 & T^{BW2} & 0 & 0 & 0 \\ 0 & 0 & 0 & I_{4 \times 4} & 0 & 0 \\ 0 & 0 & 0 & 0 & T^{BW2} & 0 \\ 0 & 0 & 0 & 0 & 0 & I_{2 \times 2} \end{bmatrix}. \quad (2.22)$$

Die Gültigkeit und Auswirkungen dieser Maßnahme in Bezug auf die Regler- und Beobachterfunktionalität werden später in den entsprechenden Abschnitten 3.3 und 5.4 analysiert und diskutiert.

Kapitel 3

Zustandsreglerentwurf

Anhand des in Kapitel 2 entwickelten zeitdiskreten Zustandsmodells soll nun eine lineare Zustandsregelung entworfen werden. Der Übergang von der kaskadierten PID-Regelung aus [Yan] zu einer Zustandsregelung erhöht die Übersichtlichkeit und Modularität des Systems. Damit wird es leichter, später das gesamte Systemverhalten zu beschreiben. Außerdem sind spätere Anpassungen einfacher realisierbar.

Der Reglerentwurf soll einen zeitdiskreten, zeitinvarianten Linear Quadratic Regulator (im Weiteren LQR) zum Ergebnis haben. Der Einsatz eines solchen Reglers führt zu einer hinsichtlich einer Kostenfunktion optimalen Regelung in einem um den Arbeitspunkt linearen Bereich, unter der Bedingung, dass die Stellsignalbegrenzung eingehalten wird [Lun16].

Zunächst werden theoretische Grundlagen zur Entwurfsmethodik erörtert. Anschließend wird schrittweise dargelegt, wie der Reglerentwurf im vorliegenden Fall erfolgt ist. Danach wird die Regelkreisdynamik betrachtet. Hierbei gilt es zusätzlich, die in Abschnitt 2.4 angesprochene Gierwinkelproblematik zu analysieren. Zuletzt werden die entwickelten Konzepte sowohl simulativ als auch am physikalischen System *Parrot Mambo* erprobt.

3.1 Grundlagen zum Linear Quadratic Regulator

Ein zeitdiskreter LQR optimiert nach [Lun16] eine Kostenfunktion der Form

$$J_c = \sum_{k=1}^{\infty} (X_k^T Q X_k + U_k^T R U_k). \quad (3.1)$$

Diese Kostenfunktion summiert einen Term vom Zeitpunkt $k = 0$ bis ∞ , der quadratisch von den Eingangsgrößen U sowie den Zustandsgrößen X abhängt. Man spricht von einer Kostenfunktion über einen *unendlichen Zeithorizont*. Minimiert man diese Funktion, findet man demnach einen Kompromiss zwischen den einzelnen Größen der Vektoren U und X . Die Matrizen Q und R enthalten dabei Gewichtungparameter, die beim Entwurf vorgegeben werden können. Sie beeinflussen, wie groß der Stellenwert der jeweiligen Größe aus X oder U für die Kostenfunktion sein soll.

Die Kostenfunktion kann durch Lösen der algebraischen *Ricatti-Gleichung* für den zeitdiskreten Fall

$$A_d^T P A_d - (A_d^T P B_d)(B_d^T P B_d + R)^{-1}(A_d^T P B_d)^T + Q = 0, \quad (3.2)$$

und der linearen Rückführung des Zustands über den sich aus P ergebenden Reglers

$$K = (R + B_d^T P B_d)^{-1} B_d^T P A_d \quad (3.3)$$

minimiert werden. Daraus ergibt sich K als im Arbeitspunkt optimaler Regler als Funktion der vorzugebenden Kostenmatrizen R und Q sowie der Systemmatrizen des zeitdiskreten Systems A_d und B_d . P ist eine konstante, positiv semidefinite Matrix. Sie dient lediglich der iterativen Entwicklung einer geschlossenen Lösung und hat keine konzeptuelle Bedeutung.

Ein **zeitvarianter** LQR würde zu jedem Zeitschritt eine ähnliche Ricatti-Gleichung auswerten, um die aktuellen Zustands- und Eingangsgrößen sowie zeitvariante Systemmatrizen mit einzubeziehen. Dies würde weitaus mehr Rechenressourcen benötigen, als dem Quadropter zur Verfügung stehen. Ein solcher zeitvarianter LQR kommt beispielsweise in [Lan] zum Einsatz.

Für einen **zeitinvarianten** LQR hingegen genügt es, Gleichung 3.2 für einen definierten Arbeitspunkt zu lösen. Dies geschieht offline während des Reglerentwurfs. Daher

wird für diese Arbeit ein zeitinvarianter LQR ausgelegt.

3.2 Auslegung des LQR

Die Entwurfsmethode, die dem LQR zugrunde liegt, erlaubt es, Kosten für die einzelnen Zustände und Eingangsgrößen zu definieren. Insbesondere für größere Systeme ist dieser Freiheitsgrad als Entwurfparameter intuitiver als beispielsweise eine Eigenwertvorgabe, die ebenfalls eine proportionale Zustandsrückführung zum Ergebnis hätte. Ein iteratives Anpassen der Parameter mit anschließender Evaluation hat sich als zielführend herausgestellt.

Die Kostenmatrizen $R \in \mathbb{R}^{n \times n}$ und $Q \in \mathbb{R}^{m \times m}$ müssen symmetrisch sein, außerdem muss $R > 0$ sowie $Q \geq 0$ gelten. Die Größe der Matrizen ergibt sich jeweils aus der Anzahl der Zustände n bzw. der Anzahl der Eingangsgrößen m [Lun16].

In der Regel werden lediglich die Hauptdiagonalen der Kostenmatrizen gefüllt. Die Einträge der Matrizen spiegeln wider, wie kostspielig die Abweichung der entsprechenden Zustands- bzw. Eingangsgröße ist. So wird beispielweise einem Zustand, der vergleichsweise wenig systemkritisch ist, ein geringerer Wert in der Kostenmatrix zugeordnet.

Ein Anpassen des Flugverhaltens kann demnach gezielt erfolgen, indem ein unzufriedenstellender Wert modifiziert wird. Wenn beispielsweise also beobachtet wird, dass das System einen Fehler in x -Koordinaten zu langsam ausgleicht, so kann dieses Verhalten durch Erhöhung des Eintrags $Q_{1,1}$ in der Kostenmatrix verschnellert werden. Dies führt schlussendlich zu dem gewünschten Gesamtverhalten.

Als Ausgangspunkt für das Vorgehen wurden die Kostenmatrizen zunächst derart gewählt, sodass sich ein LQR ergibt, dessen Werte ungefähr dem in [Yan] ausgelegten Regler entsprechen. Folgende Kostenmatrizen haben sich aus dem iterativen Testprozess für das linearisierte Koptermodell ergeben:

$$Q_R = \text{diag}(80, 80, 10^4, 10^2, 10^2, 1, 10, 10, 10^3, 50, 50, 10^2, 1, 1, 1, 10^3) \quad (3.4)$$

$$R_R = \text{diag}(2 \cdot 10^3, 10^7, 10^7, 10^8) \quad (3.5)$$

Die zeitdiskrete, algebraische Ricatti-Gleichung aus Gleichung 3.2 kann mittels des MATLAB-Befehls `idare(Ad, Bd, QR, RR)` gelöst werden. Der Regler ergibt sich für dann aus der Lösung P gemäß Gleichung 3.3.

Die Reglermatrix für die gewählten Entwurfparameter errechnet sich zu

$$K = \begin{bmatrix} 0 & 0 & -0.003 & 0 \\ 0 & 0.003 & 0 & 0 \\ 2.4 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 \\ 0 & 0 & 0.01 & 0 \\ 0 & 0 & 0 & 0.0005 \\ 0 & 0 & -0.003 & 0 \\ 0 & 0.003 & 0 & 0 \\ 0.9 & 0 & 0 & 0 \\ 0 & 0.002 & 0 & 0 \\ 0 & 0 & 0.002 & 0 \\ 0 & 0 & 0 & 0.001 \\ 0 & 0 & -0.00035 & 0 \\ 0 & 0.0003 & 0 & 0 \\ 0.7 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.0001 \end{bmatrix}^T \quad (3.6)$$

3.3 Regelkreisdynamik

Wie in Abschnitt 2.4 bereits erläutert wurde, ist die Dynamikmatrix A_{ext} vom Gierwinkel ψ_b im Arbeitspunkt abhängig. Daher muss die Abweichung des Gierwinkels anderweitig berücksichtigt werden. Dies geschieht durch eine Transformation der Basis des Zustandsfehlervektors $E_b = T^{BW16} E_w$. Der Teil des Zustandsfehler mit $x - y$ -Bezug wird statt in Weltkoordinaten E_w dann in Körperkoordinaten E_b angegeben. Ein veranschaulichendes Blockschaltbild zeigt Bild 3.1. $X_{w,s,k}$ bezeichnet dabei die Sollgröße in Weltkoordinaten zum Zeitpunkt k .

Für alle folgenden Dynamikbetrachtungen muss ψ_b als über lange Zeiträume konstant angenommen werden, da andernfalls die Linearitätseigenschaft des Regelkreises verloren gehen würde. Ein Arbeitspunktwechsel in ψ_b muss unter gemäßigten Bedingungen

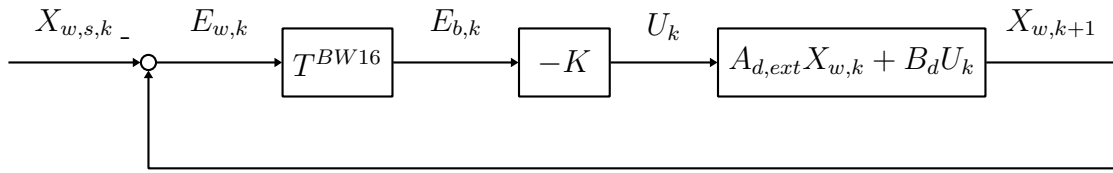


Bild 3.1: Blockschaltbild für die Regelkreisdynamik

vonstatten gehen. Diese Annahme hat sich für konservative Flugmanöver als tragbar herausgestellt.

Die vom Gierwinkel ψ_b abhängige Dynamikmatrix $A_{d,ext}$ bezieht sich auf den Zustand X_w im Weltkoordinatensystem. Für den späteren Beobachterentwurf gilt es somit zu sicherzustellen, dass der rekonstruierte Zustandsvektor für die $x - y$ -bezogenen Zustände im Weltkoordinatensystem definiert ist. Für die Betrachtung der Dynamik des geschlossenen Regelkreises wird zunächst angenommen, dass der Zustandsvektor X_w vollständig vorliegt.

Für die Dynamik des Zustandsfehlers ergibt sich aus Bild 3.1:

$$E_{w,k+1} = (A_{d,ext}(\psi_b) - B_d K T^{BW16}) E_{w,k}. \quad (3.7)$$

Es wird klar, dass die Dynamik des Zustandsfehler zunächst abhängig vom gewählten Gierwinkel ist. Um die Fehlerdynamik für weitere Betrachtungen zu vereinfachen, soll gezeigt werden, dass sich diese für beliebiges ψ_b nur unmaßgeblich von der Dynamik mit $\psi_b = 0$ unterscheidet.

Zunächst wird festgestellt, dass sich die Dynamik lediglich für die $x - y$ -bezogenen Zustandsfehler für $x, y, \dot{x}, \dot{y}, s_x, s_y$ sowie für die Kippwinkelgrößen θ, ϕ, p, q unterscheiden kann. Die Dynamiken der restlichen Zustandsfehler sind von diesem Teilsystem entkoppelt und somit durch ein abweichendes ψ_b unbeeinflusst. Es genügt also, sich auf die Fehlerdynamik des Systems

$$E_{dyn,k+1} = (A_{dyn} - B_{dyn} K_{dyn} T^{BW8}) E_{dyn,k} \quad (3.8)$$

mit

$$A_{dyn} = I_{8 \times 8} + \begin{bmatrix} O_{4 \times 4} & T_s \cdot I_{4 \times 4} \\ O_{4 \times 4} & O_{4 \times 4} \end{bmatrix} + \begin{bmatrix} O_{2 \times 2} & \frac{1}{2}T_s^2\Psi_d & O_{2 \times 2} & O_{2 \times 2} \\ O_{2 \times 2} & O_{2 \times 2} & O_{2 \times 2} & O_{2 \times 2} \\ O_{2 \times 2} & T_s\Psi_d & O_{2 \times 2} & \frac{1}{2}T_s^2\Psi_d \\ O_{2 \times 2} & O_{2 \times 2} & O_{2 \times 2} & O_{2 \times 2} \end{bmatrix}, \quad (3.9)$$

$$T^{BW8} = \begin{bmatrix} T^{BW2} & 0 & 0 & 0 \\ 0 & I_{2 \times 2} & 0 & 0 \\ 0 & 0 & T^{BW2} & 0 \\ 0 & 0 & 0 & I_{2 \times 2} \end{bmatrix} = T^{WB8,T} \quad (3.10)$$

und $E_{dyn} = X_{dyn} - X_{s,dyn}, X_{dyn} = [x \ y \ \phi \ \theta \ \dot{x} \ \dot{y} \ p \ q]$ zu beschränken. Zur Übersichtlichkeit der Betrachtung werden die erweiterten Systemzustände nicht berücksichtigt, die Argumentationsweise lässt sich auf diese problemlos erweitern.

Eine Zustandstransformation des Zustandsvektors zu

$$E_{trans} = T^{WB8} E_{dyn} \quad (3.11)$$

führt mit Einsetzen von E_{trans} in die Dynamikgleichung 3.8 zu

$$T^{WB8} E_{dyn,k+1} = (A_{dyn} - B_{dyn}K_{dyn}T^{BW8})T^{WB8} E_{dyn,k} \quad (3.12)$$

und nach linksseitiger Multiplikation beider Seiten mit der inversen bzw. transponierten Matrix T^{BW8} zu

$$E_{dyn,k+1} = (T^{BW8}A_{dyn}T^{WB8} - T^{BW8}B_{dyn}K_{dyn})E_{dyn,k}. \quad (3.13)$$

Es stellt sich heraus, dass die Matrix $A_{dyn,0} = T^{BW8}A_{dyn}T^{WB8}$ nicht vom Gierwinkel ψ abhängig ist. Für sie gilt

$$A_{dyn,0} = I_{8 \times 8} + \begin{bmatrix} O_{4 \times 4} & T_s \cdot I_{4 \times 4} \\ O_{4 \times 4} & O_{4 \times 4} \end{bmatrix} + \begin{bmatrix} O_{2 \times 2} & \frac{1}{2}T_s^2\Psi_{d,0} & O_{2 \times 2} & O_{2 \times 2} \\ O_{2 \times 2} & O_{2 \times 2} & O_{2 \times 2} & O_{2 \times 2} \\ O_{2 \times 2} & T_s\Psi_{d,0} & O_{2 \times 2} & \frac{1}{2}T_s^2\Psi_{d,0} \\ O_{2 \times 2} & O_{2 \times 2} & O_{2 \times 2} & O_{2 \times 2} \end{bmatrix} \quad (3.14)$$

mit

$$\Psi_{d,0} = \begin{bmatrix} 0 & -g \\ g & 0 \end{bmatrix}. \quad (3.15)$$

Der verbleibende, Gierwinkelabhängige Term $-T^{BW8}B_{dyn}K_{dyn}$ wird im Weiteren mit einem Gedankenexperiment analysiert, da sich K nur schwer analytisch zerlegen lässt.

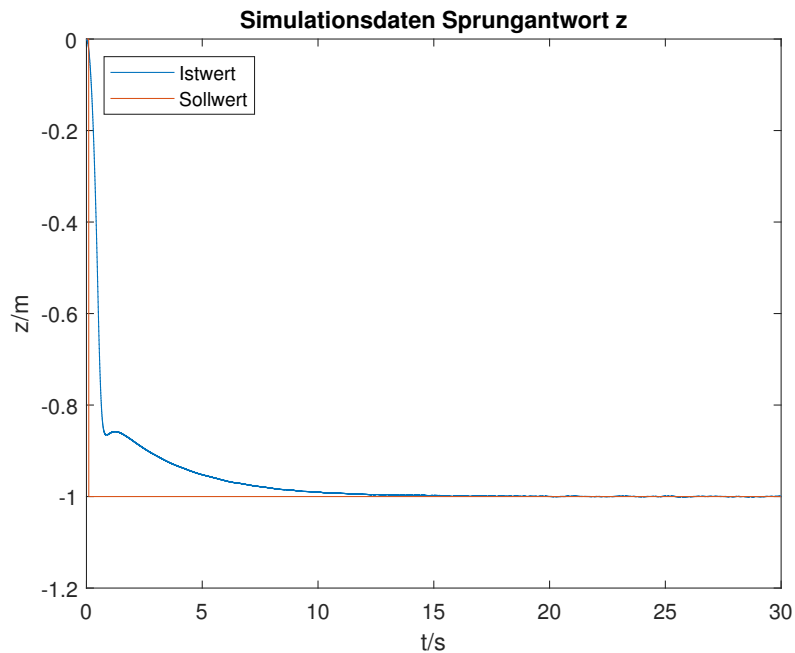
Man etabliere die Näherung, dass die Massenträgheitsmomente für die x- und die y-Hauptachse annäherungsweise gleich sind, also $J_x \approx J_y$. Da die jeweils für die x- und die y-Achse des Kopters analogen Werte für die Kostenmatrizen Q und R identisch gewählt wurden, lässt sich leicht sehen, dass die Dynamik für alle $\psi_b = \frac{\pi}{2} \cdot n, \forall n \in \mathbb{Z}$ gleich sein muss. Anschaulich verändert sich die Dynamik nicht, wenn man x- und y-Achse des Körperkoordinatensystems beliebig vertauscht oder invertiert.

Für $\psi_b \neq \frac{\pi}{2} \cdot n, \forall n \in \mathbb{Z}$, also für alle Werte, die dazwischen liegen, muss berücksichtigt werden, dass sich die Aufteilung des Zustandsfehlers auf jeweils x- und y-Achse nun ändert. Für $\psi_b = 45^\circ$ hat dies beispielsweise zur Folge, dass sich ein Zustandsfehler in ursprünglich reiner x -Richtung mit einem Faktor von $\frac{1}{\sqrt{2}}$ in sowohl x - als auch y -Richtung aufteilt. Bei gleicher Dynamik in beide Richtungen wirkt sich dies jedoch nicht auf das Flugverhalten aus.

Die Fehlerdynamik des geschlossenen Regelkreises kann also durch eine von ψ_b unabhängige Form angenähert werden. Für eine Wahl von $\psi_b = 0$ ergibt sich die Fehlerdynamik

$$E_{w,k+1} = (A_{d,ext,0} - B_d K) E_{w,k}. \quad (3.16)$$

Simulativ hat sich ebenfalls herausgestellt, dass der Zustandsfehler E_w im obigen Regelkreis für beliebige Wahl des Arbeitspunktes asymptotisch stabil ist. Der Kopter kann also mit dem für einen festen Arbeitspunkt ausgelegten LQR mithilfe der Koordinatensystemtransformation des Zustandsfehlervektors jeden beliebigen Gierwinkel ψ_b anfliegen. Dies hat sich auch in der Simulation und der physikalischen Erprobung gezeigt.

Bild 3.2: Simulation des mit LQR geregelten Systems, z

3.4 Simulative Erprobung

Zustandsgröße	Anfangswert	Endwert	Zeitpunkt
x	0	1	12
y	0	1	19
z	0	-1	0.1
ψ	0	$\frac{\pi}{4}$	5

Tabelle 3.1: Sollgrößen Sprünge zur simulativen Erprobung des geregelten Systems

Um die Herangehensweise und Implementierung des Reglers zunächst zu validieren, wird ein vereinfachtes Systemmodell konzipiert. Dieses modelliert das linearisierte System zuzüglich Stellsignalbegrenzung, Modellunsicherheit, Systemrauschen und Gierwinkelabhängigkeit der Dynamikmatrix. Dieses Simulationsmodell wird in den folgenden Kapiteln entsprechend erweitert. Das gesamte Modell befindet sich in Anhang B. Unter Verwendung des LQRs und Anti-Windup-Schaltkreis ergeben sich für die Sollgrößen sprünge nach Tabelle 3.1 die Simulationsergebnisse in Bildern 3.2 bis 3.6. Die Simulationsergebnisse zeigen, dass der LQR in der Lage ist, das System zu stabilisieren sowie Sollgrößen sprünge für x , y , z und ψ mit stationärer Genauigkeit zu realisieren. Der von Null abweichende Gierwinkel führt in der Simulation nicht zu

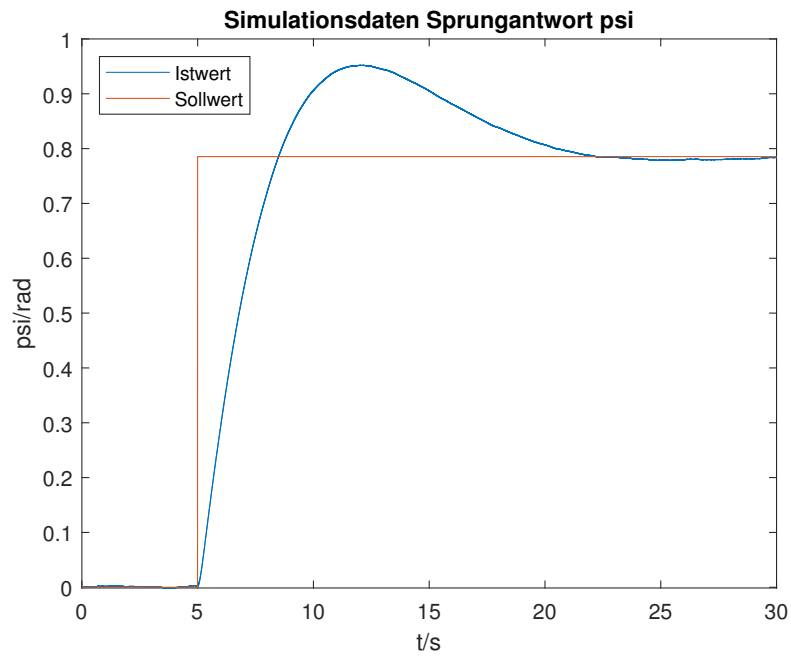
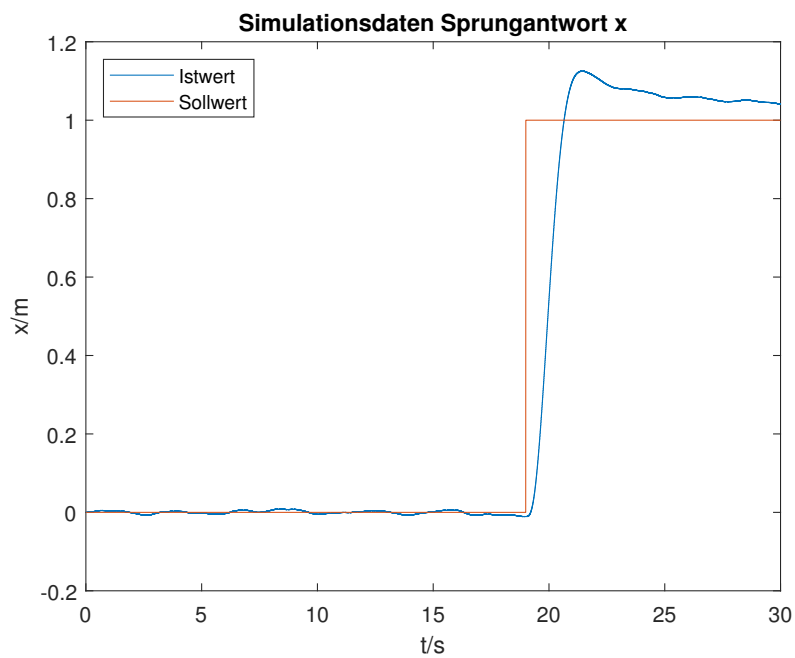
Bild 3.3: Simulation des mit LQR geregelten Systems, ψ 

Bild 3.4: Simulation des mit LQR geregelten Systems, x

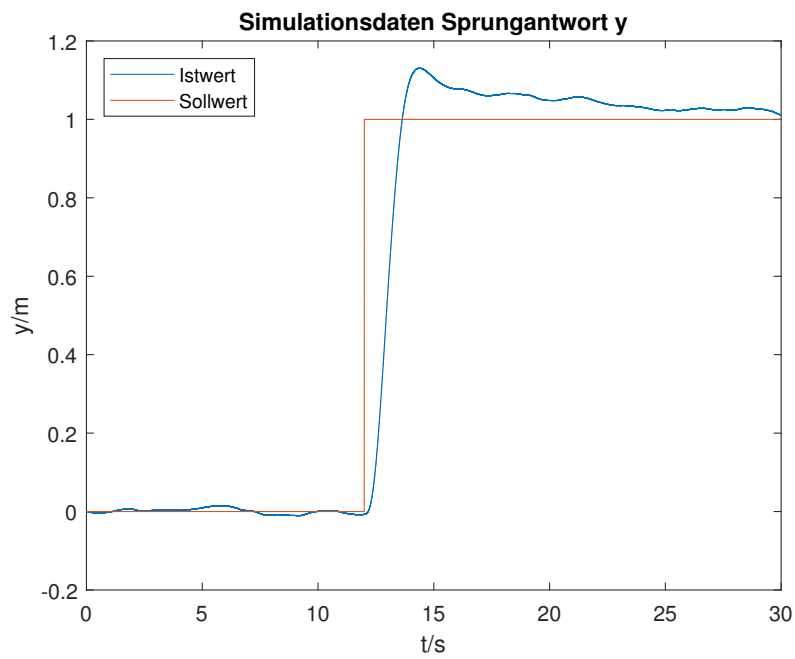


Bild 3.5: Simulation des mit LQR geregelten Systems, y

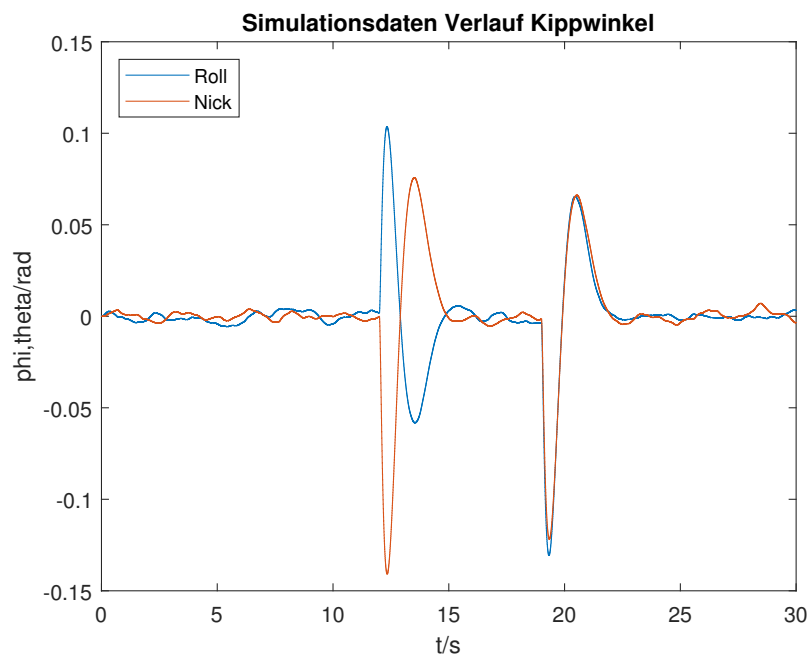


Bild 3.6: Simulation des mit LQR geregelten Systems, ϕ, θ

Problemen. Die im Weltkoordinaten vorgegebenen Sollgrößen werden korrekt in das Körperkoordinatensystem umgerechnet, wie in Bild 3.6 an den Ausschlägen jeweils beider Kippwinkel für jeden Sollwertsprung in x und y ersichtlich ist.

Die Sprungantwort erreicht für x in Bild 3.4 und y in Bild 3.5 nur sehr langsam ihren Endwert. Dies liegt in der Tatsache begründet, dass der LQR für diese Größen sehr konservativ eingestellt wurde. Eine aggressivere Einstellung hat in Tests am echten System zu einer unzufriedenstellenden Lageregelung geführt.

3.5 Erprobung am Parrot Mambo

Zustandsgröße	Anfangswert	Endwert	Zeitpunkt
x	0	1	17
y	0	1	12
z	0	-1	0
ψ	0	$\frac{\pi}{4}$	5

Tabelle 3.2: Sollwertsprünge zur Erprobung des geregelten Systems am Quadrokopter

Für die Erprobung am physikalischen System soll der Quadrokopter *Parrot Mambo* verwendet werden. Die erarbeiteten Konzepte lassen sich später leicht auf einen anderen Kopter übertragen. Ein detaillierter Versuchsaufbau wird in Anhang C geschildert.

Zur Erprobung des Reglers sollen Sprünge der Sollgrößen entsprechend Tabelle 3.2 für jeweils x , y , z und ψ auf das echte System gegeben werden. Als Zustandsbeobachter wird zunächst ein bereits existierender Beobachter eingesetzt, der in [Yan] entwickelt wurde.

Die Diagramme in Bildern 3.7 bis 3.11 zeigen die Verläufe der einzelnen Zustandsgrößen in Reaktion auf die vorgegebenen Sprünge. Eine Videoaufnahme für diesen Testflug findet sich in Anhang D. Die Ergebnisse zeigen, dass das echte System durch den entworfenen LQR stabilisiert werden kann. Der Regler folgt dem Sollwertsprung.

Der Sollwert der Flughöhe z in Bild 3.7 wird vergleichsweise langsam erreicht. Dies ist durch den Störeinfluss der Schwerkraft begründet. In dieser Arbeit wird die Störung durch die Erweiterung des Zustandsmodell um Integralzustände ausgeglichen. Die naheliegende Lösung, die entsprechenden Kostenparameter so anzupassen, dass sich die

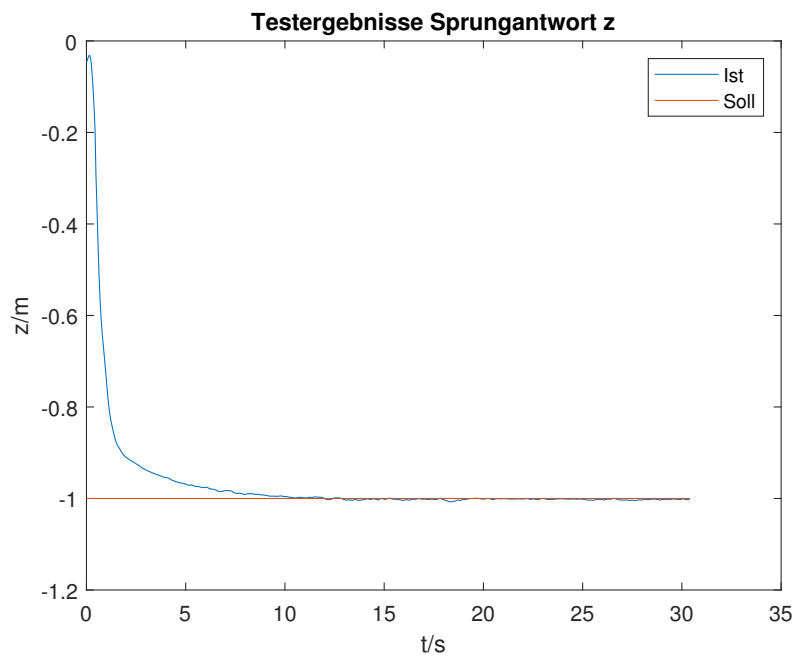


Bild 3.7: Experimentelle Erprobung des mit LQR geregelten Systems, z

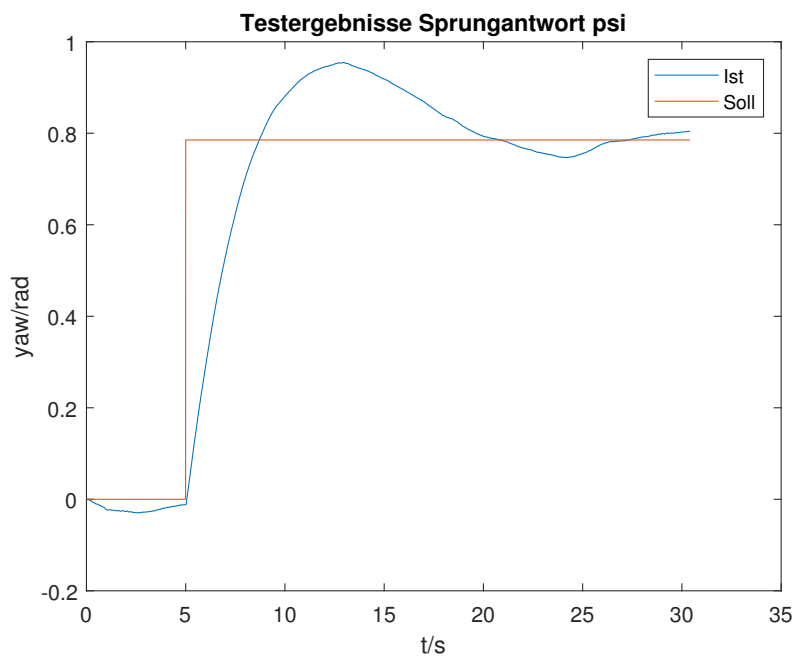


Bild 3.8: Experimentelle Erprobung des mit LQR geregelten Systems, ψ

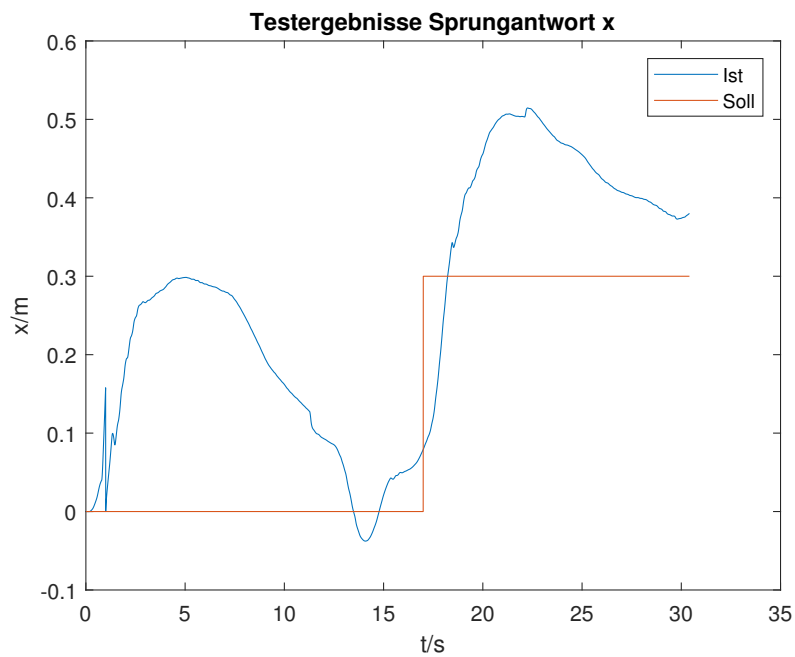


Bild 3.9: Experimentelle Erprobung des mit LQR geregelten Systems, x

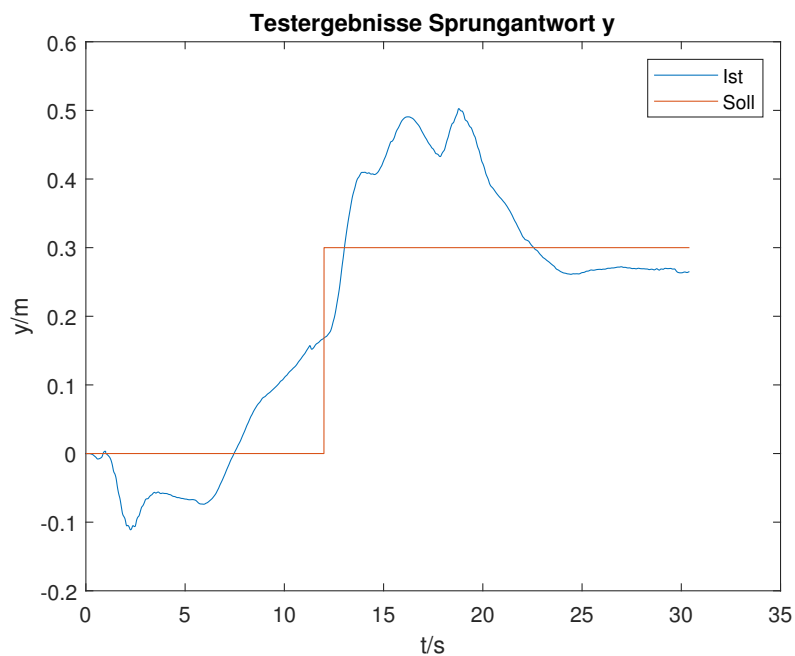


Bild 3.10: Experimentelle Erprobung des mit LQR geregelten Systems, y

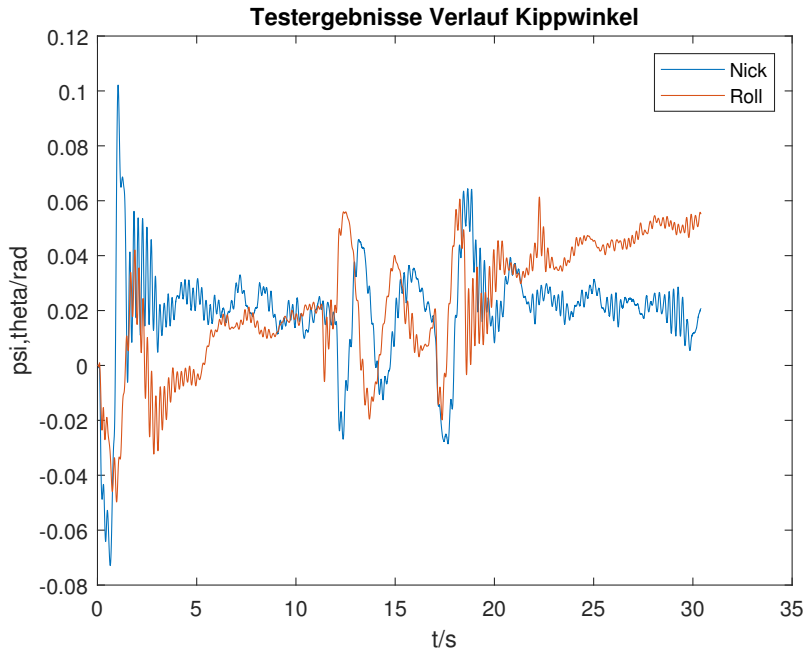


Bild 3.11: Experimentelle Erprobung des mit LQR geregelten Systems, ϕ, θ

Dynamik für den s_z -Zustand verschleunert, hat in Experimenten zu keinen zufriedenstellenden Ergebnissen geführt. Durch eine Störgrößenaufschaltung der Schwerkraft könnte man gegebenenfalls ein besseres Ergebnis erzielen [FKL⁺13].

Für den x -Wert in Bild 3.9 lässt sich in den ersten fünf Sekunden eine starke Schwankung beobachten. Dieses Phänomen kommt durch die ungleichmäßig drehenden Rotoren zustande, die den Kopter aus seiner Startposition zunächst in x -Richtung befördern. Auch die von Null abweichenden Endwerte der Kippwinkel in Bild 3.11 sind ein Artefakt dieser Modellunsicherheit. Bei unterschiedlich stark drehenden Rotoren muss das Fluggestell leicht schief stehen, sodass es die Position beibehält.

Der Schätzwert der x - y -Position weicht mit der Zeit immer mehr von dem tatsächlich beobachteten Wert ab. Dies lässt sich anhand der Messdiagramme nicht sehen, da diese lediglich den geschätzten Wert der Zustände reflektieren, nicht aber den realen. Um diesen Drift nachzuweisen, wurden heuristische Mittel verwendet. So findet sich in Anhang D eine Videoaufnahme des Kopters, die zeigt, dass dieser selbst bei Sollgröße $(x_s, y_s) = (0,0)$ nicht auf der Stelle fliegt sondern immer weiter von seiner Position abweicht.

Dieser unerwünschte Effekt entsteht durch eine fehlerhafte Zustandsschätzung. Da die

Schätzung der x - y -Position nicht direkt durch Messung unterstützt wird, sondern lediglich durch Integration der Ableitungen 1. und 2. Ordnung gebildet wird, unterliegt sie einem *Integrationsdrift*. Dieser entsteht durch eine dauerhafte Messabweichung in einer der zeitlichen Ableitungen, die aufintegriert wird und in Folge immer mehr anwächst [Bri71]. Dieser Effekt soll mittels Erweiterung des Zustandsbeobachters durch eine Kameraauswertung in den folgenden Kapiteln eliminiert werden.

Kapitel 4

Kameraauswertung

Im folgenden Kapitel soll der Zustandsbeobachter des Kopters um eine Auswertung der durch die Bauchkamera aufgenommenen Bilddaten erweitert werden. Zum Einen soll dadurch die direkte Messung des Gierwinkels ermöglicht werden. Die Schätzung des Gierwinkels ψ basierte zuvor lediglich auf einer Integration der Inertialsensordaten. Desweiteren soll die x - y -Position unmittelbar gemessen werden. Dies vermeidet gegenüber den Schätzungen durch Akzelerometer und optischen Fluss das Problem des Integrationsdrifts. Die Ergebnisse fließen in die Zustandsschätzung ein, indem sie später durch das in Kapitel 5 auszulegende *Kalman-Filter* aufgearbeitet werden. Ziel ist es, dadurch die Position des Kopters im Raum besser stabilisieren zu können.

Die Messung der Positionsgrößen durch die Bauchkamera wird mittels Farberkennung an einer Bodenmarkierung implementiert. Im Folgenden wird erarbeitet, wie mithilfe einer offline definierten Bodenmarkierung Messwerte für x , y und ψ ermittelt werden können. Es werden Beschränkungen dieser Funktionalität gezeigt, die im Rahmen einer Ausreißerbehandlung berücksichtigt werden. Die nötigen Berechnungen werden dann in den Kontext des Echtzeitsystems gebracht, indem eine geeignete Aufrufperiode bestimmt wird. Anschließend wird der Algorithmus simulativ und durch Erprobung in der Realität validiert.

4.1 Positionserkennung mittels Bodenmarkierung

Da die gegebene Problemstellung eine fest definierte Bodenmarkierung erlaubt, die außerdem frei wählbar ist, kann die Allgemeinheit des Algorithmus stark einge-



Bild 4.1: Muster für die Bodenmarkierung

schränkt werden. In [PKYK17] wird ein Konzept vorgestellt, das Gebrauch von einer schwarz-weißen, kreisförmigen Bodenmarkierung macht. Diese Markierung ist radial-symmetrisch, abgesehen von einem Achtelsegment, welches farbinvertiert ist. Durch die unterbrochene Radialsymmetrie lässt sich die Verdrehung des Musters eindeutig feststellen. Daraus ist es dann möglich, den Gierwinkel des Kopters auszurechnen.

Die in [PKYK17] vorgestellten Markererkennung verwendet Konzepte wie Kanten-erkennung, Rauschunterdrückung und Identifikation von Formen. Diese beinhalten jedoch relativ rechenintensive Algorithmen, die für die Berechnung auf dem Kopter zu langsam wären. Daher wird im Rahmen dieser Arbeit stattdessen eine Herangehensweise verwendet, die Gebrauch von Farberkennung macht. Dies hat zur Folge, dass der sonstige Untergrund gewisse farbliche Anforderungen erfüllen muss und sich die Belichtungsintensität innerhalb eines gewissen Toleranzbandes befinden muss.

Das für die Kameraauswertung verwendete Muster ist in Bild 4.1 dargestellt. Die exakte geometrische Form des Musters ist weitestgehend willkürlich. Es muss lediglich sichergestellt werden, dass sich der Verdrehwinkel auf Basis des Musters eindeutig identifizieren lässt. Dies ist hier gegeben. Um Überbelichtungseffekte zu vermeiden, wurde eine möglichst dunkle Farbwahl getroffen.

4.2 Berechnung von ψ , x und y aus RGB-Daten

Im Folgenden wird die Vorgehensweise beschrieben, mit der die RGB-Daten in Messwerte für ψ , x und y umgerechnet werden. In Bild 4.2 wird in einem anschaulichen Ablaufdiagramm dargestellt, mit welchen Schritten der Algorithmus zum Ergebnis kommt. Für die Berechnung liegt die Bildaufnahme in Form eines Tensors dritter

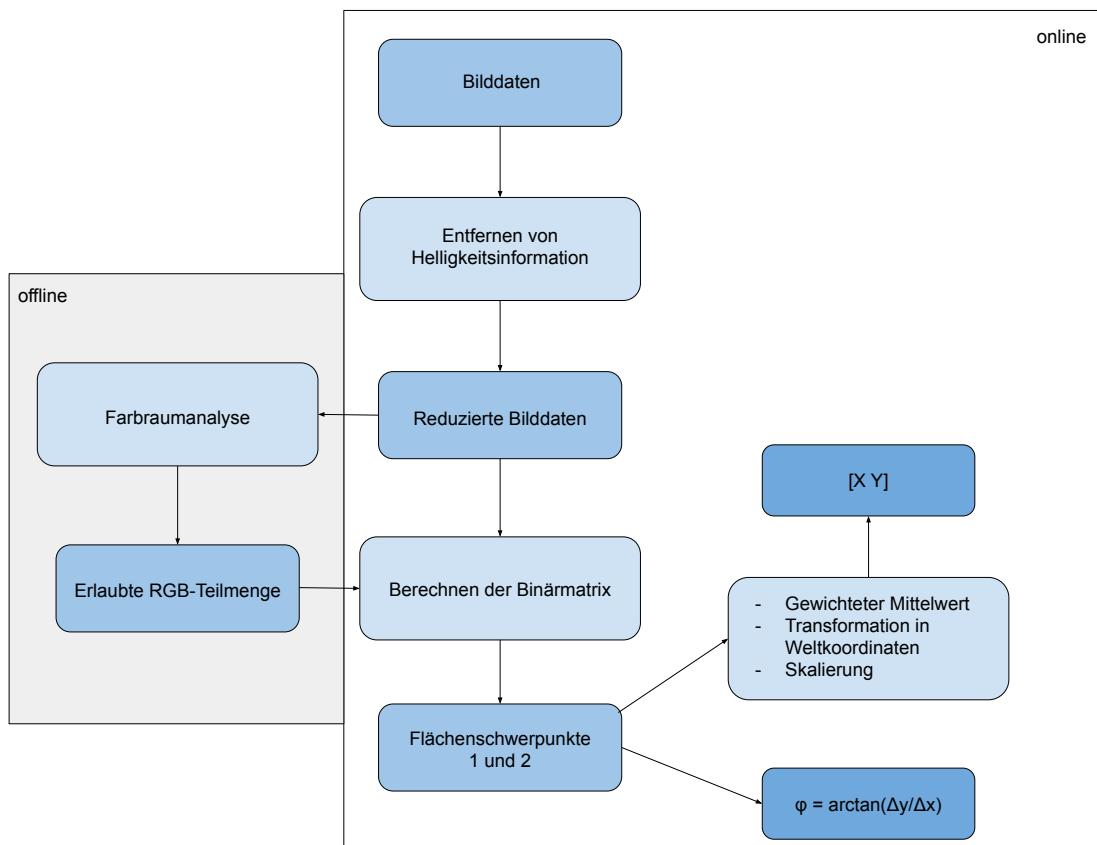


Bild 4.2: Ablaufdiagramm für den Bildverarbeitungsalgorithmus

Ordnung $G \in \mathbb{R}^{120 \times 160 \times 3}$ vor. $i \in [1, \dots, 120]$ ist der Index, der die Pixel in Richtung negativer x -Richtung des Kopters zählt. $j \in [1, \dots, 160]$ zählt die Pixel in positive y -Richtung des Kopters. Es gilt $k \in \{1, 2, 3\}$, wobei die Matrizen für $k = 1$ dem Rotanteil, $k = 2$ dem Grünanteil und $k = 3$ dem Blauanteil der Aufnahme entsprechen. Die Einträge des Tensors nehmen dabei Werte zwischen 0 und 255 an, wobei 0 der minimalen Intensität des jeweiligen Farbanteils entspricht und 255 der maximalen Intensität. Es gilt also

$$G_{i,j,k} \in [0, \dots, 255] \forall i, j, k. \quad (4.1)$$

Es hat sich herausgestellt, dass zunächst eine Reduktion der Bildaufnahme gemäß Gleichung 4.2 dienlich ist, um die Bedingungen an die äußeren Belichtungsverhältnisse zu lockern.

$$G_{reduced,i,j,k} = G_{i,j,k} - \min_k G_{i,j,k} \quad (4.2)$$

Dabei wird das Minimum für jeden Pixel in der Farbdimension k gebildet.

Mindestens einer der drei Farbeinträge in jedem Pixelvektor von $G_{reduced}$ wird demnach zu null. Dies ist anschaulich eine stark vereinfachte Vorgehensweise um die Helligkeitsinformation zu entfernen. Experimente haben gezeigt, dass dadurch der Algorithmus an Allgemeinheit bezüglich Belichtungsverhältnissen gewinnt.

Der resultierenden Pixeltensor wird nun virtuell in 120×160 Vektoren mit je drei Einträgen zerlegt. Diese Vektoren entsprechen jeweils einem Pixel aus der Aufnahme. Für diese Pixelvektoren wird dann jeweils geprüft, ob sie sich im RGB-Farbraum nahe genug an einem definierten Farbwert f befinden. Daraus ergibt sich jeweils für Dunkelrot und Dunkelblau eine Binärmatrix der Größe 120×160 . In dieser Matrix sagt jeder Pixel aus, ob sich der entsprechende Pixelvektor der Aufnahme innerhalb des zulässigen Farbbereichs befunden hat.

Um diesen zulässigen Farbbereich zu bestimmen, wurde im Rahmen einer Offline-Analyse mit der Kopteramera Bildaufnahmen der Bodenmarkierung bei unterschiedlichen Lichtverhältnissen gemacht. Mithilfe derer wurde dann erprobt, welcher Teil des RGB-Farbraums am besten geeignet ist, um die Markierung möglichst vollständig zu identifizieren, ohne dabei andere Bereiche des Bildes fälschlicherweise einzubeziehen. Das MATLAB-Skript zur Durchführung dieser Analyse findet sich in Anhang

E.1.

Die sich aus der Analyse ergebenden Farbwerte f sowie Toleranzen d ergeben sich zu

$$\begin{aligned}
 f_{\text{dunkelrot}} &= [63 \ 0 \ 0]^T \\
 f_{\text{dunkelblau}} &= [0 \ 38 \ 95]^T \\
 d_{\text{dunkelrot}} &= [15 \ 20 \ 20]^T \\
 d_{\text{dunkelblau}} &= [45 \ 45 \ 45]^T.
 \end{aligned} \tag{4.3}$$

Dabei ist d der maximal erlaubte Betrag der Abweichung von f in R-, G- bzw. B-Richtung.

Idealerweise würde nach diesem Schritt eine Rauschunterdrückung auf die Binärmaske vorgenommen werden. Eine solche Rauschunterdrückung könnte durch Entfernung unzusammenhängender Anteile der Maske erreicht werden, beispielsweise durch einen *Area-Opening-Algorithmus* [HS92]. Da dieser Vorgang allerdings rechenintensiv ist, muss darauf verzichtet werden. Dies geht mit einer Minderung der Genauigkeit des Algorithmus einher. Durch geschickte Wahl der Bedingungen und angepasste Lichtverhältnisse können nichtsdestotrotz gute Ergebnisse erzielt werden.

Die resultierenden Binärmatrizen werden nun verwendet, um den jeweiligen Flächen-schwerpunkt $(x_{\text{img},i}, y_{\text{img},i})$ der beiden Farbanteile zu bestimmen. Diese Koordinaten beziehen sich auf die Position des Pixels in der zweidimensionalen Matrix.

Berechnung des Gierwinkels

Aus den beiden Koordinatenpaaren $(x_{\text{img},1}, y_{\text{img},1})$ für Dunkelrot und $(x_{\text{img},2}, y_{\text{img},2})$ für Dunkelblau kann über die Auswertung des vier-Quadranten-Arkustangens $\arctan2$ gemäß (4.4) der relative Gierwinkel ψ_{vis} des Kopters errechnet werden. Dafür werden die Richtungsvektoren $y_{\text{img},2} - y_{\text{img},1}$ und $x_{\text{img},2} - x_{\text{img},1}$ berechnet. Die x-Richtung muss invertiert werden, da die Matrixform des Bildes die Pixel in x -Richtung von oben nach unten zählt, der x -Einheitsvektor des Körperkoordinatensystems jedoch in Bildrichtung nach oben definiert ist.

$$\psi_{\text{vis}} = \arctan2(y_{\text{img},2} - y_{\text{img},1}, -(x_{\text{img},2} - x_{\text{img},1})) \tag{4.4}$$

Wenn der Kopter derart ausgerichtet ist, dass der dunkelblaue Flächenschwerpunkt exakt rechts vom dunkelroten Flächenschwerpunkt liegt (siehe Bild 4.1), dann gilt somit $\psi_{vis} = 0$.

Berechnung der x - y -Weltkoordinaten

Der gemeinsame Mittelpunkt kann durch eine Mittelwertbildung der zwei Schwerpunkte bestimmt werden. Hierbei muss ebenfalls aufgrund der Definition des Bildkoordinatensystems und des Kopterkoordinatensystems die x -Richtung invertiert werden. Außerdem wird eine Verschiebung des Koordinatensystems in den Bildmittelpunkt durchgeführt. Gleichung 4.5 bildet somit den Mittelwert der zwei Schwerpunkte, verschiebt das Zentrum des Koordinatensystems um $\frac{n}{2}$ und invertiert die x -Richtung. $n = [n_x \ n_y]^T = [120 \ 160]^T$ bezeichnet die Anzahl der Bildpixel in die jeweilige Richtung.

$$\begin{bmatrix} x_{img} \\ y_{img} \end{bmatrix} = \frac{1}{2} \cdot \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \cdot \left(\begin{bmatrix} x_{img,1} \\ y_{img,1} \end{bmatrix} + \begin{bmatrix} x_{img,2} \\ y_{img,2} \end{bmatrix} - \begin{bmatrix} n_x \\ n_y \end{bmatrix} \right) \quad (4.5)$$

$[x_{img} \ y_{img}]^T$ stellt somit die Position des Bildmittelpunkts in Bezug auf die Position des Kopters im Körperkoordinatensystem des Kopters dar.

Mithilfe des vom Beobachter im letzten Schritt geschätzten Gierwinkels ψ_{est} kann daraus dann nach (4.6) der x - y -Abstand des Kopters vom Mittelpunkt des Musters in Weltkoordinaten umgerechnet werden.

$$\begin{bmatrix} x_{w,pix} \\ y_{w,pix} \end{bmatrix} = -T^{WB}(\psi_{est}) \begin{bmatrix} x_{img} \\ y_{img} \end{bmatrix} = - \begin{bmatrix} \cos \psi_{est} & -\sin \psi_{est} \\ \sin \psi_{est} & \cos \psi_{est} \end{bmatrix} \begin{bmatrix} x_{img} \\ y_{img} \end{bmatrix} \quad (4.6)$$

T^{WB} bezeichnet die Transformationsmatrix von Körper- zu Weltkoordinaten, die durch Transposition aus der in Gleichung 2.20 bereits definierten Transformationsmatrix von Welt- zu Körperkoordinatensystem gewonnen werden kann. Außerdem muss der Vektor umgekehrt werden, sodass er die Position des Kopters in Bezug auf den Bildmittelpunkt beschreibt. Dies erfolgt durch das negative Vorzeichen in Gleichung 4.6.

Die in Einheit Pixellänge gemessenen x - y -Koordinaten müssen nun in Meter umgerechnet werden. Der dazu verwendete Faktor $t_{w,pix}(z)$ hängt anschaulicherweise von der Flughöhe z des Kopters ab; bei einer Erhöhung der Flughöhe wächst die reale

Distanz, die durch einen Pixel repräsentiert wird. Die Funktion wurde empirisch ermittelt und ergibt sich zu $t_{w,pix}(z_{est}) = 0,005 \cdot |z_{est}|$.

Die z -Abhängigkeit wurde hier als linear angenommen. Dies ist gültig für ausreichend große Werte von z , eine Bedingung, die von anderen Subsystemen des Kopters ohnehin gefordert wird.

Damit ergibt sich die reale Distanz des Kopters zum Mittelpunkt des Musters in Weltkoordinaten zu

$$\begin{bmatrix} x_w \\ y_w \end{bmatrix} = t_{w,pix}(z_{est}) \cdot \begin{bmatrix} x_{w,pix} \\ y_{w,pix} \end{bmatrix}. \quad (4.7)$$

Die entsprechenden MATLAB-Funktionen finden sich in Anhang E.3.

4.3 Arten der Ausreißerbehandlung

Im Folgenden wird eine Ausreißerbehandlung auf die Ergebnisse des Algorithmus angewendet. Diese dient dazu, unplausible Messwerte auszusortieren, schon bevor sie an den Beobachter weitergegeben werden. Generell kann eine Ausreißerbehandlung auf zwei Achsen operieren. Zur leichteren Unterscheidung im Rahmen dieser Arbeit sollen entsprechende Begriffe eingeführt werden. Die Konzepte und konkreten Werte sind Ergebnisse experimenteller Beobachtungen und verwandter Arbeiten wie [Yan] [Gre] [Lan].

Präventive Ausreißerbehandlung stellt sicher, dass gewissen Rahmenbedingungen während der Aufnahme erfüllt sind. So können voraussichtlich ungültige Werte schon im Vorfeld unberücksichtigt bleiben. Diese Art von Ausreißerbehandlung kommt in dieser Arbeit zum Einsatz.

Komparative Ausreißerbehandlung vergleicht die Ergebnisse mit den aktuellen Schätzwerten für die entsprechenden Zustände, und entfernt stark abweichende Ausreißer. Diese Vorgehensweise kann allerdings zu einer dauerhaften Blockierung führen. Dies geschieht genau dann, wenn für einen kurzen Zeitraum die Messwerte deutlich näher an der Realität liegen als die Schätzwerte. Das System ignoriert ab dann jeglichen zukünftigen Messwert. Dadurch geht die Möglichkeit verloren, den Schätzwert wieder an die Realität anzunähern.

Aus diesem Grund wird hier auf eine komparative AuBreiBerbehandlung verzichtet. Eventuelle AusreiBer, die der praventiven Behandlung entgangen sind, k6nnen durch das Kalman-Filter gehandhabt werden.

4.4 Praventive AusreiBerbehandlung

Damit der Algorithmus nutzbare Werte hervorbringen kann, m6ussen einige Bedingungen erf6ullt sein. Es gibt zwei Kriterien, nach denen eine Aufnahme verworfen werden kann. Zum Einen erfolgt auf Basis der Gesamtanzahl an Pixeln eine *Plausibilit6atspr6ufung*. Desweiteren kann unter Ber6ucksichtigung des gesch6atzten Momentanzustands vorhergesagt werden, ob das n6achste Kamerabild verwendet werden kann. So verursachen beispielsweise hohe Geschwindigkeiten Bewegungsunsch6arfe und machen damit eine Aufnahme unbrauchbar. AuBerdem kann ein *zul6assiger Zustandsraum* eingegrenzt werden.

Plausibilit6atspr6ufung

Durch ung6unstige Belichtungsverh6altnisse kann es dazu kommen, dass die offline definierten Farbr6aume ihre G6ultigkeit verlieren. Dies kann wiederum dazu f6uhren, dass die Markierung nur teilweise erkannt wird oder aber der Rest des Bildes f6alschlicherweise als Markierung identifiziert wird. Um in diesen F6allen keine falschen Ergebnisse an den Zustandsbeobachter weiterzugeben, werden zwei Plausibilit6atspr6ufungen durchgef6uhrt:

- Wenn $|\log(\frac{A_1}{A_2})| > 0,5$, wobei A_i die Gesamtzahl der Pixel in der jeweiligen Maske bezeichnet, dann bedeutet das anschaulich, dass sich die Maskengr6oBen stark voneinander unterscheiden. Dies kann auftreten, wenn eine der Farbmasken aufgrund von Beleuchtungseffekten verrauscht ist. Es kann auch ein Indikator daf6ur sein, dass der Kopter so weit von der Markierung entfernt ist, dass diese in der Aufnahme abgeschnitten wird. Die Gierwinkelberechnung ist in beiden F6allen nicht mehr m6oglich. Die x - y -Berechnung ist im Allgemeinen weiterhin nutzbar.
- Wenn $(A_1 + A_2 < 50) \vee (A_1 + A_2 > 5000)$, dann wird sowohl Gierwinkel als auch x - y -Position ignoriert. Dies tritt dann auf, wenn die Markierung entweder gar nicht erkannt oder die Maske von Rauschen dominiert wird.

Zulässiger Zustandsraum

Um eine Aussage über die grundsätzliche Nutzbarkeit der Aufnahme zu treffen, wird außerdem geprüft, ob gewissen Grenzen im Zustandsraum eingehalten werden.

- Überschreitet der Betrag einer der Kippwinkel ϕ oder θ einen Maximalwert, dann verschiebt sich die Markierung in der Aufnahme relativ zum Bildmittelpunkt. Diese Verschiebung ist sehr empfindlich auch für kleine Winkel. Weil zudem das Subsystem der Kippwinkel deutlich schneller ist als der bildverarbeitende Systemteil, können diese Effekte nicht berücksichtigt werden. Dieser Bereich des Zustandsraums muss daher ausgeschlossen werden.
- Überschreitet eine der Geschwindigkeiten $\sqrt{p^2 + q^2}$ oder $\sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}$ einen Maximalwert, so wird die Aufnahme aufgrund von Bewegungsunschärfe nutzlos.
- Unterschreitet die Flughöhe z einen Mindestwert, so werden Teile der Markierung abgeschnitten. Außerdem kommen dann perspektivische Verzerrungseffekte zum Tragen. Die Aufnahme wird unbrauchbar.

Wird also eines der gelisteten Kriterien verletzt, so gilt die Annahme, dass das Ergebnis oder Teile davon unbrauchbar sind. Die entsprechenden Werte werden somit nicht an den Zustandsbeobachter weitergegeben.

4.5 Berechnungsperiode

Da man es in der bildverarbeitenden Algorithmik mit vergleichsweise großen Datenmengen zu tun hat, müssen wegen der geringen Rechenleistung und der Timing-Anforderungen des Quadropters als Echtzeitsystem eventuelle Laufzeitprobleme berücksichtigt werden [Liu00]. Daher musste bei der Konzeption der Bilddatenauswertung ein Augenmerk auf den Gebrauch von Rechenkapazitäten gelegt werden. Zudem müssen Überlegungen angestellt werden, mit welcher Häufigkeit der Bildverarbeitungsalgorithmus ausgeführt werden soll, um einen guten Kompromiss zwischen Informationsgewinn und Rechenressourcen zu erreichen.

Die Kamera liefert ein RGB-Bild mit einer Auflösung von 120 x 160 Pixeln. Es liegt nahe, dass Matrixoperationen von dieser Größenordnung auf dem im *Parrot Mambo* verbauten Rechner eine nicht zu vernachlässigende Laufzeit beanspruchen.

Trotz Einsparnissen in der Architektur des Algorithmus hat sich bei einer Aufrufperiode von 300ms die durchschnittliche Laufzeit der Bildverarbeitung zu 192ms ergeben. Die Stichprobengröße entsprach dabei 65 Aufrufzyklen. Diese Messung ist im Parallelbetrieb zur sonstigen Flugregelung erfolgt, wie es auch unter Normalbedingungen der Fall wäre. Für die Messung wurde die in den *Flight Log* des Quadropters bereits integrierte Funktionalität zur Messung der Laufzeit der Bildverarbeitung verwendet.

Um einen verlässlichen Betrieb zu ermöglichen, wurde daher die Aufrufperiode des Kameraauswertungsalgorithmus zu 300ms gewählt. Dies ist zwar kein Vielfaches von 200ms und bringt dadurch eine zwischen 0ms und 100ms alternierende Latenz mit sich. Die mittlere Latenz fällt jedoch geringer aus als bei dem nächsten möglichen Vielfachen von 200ms, nämlich 400ms.

Gegenüber dem Regler, der alle 5ms berechnet wird, ist dies extrem langsam. Da die Dynamik des x - y -Systemanteils jedoch ohnehin eine vergleichsweise hohe Zeitkonstante aufweist und die Kamera lediglich den langfristigen Drift dieser Zustände kompensieren soll, ist eine langsame Bildverarbeitung akzeptabel. In der Praxis hat sich diese Häufigkeit ebenfalls als funktionsfähig erwiesen.

4.6 Erprobung des Algorithmus

Der in diesem Kapitel entwickelte Algorithmus soll nun validiert werden. Um vorläufige Aussagen über die Fehlerfreiheit der Implementierung treffen zu können, wurde eine stark vereinfachte Simulation des Kamerabildes entwickelt. Diese nimmt als Eingabe den Zustandsvektor $[x \ y \ z \ \phi \ \theta \ \psi]^T$ und rechnet ihn in ein entsprechendes 120×160 RGB-Bild der Bodenmarkierung um. Berücksichtigt wurden ebenfalls die linearisierte Verschiebung des Bildes in Abhängigkeit von θ und ϕ , der Untergrund sowie Helligkeitsrauschen. Mit Schieberegler für die Eingangswerte konnte diese Simulation visuell-intuitiv verifiziert werden. Die entsprechenden Skripte finden sich in Anhang E.2.

Die Simulation wurde dann mit Zustandsverläufen verbunden. Es wurde überprüft, ob der Algorithmus aus den von der Simulation produzierten RGB-Bildern auf die Zustandsverläufe zurückschließen kann. Dies stellte sich als erfolgreich heraus.

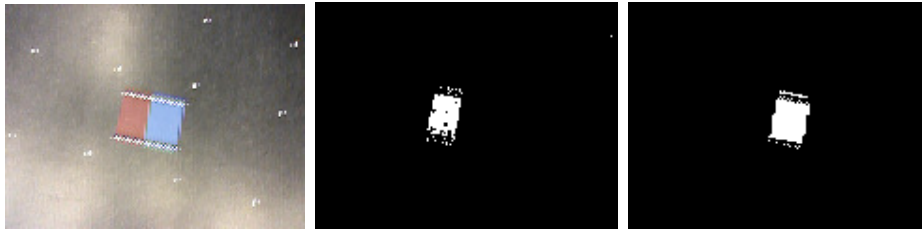


Bild 4.3: Maskenanalyse: Aufnahme, Rot-Maske, Blau-Maske

Als zweiter Validierungsschritt wurden Bilder mit dem Kopter aus unterschiedlichen Positionen und Lagen aufgenommen. Es wurde dann überprüft, ob der Algorithmus die zu erwartenden Messungen liefert. Die tatsächlichen Werte für die Zustände wurden heuristisch abgeschätzt.

Zudem wurden bei unterschiedlicher Belichtung stichprobenartig Detektionsmasken analysiert. Jeweils eine Maske für die Rot-Erkennung und für die Blau-Erkennung sowie die ursprüngliche Aufnahme zeigt Bild 4.3. Die weißen Pixel indizieren dabei, dass dort die entsprechende Farbe erkannt wurde. Die Aufnahme liegt in Originalauflösung vor. Diese Analyse diente dem Zweck, den Einfluss der Umgebungsbelichtung auf den Farberkennungsalgorithmus zu verstehen. Es hat sich herausgestellt, dass die Funktionsfähigkeit des Algorithmus trotz ergriffener Gegenmaßnahmen weiterhin zu einem geringen Grad von der Art und Intensität der Beleuchtung abhängig ist. Unter Berücksichtigung dieser Tatsache arbeitet der Algorithmus jedoch zufriedenstellend und kann für den Betrieb des Kopters zur Erweiterung der Zustandsschätzung eingesetzt werden.

Die technische Umsetzung der Bodenmarkierung birgt noch Verbesserungspotential. Als Prototyp wurden hier bisher zwei farbige Blätter Papier verwendet, die entsprechend zusammengetackert wurden. Diese wurden dann mit Gewichten am Boden fixiert. Um die Lebensdauer und Handlichkeit des Versuchsaufbaus zu verbessern, könnte man hier beispielsweise ein Stück Holz zurechtschneiden und dort das Papier aufbringen oder gar direkt das Holz einfärben. Eine Bodenmarkierung muss Schall reflektieren, sodass die Funktion des Ultraschallsensors für die Höhenmessung des Kopters nicht gestört wird. Eine bedruckte Spielmatte aus Gummi wurde beispielsweise schon erprobt und hat sich dahingehend als untauglich erwiesen. Sollten die Farben der Markierung sich ändern, so müssen gemäß Abschnitt 4.2 die Farbwerte erneut ermittelt und das Matlab-Skript `XYPsiFromVis` entsprechend angepasst werden.

Kapitel 5

Entwurf eines Zustandsbeobachters

Aus der Kameraauswertung ergeben sich alle 300 Millisekunden Messwerte für die x - y -Position und den Gierwinkel ψ . Die bestehende Sensorik liefert bereits Messwerte für Ableitungen dieser Größen auf Basis der inertialen Messeinheit sowie der Auswertung des optischen Flusses. Um diese drei Informationsquellen zu einem Schätzwert zu kombinieren, soll ein Kalman-Filter zum Einsatz kommen.

Die Beobachtung der übrigen Teilsysteme des Kopters sollen ebenfalls in Form eines Kalman-Filters erfolgen. Für die Höhenschätzung und Kippwinkelschätzung werden die in [Yan] bereits entwickelten Filter als Grundlage verwendet, für die restlichen Größen werden neue Filter entworfen. Die Teilbeobachter sollen dann in einem letzten Schritt zu einem systemumfassenden Beobachter zusammengeführt werden, um später leichter ein Modell von der gesamten Systemdynamik ermitteln zu können. Außerdem erleichtert ein übersichtliches, ganzheitliches Modell die Anpassbarkeit der Beobachterdynamik zu einem späteren Zeitpunkt.

Dafür sollen zunächst die allgemeinen Grundlagen zum Kalman-Filter vorgestellt werden. Es wird dann darauf eingegangen, wie diese Konzepte sich auf den linearen, zeitinvarianten Fall übertragen lassen. Für jedes Teilsystem des Kopters soll dann ein entsprechendes Teilfilter entworfen werden. Diese Teilfilter werden in einem letzten Schritt zu einem Gesamtfiter fusioniert. Anhand der resultierenden Beschreibung für den Zustandsbeobachter soll die Gesamtdynamik des Quadroptersystems aufgestellt werden. Zuletzt werden die erarbeiteten Konzepte simulativ sowie physikalisch erprobt.

5.1 Stochastische Grundlagen zum Kalman-Filter

Ein Kalman-Filter ist ein Algorithmus, der alle verfügbaren Informationen zu einem Schätzwert verarbeitet. Dazu zählen aktuelle Messwerte, Messunsicherheit bzw. -rauschen sowie Systemrauschen. Der Einsatz eines Kalman-Filters erfordert keine Speicherung der gesamten Historie an Messwerten, da diese implizit im letzten Schätzwert enthalten ist. Man spricht von einem *rekursiven* Algorithmus [May].

Information über einen Systemzustand kann unter Verwendung der aktuellen Messwerte und der Rausch- sowie Systemmatrizen in Form von Wahrscheinlichkeitsdichtefunktionen dargestellt werden. Durch die Annahme, dass lediglich weiße gaußsche Rauschprozesse auftreten, kann die Funktion einer Informationsquelle jeweils durch Mittelwert μ sowie Standardabweichung σ eindeutig beschrieben werden. Hierbei wird die Annahme getroffen werden, dass es sich bei System- und Messrauschen um weißes gaußsches Rauschen handelt.

$$\mu_{comb} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} \mu_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \mu_2 \quad (5.1)$$

$$\frac{1}{\sigma_{comb}^2} = \frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \quad (5.2)$$

Diese Wahrscheinlichkeitsverteilungsfunktionen mehrerer Informationsquellen über den selben Systemzustand können nun zusammengeführt werden indem man Formeln (5.1) für den kombinierten Mittelwert und (5.2) für die kombinierte Standardabweichung anwendet [May]. Diese Formeln beziehen sich auf zwei Informationsquellen, lassen sich aber durch mehrfache Anwendung trivial auf n Quellen erweitern.

Anschaulich hat diese Kombination mehrerer Informationsquellen zum Zweck, aus mehreren Schätzungen für dieselbe Größe eine einzelne, möglichst genaue Schätzung zu extrahieren. Einer Schätzung, die mit einer hohen Standardabweichung einhergeht, wird dabei weniger Vertrauen geschenkt als einer Schätzung mit einem niedrigen Wert für die Standardabweichung.

Bild 5.1 veranschaulicht den Vorgang für zwei beispielhafte Funktionen z_1 und z_2 . Dabei gelten für die Standardabweichungen $\sigma_1 = 0,3$, $\sigma_2 = 0,1$ und für die Mittelwerte

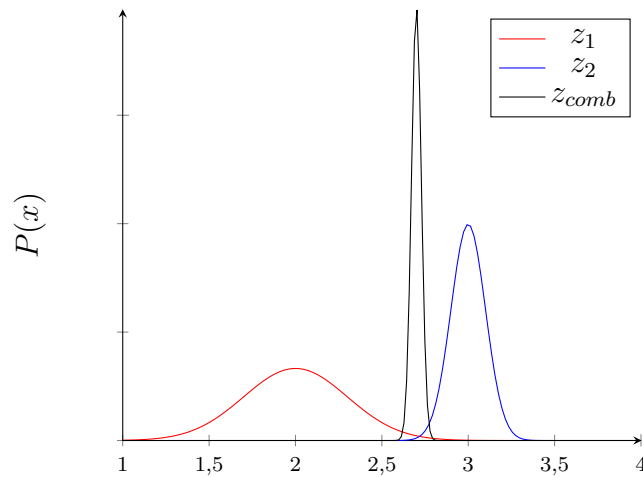


Bild 5.1: Fusion zweier Wahrscheinlichkeitsdichtefunktionen

$\mu_1 = 2$, $\mu_2 = 3$. Der Maximalwert des kombinierten Graphen entspricht dem Mittelwert der kombinierten Funktion $\mu_{comb} = 2,702$ und ist somit der beste Schätzwert für den Systemzustand. Die Standardabweichung der kombinierten Funktion beträgt $\sigma_{comb} = 0,009$. Man kann leicht sehen, dass der Schätzung mit der größeren Standardabweichung und damit breiteren Funktion z_1 weniger vertraut wird, als derjenigen mit der schlanken Funktion z_2 mit geringer Standardabweichung. Der gemeinsame Mittelwert ist deutlich näher an dem Mittelwert von z_2 . Ebenso ist die Standardabweichung der kombinierten Funktion geringer, die Funktion ist schmaler als beide Ausgangsfunktionen. Dies ist zu erwarten, und spiegelt einen Informationsgewinn wider.

5.2 Kalman-Filter für das lineare Zustandssystem

Für den Sonderfall, dass in Bezug auf die zu schätzenden Größen ein lineares System vorliegt, dessen Dynamik bekannt ist, dessen Ausgang gemessen und dessen Eingang vorgegeben wird, kann eine etablierte Form des Kalman-Filters verwendet werden [WB06].

Die Systemgleichungen müssen dafür in folgender, zeitdiskreter Form vorliegen:

$$\begin{aligned} X_k &= A_k X_{k-1} + B_k U_k + \omega_k \\ Y_k &= C_k^T X_k + \nu_k. \end{aligned} \tag{5.3}$$

Dabei sind ω_k weißes Systemrauschen und ν_k weißes Messrauschen.

Zeitvariantes Kalman-Filter

Um das Rauschverhalten des Systems und der Messeinrichtungen zu berücksichtigen, müssen die entsprechenden statistischen Eigenschaften modelliert werden. Dies geschieht in Form der Rauschmatrizen Q für das Messrauschen und R für das Systemrauschen.

Es lässt sich dann ein zweiteiliger, rekursiver Algorithmus definieren, der zu jedem Zeitschritt einmal durchlaufen wird. Dabei werden Systemeingang, Messwert und Rauschmatrizen nach den in Kapitel 5.1 besprochenen Prinzipien verrechnet. Ergebnis dessen ist ein Schätzwert aller Zustände. Dieser Schätzwert bezieht jegliche gegebenen Informationen mit ein. Für den Algorithmus müssen außerdem Anfangsbedingungen \hat{X}_0 für den Anfangszustand und P_0 für die initiale Fehlerkovarianzmatrix definiert werden.

Der Algorithmus lässt sich in zwei konzeptuelle Schritte aufgliedern.

$$\begin{aligned}\hat{X}_k^- &= A\hat{X}_{k-1} + BU_k \\ P_k^- &= AP_{k-1}A^T + Q\end{aligned}\tag{5.4}$$

Im Prädiktionsschritt (5.4) werden die Systemgleichungen auf Zustands- und Eingangsgrößen des letzten Zeitschritts angewandt. Die Schätzfehler-Kovarianzmatrix P_{k+1}^- wird unter Berücksichtigung der positiv semidefiniten Systemrauschkovarianzmatrix Q vorhergesagt.

$$\begin{aligned}K_k &= P_k^- C(C^T P_k^- C + R)^{-1} \\ \hat{X}_k &= \hat{X}_k^- + K_k(Y_k - C^T \hat{X}_k^-) \\ P_k &= (I - K_k C^T) P_k^-\end{aligned}\tag{5.5}$$

Im Korrekturschritt (5.5) wird die Kalman-Verstärkung K_k berechnet, aus der sich dann die neuen Schätzungen für Schätzwert und Schätzfehler-Kovarianzmatrix ergeben. Es fließt die Messrauschkovarianzmatrix R ein. \hat{X}_k und P_k beschreiben die durch den Messwert Y_k korrigierten Werte für den Schätzwert und die Schätzfehler-Kovarianzmatrix.

Zeitinvariantes Kalman-Filter

In dieser Arbeit soll für den Kopter zunächst ein zeitinvariantes Kalman-Filter entworfen werden. Dieses wird wie ein Luenberger-Beobachter angesehen [Lun16] [FKL⁺13]. Es wird einmalig eine Beobachter-Verstärkungsmatrix L berechnet. Diese entspricht einer zeitinvarianten Kalman-Verstärkung K_k wie in Gleichung 5.5. Im Weiteren wird die Beobachterverstärkung als L bezeichnet, d.h. $L = K_k$. Die Systemgleichung des Beobachters ergibt sich dann mittels Einsetzen von \hat{X}_k^- aus Gleichung 5.4 in Gleichung 5.5 zu

$$\hat{X}_{k+1} = A\hat{X}_k + BU_k + L(Y_k - \hat{Y}_k), \quad (5.6)$$

wobei A die zeitdiskrete Dynamikmatrix, B die zeitdiskrete Eingangsmatrix und C die Ausgangsmatrix des Systems bezeichnen. Y_k ist der gemessene Ausgang des Systems, $\hat{Y}_k = C^T \hat{X}_k$ ist der Ausgang, der sich aus Verrechnung des rekonstruierten Zustands mit der Ausgangsgleichung ergibt. $A\hat{X}_k + BU_k$ wird dabei als *Simulatorterm* bezeichnet, $L(Y_k - \hat{Y}_k)$ als *Korrekturterm* [Lun16].

L kann für den Fall des zeitinvarianten Filters bestimmt werden, indem die Berechnung der Kovarianzmatrix P_k für einen Zeitpunkt $\lim_{t \rightarrow \infty}$ betrachtet wird und sich dadurch die Vereinfachung $P_\infty = P_k = P_{k+1}$ erlaubt. Dies überführt die zwei Differenzgleichungen aus Gleichung 5.4 und Gleichung 5.5 zur Berechnung der Kovarianz-Matrix in eine algebraische Gleichung. Diese Gleichung lautet mit $P := P_\infty$

$$APA^T - (APC)(C^T PC + R)^{-1}(APC)^T + Q = 0. \quad (5.7)$$

Gleichung 5.7 ist bezüglich eines linearen Zustandssystems dual zu der in Kapitel 3 aufgestellten Gleichung zum LQR-Entwurf und ist somit ebenfalls eine Riccati-Gleichung. Mithilfe des MATLAB-Befehls `idare(AdT, C, Q, R)` kann diese später für das Koptersystem gelöst werden. Entsprechend K_k in Gleichung 5.5 berechnet sich dann aus P die zeitinvariante Beobachterverstärkung zu

$$L = PC(C^T PC + R)^{-1}. \quad (5.8)$$

5.3 Auslegung des Kalman-Filters

Die Auslegung des Kalman-Filters lässt sich in vier entkoppelte Teilaufgaben untergliedern. Im Folgenden soll die Funktionsweise der einzelnen Teilfilter erläutert werden. Die Unterteilung erfolgt dabei in Gierwinkelbeobachter, Kippwinkelbeobachter, Flughöhenbeobachter sowie x-y-Beobachter. Diese Teilfilter werden anschließend zusammengeführt. Schlussendlich soll aus den gesammelten Entwurfsparametern die Beobacherverstärkung L bestimmt werden.

Gierwinkelbeobachter

Für den Gierwinkelbeobachter steht die Drehratenmessung r_{imu} aus dem Gyroskop der Inertial Measurement Unit (IMU) unmittelbar zur Verfügung [Bri71]. Außerdem ergibt sich aus der in Kapitel 4 entwickelten Kameraauswertung ein direkter Messwert des Gierwinkels ψ_{vis} .

Die Sonderfälle, für die die Kameraauswertung keine brauchbaren Werte für den Gierwinkel liefert, wurden in Abschnitt 4.4 genannt. Ist ein Messwert unbrauchbar, so bleibt der Gierwinkel-Messausgang von dem Beobachter unberücksichtigt. Dafür wird für den entsprechenden Zeitschritt der Korrekturterm zu Null gesetzt. Auf die Implementierung dieser Funktionalität wird später eingegangen.

Die diskrete Teilsystembeschreibung ergibt sich aus der in Kapitel 2 durchgeführten Systemanalyse zu

$$X_{\psi,k+1} = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} X_{\psi,k} + \begin{bmatrix} \frac{1}{2}T_s^2 J_z^{-1} \\ T_s J_z^{-1} \end{bmatrix} U_{\psi,k} + \omega_{\psi,k}, \quad (5.9)$$

und

$$Y_{\psi,k} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x_{\psi,k} + \nu_{\psi,k}, \quad (5.10)$$

wobei T_s die Abtastperiode bezeichnet und $X_{\psi} = [\psi \ r]^T$ gilt. Als Systemeingang $U_{\psi,k}$ wird der Motorschub um die z-Achse τ_z verwendet. Für die dem Kalman-Filter zuzuführenden Messwerte gilt $Y_{\psi,k} = [\psi_{vis} \ r_{imu}]^T$. ω bezeichnet das Systemrauschen, ν bezeichnet das Messrauschen.

Für die Rauschmatrizen und Anfangswerte werden

$$\begin{aligned} Q_\psi &= \begin{bmatrix} 10^{-6} & 0 \\ 0 & 10^{-4} \end{bmatrix} \\ R_\psi &= \begin{bmatrix} 5 \cdot 10^{-3} & 0 \\ 0 & 10^{-5} \end{bmatrix} \\ X_{\psi,0} &= \begin{bmatrix} 0 & 0 \end{bmatrix} \end{aligned} \tag{5.11}$$

verwendet. Diese Werte haben sich als praxistauglich erwiesen.

Kippwinkelbeobachter

Der Kippwinkelbeobachter schätzt die Werte von Rollwinkel ϕ , Nickwinkel θ sowie deren zeitliche Ableitungen p und q . Dafür stehen, wie auch beim Gierwinkelbeobachter, Gyroskop-Messwerte aus der IMU für die entsprechenden Winkelgeschwindigkeiten p_{imu} und q_{imu} zur Verfügung. Zudem können aus der Linearbeschleunigungsmessung des Akzelerometers in der IMU mithilfe der Gleichung 5.12 Werte für die Winkel errechnet werden.

$$\begin{bmatrix} \phi_{imu} \\ \theta_{imu} \end{bmatrix} = \begin{bmatrix} \arctan\left(\frac{a_y}{a_z}\right) \\ -\arctan\left(\frac{a_x}{g}\right) \end{bmatrix} \tag{5.12}$$

Die Messung der Linearbeschleunigung liefert jedoch nur dann brauchbare Werte, wenn keine Störeinflüsse, wie bsw. Wind auf den Kopter wirken. Es kann sichergestellt werden, ob die Werte nutzbar sind, indem überprüft wird, ob sich die Summe der Vektoren

$$a = \sqrt{(\ddot{x}^2 + \ddot{y}^2 + \ddot{z}^2)} \tag{5.13}$$

in einer akzeptablen Umgebung von g befindet. Dies trifft genau dann zu, wenn außer dem Gegenschub zum Gravitationsfeld der Erde keine weitere Kraft auf dem Kopter wirkt. Ist dies nicht der Fall, so werden die entsprechenden Messergebnisse für diesen Zeitschritt ignoriert.

Das Teilsystemmodell ergibt sich auf Basis von Kapitel 2 zu

$$X_{t,k+1} = \begin{bmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} X_{t,k} + \begin{bmatrix} \frac{1}{2}T_s^2 J_x^{-1} & 0 \\ 0 & \frac{1}{2}T_s^2 J_y^{-1} \\ T_s J_x^{-1} & 0 \\ 0 & T_s J_y^{-1} \end{bmatrix} U_{t,k} + \omega_{t,k} \quad (5.14)$$

und

$$Y_{t,k} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} X_{t,k} + \nu_{t,k}. \quad (5.15)$$

Es gilt dabei $X_{t,k} = [\phi \ \theta \ p \ q]^T$. Als Systemeingang $Z_{t,k} = [\tau_x \ \tau_y]^T$ werden die Schübe um entsprechend x-Achse für Roll- und y-Achse für Nickwinkel verwendet. Für die Messwerte gilt $Y_{t,k} = [\phi_{imu} \ \theta_{imu} \ p_{imu} \ q_{imu}]^T$.

Für die Rauschmatrizen und Anfangswerte werden

$$Q_t = \begin{bmatrix} 10^{-4} & 0 & 0 & 0 \\ 0 & 10^{-4} & 0 & 0 \\ 0 & 0 & 10^{10} & 0 \\ 0 & 0 & 0 & 10^{10} \end{bmatrix} \quad (5.16)$$

$$R_t = \begin{bmatrix} 10^{-5} & 0 & 0 & 0 \\ 0 & 10^{-5} & 0 & 0 \\ 0 & 0 & 10^{-10} & 0 \\ 0 & 0 & 0 & 10^{-10} \end{bmatrix}$$

$$X_{t,0} = [0 \ 0 \ 0 \ 0]$$

verwendet. Diese Werte sind vorläufig. Das Verhalten dieses Teilfilters muss durch weitere Erprobung und Anpassung noch verbessert werden.

Flughöhenbeobachter

Der Flughöhenbeobachter wird aus [Yan] nahezu identisch übernommen. Die System-

gleichungen ergeben sich zu

$$X_{z,k+1} = \begin{bmatrix} 1 & T_s \\ 0 & 1 \end{bmatrix} X_{z,k} + \begin{bmatrix} \frac{1}{2}T_s^2 \\ T_s \end{bmatrix} U_{z,k} + \omega_{z,k}, \quad (5.17)$$

$$Y_{z,k} = \begin{bmatrix} 1 & 0 \end{bmatrix} X_{z,k} + \nu_{z,k}. \quad (5.18)$$

Es gilt dabei $X_{z,k} = \begin{bmatrix} z & \dot{z} \end{bmatrix}^T$. Als Systemeingang $U_{z,k}$ wird die Projektion des gemessenen Beschleunigungsvektors der IMU in Welt- z -Richtung \ddot{z}_{imu} verwendet. Dies liefert einen akkurateren Wert als der tatsächliche Systemeingang, nämlich der Gesamtschub des Motors τ_{tot} . Grund dafür ist, dass bei der direkten Messung eventuelle Modellunsicherheiten des Koptersystems umgangen werden. Der Messwert $Y_{z,k}$ entspricht dem mithilfe von Drucksensordaten korrigierten Ultraschallsensor-Messwert z_{son} .

Die Rauschmatrizen und der Anfangswert werden zu

$$\begin{aligned} Q_z &= \begin{bmatrix} 10 & 0 \\ 0 & 10^{-4} \end{bmatrix} \\ R_z &= 10^{-1} \\ X_{z,0} &= \begin{bmatrix} 0 & 0.5 \end{bmatrix} \end{aligned} \quad (5.19)$$

übernommen.

x-y-Beobachter

Schlussendlich soll für die x - y -Position des Kopters ein Beobachter entworfen werden. Aus Abschnitt 2.4 und Abschnitt 3.3 geht hervor, dass die x - y -bezogenen Größen im Weltkoordinatensystem geschätzt werden sollen. Das bedeutet, dass Messwerte, die im Körperkoordinatensystem vorliegen, mit dem Gierwinkel ψ_b in das Weltkoordinatensystem umgerechnet werden müssen. Der Gierwinkel wird weiterhin als konstant über längere Zeitdauer angenommen. Dieser Abschnitt dient zunächst der Findung der Entwurfsparameter. In Abschnitt 5.4 wird anschließend auf das resultierende Systemverhalten weiter eingegangen.

Für den x - y -Beobachter stehen die Messwerte für die Linearbeschleunigung $\begin{bmatrix} \ddot{x}_{imu,b} & \ddot{y}_{imu,b} \end{bmatrix}^T$ aus dem Akzelerometer der IMU im Körperkoordinatensystem zur

Verfügung. Diese werden mittels der dreidimensionalen Rotationsmatrix $R_{b2w} = R(-\psi_{est}) \cdot R(-\theta_{est}) \cdot R(-\phi_{est})$ zu $\begin{bmatrix} \ddot{x}_{imu,w} \\ \ddot{y}_{imu,w} \end{bmatrix} = R_{b2w} \begin{bmatrix} \ddot{x}_{imu,b} \\ \ddot{y}_{imu,b} \end{bmatrix}$ ins Weltkoordinatensystem transformiert. Dabei repräsentiert jeweils eine Rotationsmatrix $R(arg_{est})$ die Drehung des Koordinatensystems um einen der drei Schätzungen für die Euler-Winkel, Gier, Nick oder Roll. Die so errechneten Linearbeschleunigungen werden dann als Systemeingang für das Teilsystem definiert.

Der optische Flusssensor liefert Werte für die Geschwindigkeit im Körperkoordinatensystem. Auch diese muss mittels Rotationsmatrix abhängig vom geschätzten Gierwinkel in das Weltkoordinatensystem überführt werden. Es ergibt sich

$$\begin{bmatrix} \dot{x}_{flow} \\ \dot{y}_{flow} \end{bmatrix} = \begin{bmatrix} \cos \psi_{est} & -\sin \psi_{est} \\ \sin \psi_{est} & \cos \psi_{est} \end{bmatrix} \begin{bmatrix} \dot{x}_{flow,b} \\ \dot{y}_{flow,b} \end{bmatrix}. \quad (5.20)$$

Der in Kapitel 4 entwickelte Algorithmus liefert zudem Messwerte $[x_{vis} \ y_{vis}]^T$, die bereits im Rahmen der Positionsbestimmung in das Weltkoordinatensystem umgerechnet wurden. Für den Fall, dass diese Messwerte als unbrauchbar erachtet werden, so werden die entsprechenden Messausgänge für diesen Zeitschritt von der Beobachterdynamik nicht berücksichtigt, worauf später noch eingegangen wird.

Das Teilsystem wird als System vierter Ordnung folgendermaßen dargestellt:

$$X_{p,k+1} = \begin{bmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} X_{p,k} + \begin{bmatrix} \frac{1}{2}T_s^2 & 0 \\ 0 & \frac{1}{2}T_s^2 \\ T_s & 0 \\ 0 & T_s \end{bmatrix} U_{p,k} + \omega_{p,k}, \quad (5.21)$$

$$Y_{p,k} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} X_{p,k} + \nu_{p,k}. \quad (5.22)$$

Es gilt dabei $X_{p,k} = [x \ y \ \dot{x} \ \dot{y}]^T$. Als Systemeingang $U_{p,k} = [\ddot{x}_{imu} \ \ddot{y}_{imu}]^T$ werden die x - und y -Beschleunigungsmessungen aus dem Akzelerometer verwendet. Für den Messausgang gilt $Y_{p,k} = [x_{vis} \ y_{vis} \ \dot{x}_{flow} \ \dot{y}_{flow}]^T$.

Das Teilsystem unterscheidet sich von dem in Kapitel 2 beschriebenen Modell. Hier wird es durch einen einfachen Doppelintegrierer beschrieben, während das in Kapitel 2 dargestellte Teilsystem zusätzlich durch die Kippwinkel ψ und θ beeinflusst wird. Die Verkopplung mit dem Kippwinkel-Teilsystem wird hier über den zusätzlichen Eingangsvektor der IMU-Messwerte für \ddot{x} und \ddot{y} ersetzt. Dieser Sachverhalt wird später in Abschnitt 5.4 im Zuge der Beobachterdynamik weiter analysiert und diskutiert.

Für die Rauschmatrizen und Anfangswerte werden

$$\begin{aligned}
 Q_p &= \begin{bmatrix} 10^{-5} & 0 & 0 & 0 \\ 0 & 10^{-5} & 0 & 0 \\ 0 & 0 & 10^{-2} & 0 \\ 0 & 0 & 0 & 10^{-2} \end{bmatrix} \\
 R_p &= \begin{bmatrix} 10^{-2} & 0 & 0 & 0 \\ 0 & 10^{-2} & 0 & 0 \\ 0 & 0 & 10^{-3} & 0 \\ 0 & 0 & 0 & 10^{-3} \end{bmatrix} \\
 X_{p,0} &= [0 \ 0 \ 0 \ 0]
 \end{aligned} \tag{5.23}$$

verwendet. Diese Werte haben sich als praxistauglich erwiesen.

Zusammenführen der Teilsystembeobachter

Die vorliegenden vier Teilsystembeobachter sollen nun zu einer einzigen Systembeschreibung zusammengefasst werden. Dies dient dem Zwecke der Übersicht und vereinfacht spätere Rechenoperationen. Das Zusammenführen der Teilsysteme ist unkompliziert, da die Teilsysteme voneinander entkoppelt sind. Es wird die Reihenfolge für die Zustände entsprechend Kapitel 2 festgelegt auf

$$x_{t,k} = [x \ y \ z \ \phi \ \theta \ \psi \ \dot{x} \ \dot{y} \ \dot{z} \ p \ q \ r]^T. \tag{5.24}$$

Die Systemmatrizen zur Bestimmung der Kalman-Verstärkung lauten somit:

$$\begin{aligned}
 A_B &= I_{12 \times 12} + \begin{bmatrix} O_{6 \times 6} & T_s \cdot I_{6 \times 6} \\ O_{6 \times 6} & O_{6 \times 6} \end{bmatrix} \\
 B_B &= \begin{bmatrix} \frac{1}{2} T_s^2 \cdot \beta \\ T_s \cdot \beta \end{bmatrix} \\
 \beta &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & J_x^{-1} & 0 & 0 \\ 0 & 0 & 0 & 0 & J_y^{-1} & 0 \\ 0 & 0 & 0 & 0 & 0 & J_z^{-1} \end{bmatrix} \\
 C_B &= \begin{bmatrix} I_{8 \times 8} & O_{8 \times 3} \\ O_{1 \times 8} & O_{1 \times 3} \\ O_{3 \times 8} & I_{3 \times 3} \end{bmatrix}^T
 \end{aligned} \tag{5.25}$$

Die Rauschmatrizen und der Anfangsvektor lauten

$$\begin{aligned}
 Q_B &= \text{diag}(10^{-5}, 10^{-5}, 10, 10^{-4}, 10^{-4}, 10^{-6}, 10^{-2}, 10^{-2}, 10^{-4}, 10^{10}, 10^{10}, 10^{-4}) \\
 R_B &= \text{diag}(10^{-2}, 10^{-2}, 10^{-1}, 10^{-5}, 10^{-5}, 5 \cdot 10^{-3}, 10^{-3}, 10^{-3}, 10^{-10}, 10^{-10}, 10^{-5}) \\
 X_{0,B} &= [0 \ 0 \ -0.5 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0]^T.
 \end{aligned} \tag{5.26}$$

Für Zustandsvektor, Eingangsvektor sowie Messgrößenvektor gelten

$$\begin{aligned}
 X_k &= [x \ y \ z \ \phi \ \theta \ \psi \ \dot{x} \ \dot{y} \ \dot{z} \ p \ q \ r]^T \\
 U_k &= [\ddot{x}_{imu} \ \ddot{y}_{imu} \ \ddot{z}_{imu} \ \tau_x \ \tau_y \ \tau_z]^T \\
 Y_k &= [x_{vis} \ y_{vis} \ z_{son} \ \phi_{imu} \ \theta_{imu} \ \psi_{vis} \ \dot{x}_{flow} \ \dot{y}_{flow} \ p_{imu} \ q_{imu} \ r_{imu}]^T.
 \end{aligned} \tag{5.27}$$

nach Abschnitt 5.2 ergibt sich für die zeitinvariante Beobacherverstärkung

$$L = \begin{bmatrix} 0.03 & 0 & 0 & 0 & 0 & 0 & 0.0004 & 0 & 0 & 0 & 0 \\ 0 & 0.03 & 0 & 0 & 0 & 0 & 0 & 0.0004 & 0 & 0 & 0 \\ 0 & 0 & 0.99 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.92 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.92 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.01 & 0 & 0 & 0 & 0 & 0.0004 \\ 0.00004 & 0 & 0 & 0 & 0 & 0 & 0.91 & 0 & 0 & 0 & 0 \\ 0 & 0.00004 & 0 & 0 & 0 & 0 & 0 & 0.912 & 0 & 0 & 0 \\ 0 & 0 & 0.003 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.92 \end{bmatrix}. \quad (5.28)$$

Für den Fall, dass eine bestimmte Messung zum Zeitpunkt k nicht vorliegt, so soll der Beobachter für diesen Zeitpunkt und Zustand lediglich den Simulatorterm berechnen und den Korrekturterm ignorieren. In Simulink wurde diese Funktionalität implementiert, indem dem Beobachter ein *enable*-Vektor der Länge n zugeführt wird, wobei n die Anzahl der Messgrößen bezeichnet. Jeder Eintrag dieses *enable*-Vektors enthält ein logisches Bit, das den Korrekturterm $L(Y - \hat{Y})$ für die jeweilige Messung berücksichtigt bzw. nullt. Da jede Zustandsschätzung von höchstens einer Messung im Korrekturterm beeinflusst wird, ist es zulässig, einfach den kompletten Korrekturterm für die zugehörige Schätzung zu Null zu setzen. Diese Implementierung ist analog zu der Funktionalität des Simulink-Blocks `Kalman Filter`.

5.4 Dynamik des über Beobachter geschlossenen Regelkreises

Um die Gesamtdynamik zu vereinfachen, ist es sachdienlich, zunächst den Teilsystembeobachter für die x - y -bezogenen Zustände weiter zu beleuchten. Die Systemgleichung des in Abschnitt 5.3 entwickelten Kalman-Filters ergibt sich nach Abschnitt 5.2 zu

$$\hat{X}_{k+1} = A_B \hat{X}_k + B_B U_{B,k} + L(Y_k - \hat{Y}_k). \quad (5.29)$$

A_B und B_B bezeichnen die für den Beobachterentwurf verwendeten A - und B -Matrizen entsprechend Gleichung 5.25. Die erweiterten Systemzustände werden nicht

vom Beobachter rekonstruiert, da sie durch einfache Integration ermittelt werden. Es wird zudem die Annahme getroffen, dass die Messausgänge des Systems Y_k zu jedem Zeitpunkt vorliegen und plausibel sind. Für den Eingangsvektor des Simulatorterms des Beobachters U_B gilt:

$$U_B = [\ddot{x}_{imu,w} \quad \ddot{y}_{imu,w} \quad \ddot{z}_{imu,w} \quad \tau_x \quad \tau_y \quad \tau_z]^T. \quad (5.30)$$

Da annäherungsweise $z_{imu,w} = \tau_{tot}$ gilt, kann der Eingangsvektor des Beobachters in Termen des in Kapitel 2 verwendeten Eingangsvektors U_S folgendermaßen ausgedrückt werden:

$$U_B = [U_{imu}(X) \quad U_S(\hat{X})], \quad (5.31)$$

wobei U_{imu} die ergänzenden Messgrößen aus der IMU bezeichnet. Da der Eingang U_{imu} nicht vom Regler gebildet wird sondern sich aus Messwerten für die Kippwinkel ergibt, ist er direkt von X abhängig, anstatt von \hat{X} . Bild 5.2 veranschaulicht diesen Sachverhalt. K ist dabei der Regler, S das Quadroptersystem und L der Beobachter. CC verkettet die beiden Eingangsvektoren U_S und U_{imu} .

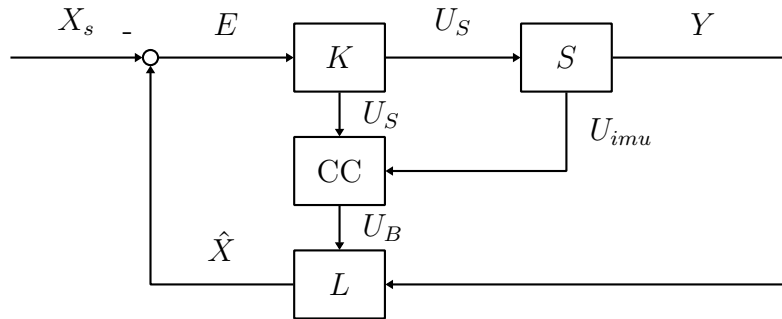


Bild 5.2: Blockschaltbild für das Gesamtsystem

Das Verhalten des Simulatorterms mit den Matrizen A_B und B_B aus Abschnitt 5.3 soll dem Verhalten des Modell-Systems

$$X_{k+1} = A_S(\psi)X_k + B_S U_{S,k}, \quad (5.32)$$

mit den Matrizen A_S und B_S aus Kapitel 2 gleichen. Dass dies der Fall ist, lässt sich sehen, indem man die zerlegten Matrizen aus jeweils Gleichung 5.25 für den Simulatorterm des Beobachters und Gleichung 2.15 für die zeitdiskretisierte Systemdynamik des Quadropters vergleicht. Es stellt sich heraus, dass sich die Dynamikmatrix A_S

den zusätzlichen Summanden

$$\begin{bmatrix} O_{2 \times 3} & \frac{1}{2}T_s^2\Psi_d & O_{2 \times 4} & O_{2 \times 2} & O_{2 \times 1} \\ O_{4 \times 3} & O_{4 \times 2} & O_{4 \times 4} & O_{4 \times 2} & O_{4 \times 1} \\ O_{2 \times 3} & T_s\Psi_d & O_{2 \times 4} & \frac{1}{2}T_s^2\Psi_d & O_{2 \times 1} \\ O_{4 \times 3} & O_{4 \times 2} & O_{4 \times 4} & O_{4 \times 2} & O_{4 \times 1} \end{bmatrix} \quad (5.33)$$

aufweist. Dieser Term reflektiert die gierwinkelabhängige Verkopplung des x - y -Teilsystems mit dem Kippwinkel-Teilsystem. Der Eingangsvektor U_B enthält gegenüber dem Eingang U_S den zusätzlichen Anteil aus der IMU U_{imu} , welcher sich über die Beziehung 5.12 unmittelbar aus den Kippwinkeln ergibt. Somit wird die Kopplung der Teilsysteme nicht über die Dynamikmatrix A_B sondern über die erweiterte Eingangsmatrix B_B erreicht. Aus der Simulation geht ebenfalls hervor, dass die Systemverhalten identisch sind, d.h. es gilt zu jedem Zeitpunkt k

$$A_B\hat{X} + B_B U_B = A_S\hat{X} + B_S U_S \quad (5.34)$$

Die Gierwinkelabhängigkeit der verbleibenden Dynamik verschwindet analog zu Abschnitt 3.3. Es verbleibt für einen Sollwert $X_s = 0$ folgende Dynamik für das Gesamtsystem entsprechend [FKL⁺13]:

$$\begin{bmatrix} \hat{X}_{k+1} \\ X_{k+1} \end{bmatrix} = \begin{bmatrix} A - LC^T - BK & LC^T \\ -BK & A \end{bmatrix} \begin{bmatrix} \hat{X}_k \\ X_k \end{bmatrix}. \quad (5.35)$$

Die Matrizen A und B sind dabei die Systemmatrizen des linearisierten Systems für eine Wahl des Gierwinkels zu $\psi_b = 0$.

5.5 Simulative Erprobung

Die in Abschnitt 3.4 eingeführte Simulation wird nun erweitert. Um die Beobachterfunktionalität zu testen, werden die Zustände des Kopters mit einem Messrauschen beaufschlagt. Die IMU-Messwerte werden aus den Kippwinkeln rekonstruiert. Diese simulierten Messwerte werden dann an den zu prüfenden Beobachter gegeben. Der Regelkreis wird über die Zustandsschätzung \hat{X} geschlossen.

Die Aussagekraft dieser Simulation ist stark limitiert, da das Verhalten der Sensoren

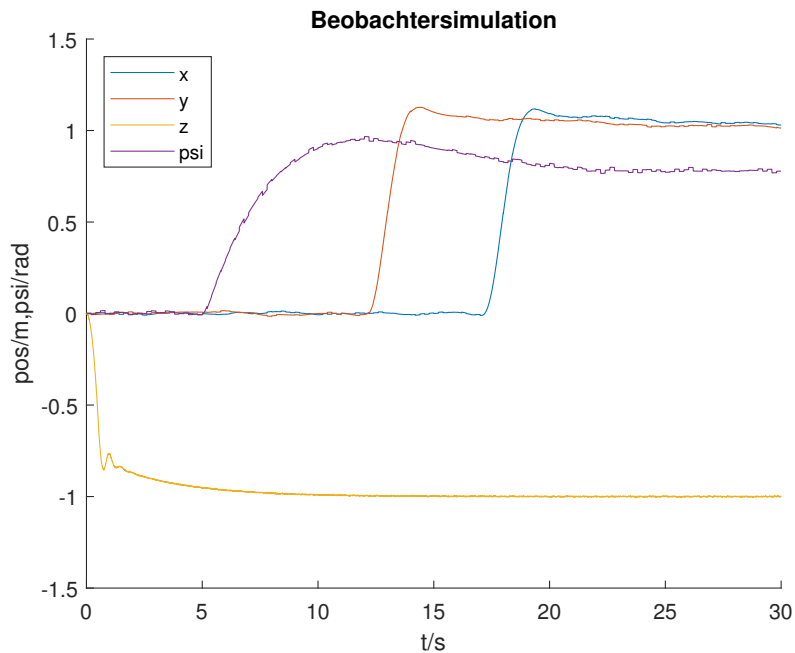
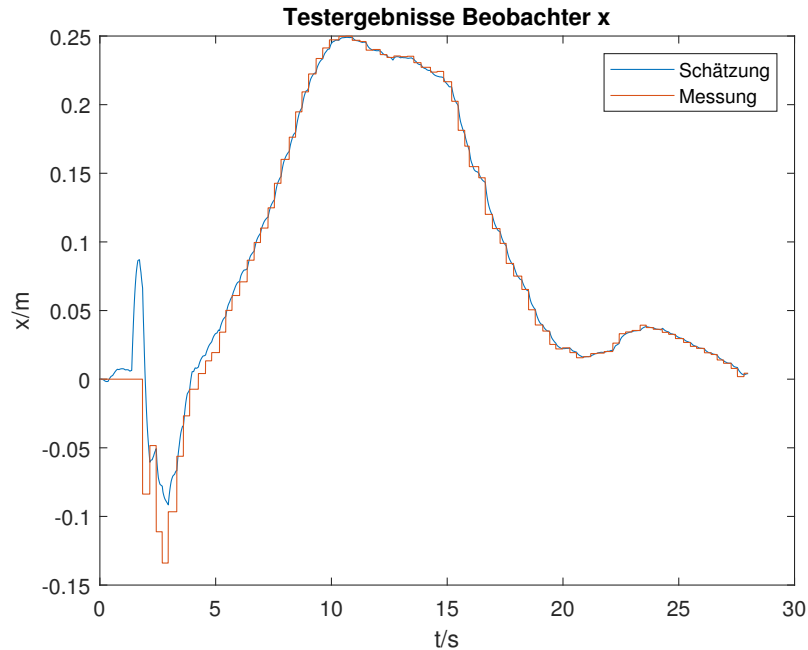


Bild 5.3: Simulation des Trajektorienverlaufs ausgewählter Zustände für den über den Beobachter geschlossenen Regelkreis

extrem vereinfacht werden musste. Der Hauptzweck war es, Fehler in der Implementierung aufzudecken. Das Ermitteln geeigneter Parameter für den Beobachterentwurf musste am echten System erfolgen. In Bild 5.3 ist der Zustandsverlauf einer beispielhaften Trajektorie darstellt, um zu zeigen, dass der Beobachter und das vereinfachte Gesamtsystem unter beliebiger Wahl des Gierwinkels asymptotisch stabil sind.

5.6 Erprobung am Parrot Mambo

Vordergründig gilt es nun, zu überprüfen, ob der Integrationsdrift durch die Kameraauswertung eliminiert werden konnte. Um zu testen, ob der Beobachter akkurate Absolutwerte für x , y und ψ gewinnen kann, wird der Quadrokopter aus einer Startposition heraus gestartet, die von $(x, y, \psi) = (0, 0, 0)$ abweicht. Findet der Kopter konsistent selbstständig in die Position, die im Weltkoordinatensystem als $(0, 0, 0)$ gilt, so funktionieren der Kameraauswertungsalgorithmus und der zugeordnete Teilbeobachter. $(0, 0, 0)$ wurde in Abschnitt 4.2 definiert als Position direkt über der Bodenmarkierung mit einem Gierwinkel, sodass die blaue Fläche im Kamerabild rechts von der roten Fläche erscheint.

Bild 5.4: Schätzung auf Basis der Kameraauswertung, x

Dieser Versuch wurde mehrfach durchgeführt. Damit der Kopter zur Ursprungsposition findet, darf der tatsächliche Anfangswert für den Gierwinkel nicht zu stark vom Sollwert für diesen abweichen. Außerdem müssen beide Flächen der Bodenmarkierung für die Kamera sichtbar sein. Wenn der Anfangswert für den Gierwinkel nur geringfügig von Null abweicht, dann ist es sogar hinreichend, wenn nur eine Teilfläche der Markierung sichtbar ist. In einem solchen Fall steuert der Kopter auf die Markierung zu, entdeckt die zweite Teilfläche, und kann somit ebenfalls die Gierwinkelschätzung korrigieren.

In Anhang D findet sich ein Video eines solchen Versuches, bei dem der *Parrot Mambo* aus einer willkürlichen Anfangsposition startet, aus der er dann den Ursprung anfliegt. Bilder 5.4 bis 5.6 zeigen Messdiagramme für diesen Flugtest, welche die Messungen der Kameraauswertung und die entsprechenden Zustandsschätzungen zeigen.

Bei dem Gierwinkel-Beobachter wird in Bild 5.6 sichtbar, dass der Schätzwert sich nur langsam dem Messwert annähert. Dies hat erfahrungsgemäß zu einem besseren Flugverhalten geführt, insbesondere dann, wenn die Gierwinkelmessung nur sporadisch vorlag. Bilder 5.4 und 5.5 zeigen, dass x - und y -Wert plausibel gemessen werden.

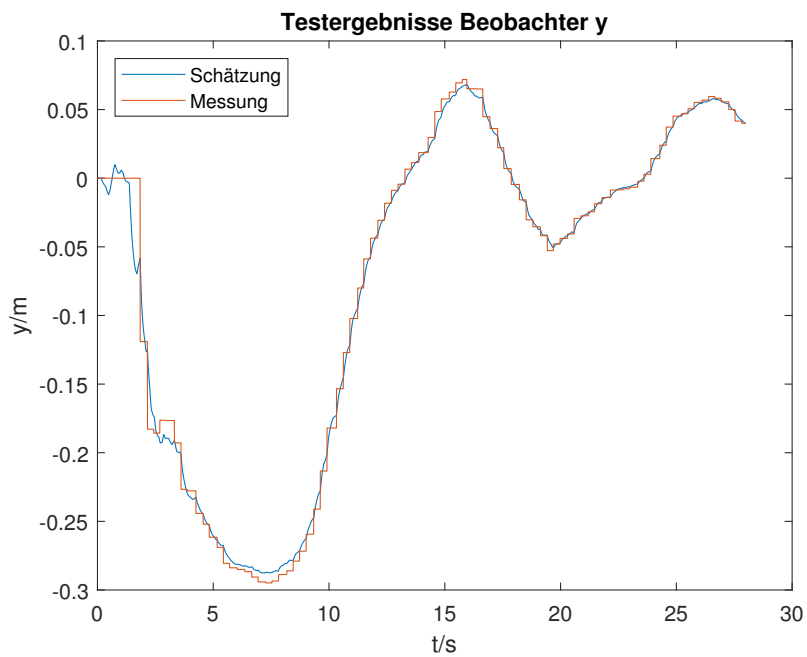


Bild 5.5: Schätzung auf Basis der Kameraauswertung, y

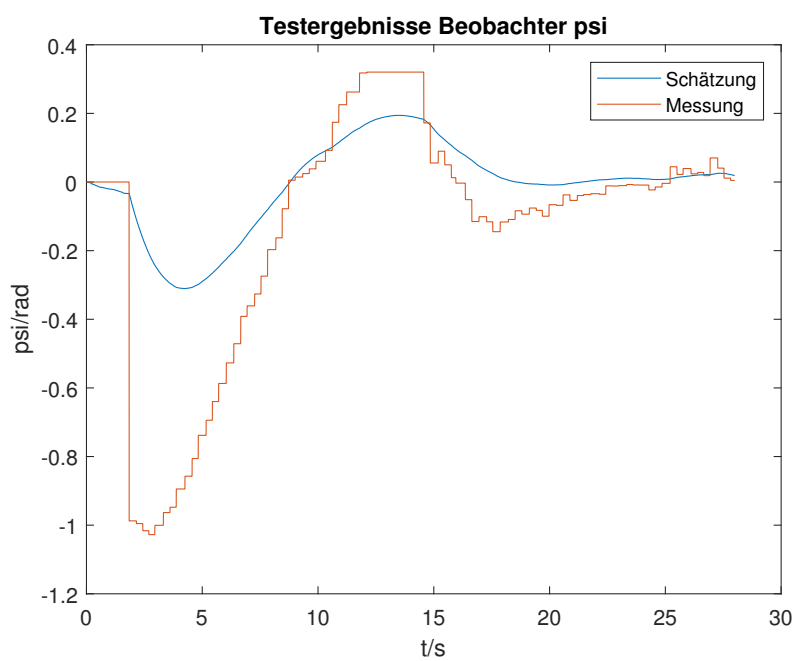


Bild 5.6: Schätzung auf Basis der Kameraauswertung, ψ

Kapitel 6

Sicherheitsüberwachung

Im Folgenden soll eine Sicherheitsüberwachung entwickelt werden. Diese dient dem Zweck, neue Regler erproben zu können, ohne dabei zu riskieren, dass der Kopter zu Schaden kommt. Die verwendete Architektur basiert auf den in [SS99] und [Apf] vorgestellten Konzepten.

Die Architektur beinhaltet einen bereits erprobten, sicheren, sowie einen experimentellen, gegebenenfalls unsicheren Regler. Beide Regelgesetze sollen zur Laufzeit evaluiert werden. Im Normalfall werden die Ausgänge des experimentellen Reglers auf den Systemeingang gegeben. Droht das System allerdings, in einen unsicheren Zustand überzugehen, soll der sichere Regler übernehmen und das System in seine Ruhelage zurückführen.

Um Stabilitätsbetrachtungen anstellen zu können, muss zunächst eine Beschreibung für das Gesamtsystem festgelegt werden. Anschließend wird ein zulässiger Zustandsraum auf Basis von Modellgrenzen und physikalischen Restriktionen etabliert. Dieser wird dann in einem zweiten Schritt weiter eingeschränkt, sodass eine Zustandsmenge vorliegt, in der sich ausschließlich Zustände befinden, deren weitere Trajektorienverläufe innerhalb ebendieser Menge verbleiben. Diese sichere Zustandsmenge kann dann verwendet werden, um einen kritischen Fall zu identifizieren und auf den sicheren Regler umzuschalten. Um ungewolltes, hochfrequentes Schalten zu vermeiden, wird eine gemäß der in [Apf] vorgestellten Umschaltlogik in angepasster Form implementiert. Die Konzepte werden simulativ erprobt. Anschließend werden Experimente an dem physikalischen Quadrokopter *Parrot Mambo* durchgeführt.

6.1 Dynamik des autonomen Systems

Die Systemdynamik des über den Beobachter geschlossenen Regelkreises wurde in Abschnitt 5.4 erörtert. Um eine Lyapunov-Stabilitätsanalyse für das Gesamtsystem durchführen zu können, soll nun daraus nach [SS99] ein autonomes, zeitdiskretes, lineares, zeitinvariantes System der Form

$$X_{k+1} = AX_k. \tag{6.1}$$

herausgearbeitet werden. Der Zustandsvektor X muss vollständig vorliegen.

Dafür müssen einige Vereinfachungen getroffen werden. Zunächst soll lediglich die Dynamik des geschlossenen Regelkreises verwendet werden. Die Beobachterdynamik soll dabei vorerst unberücksichtigt bleiben. Dafür muss die Annahme gelten, dass die Dynamik des Beobachterfehlers deutlich schneller ist, als die Dynamik des Zustandsfehlers, und sie diese somit qualitativ nicht beeinflusst [FKL⁺13]. Es gilt also im Weiteren die vereinfachende Annahme $X = \hat{X}$. Es wird dafür später bei der Definition des sicheren Zustandsraums eine Reserve für eventuelle durch die Beobachterdynamik hervorgerufene Verzögerungen mit eingeplant. Die erweiterten Integralzustände bleiben ebenfalls für Stabilitätsuntersuchungen zunächst unberücksichtigt.

Im Weiteren sollen für alle in diesem Kapitel angestellten Betrachtungen die Zustandsfehlergrößen verwendet werden. Die **Fehlerdynamikmatrix** des geschlossenen Regelkreises ergibt sich nach Abschnitt 3.3 sich zu

$$A_g = A - BK, \tag{6.2}$$

wobei A die diskrete, nicht-erweiterte Dynamikmatrix des Quadroptersystems ist, B die Eingangsmatrix und K die Zustandsfehlerückführung des in Kapitel 3 entworfenen Linear Quadratic Regulators ohne die I-Zustände darstellt. Die Systemmatrix A_g beschreibt dann die Dynamik des Zustandsfehlers im geschlossenen Regelkreis, d.h. es gilt:

$$E_{k+1} = A_g E_k, \tag{6.3}$$

wobei $E_k = \hat{X} - X_s$ definiert ist als die Abweichung des geschätzten Systemzustands vom vorgegebenen Sollwert zum Zeitpunkt k . Die erweiterten Zustände bleiben unbe-

rücksichtigt, d.h. die Systemordnung beträgt 12. Um die konventionelle Darstellung für Lyapunov-Funktionen zu ermöglichen, soll für den Rest des Kapitels der betrachtete Zustandsvektor E stattdessen als x bezeichnet werden.

6.2 Definition eines zulässigen Zustandsraums

Es wird zunächst ein zulässiger Zustandsraum \mathcal{X} definiert. Dieser Zustandsraum wird durch Schranken begrenzt, die einerseits direkt aus den Zustandsgrößen und andererseits aus der Stellsignalbegrenzung hervorgehen.

Schranken aus Zustandsbegrenzungen

Zunächst werden jene Zustandsbegrenzungen betrachtet, die durch Grenzen des linearisierten Modells oder durch den eingeschränkten Betriebsbereich des Beobachters gegeben sind. Dazu gehören

- Mindest- und Maximalflughöhe des Quadropters
- Grenzen der Systemlinearisierung, insbesondere Kippwinkel
- Grenzen der Flugzone innerhalb des Innenraums in x - y -Richtung
- Grenzen der Kameraauswertung in x - y -Richtung.

Diese Schranken werden in der Form von p_x Ungleichungen

$$a_i^T x \leq 1, i = 1, \dots, p_x \tag{6.4}$$

angegeben. x ist dabei die Bezeichnung für den Zustandsfehlervektor.

Da sich die Schranken auf den Zustandfehler beziehen, verschieben sich bei Änderung der Sollgröße sich die definierten Schranken in Bezug auf die absolute Zustandsgröße. Diese Tatsache muss bei der Wahl der Schranken und bei einer Trajektorienplanung berücksichtigt werden.

Schranken aus Stellsignalbegrenzung

Der zulässige Zustandsraum soll zudem von den Stellsignalbegrenzungen eingegrenzt werden. Die sich so ergebenden Schranken können ebenfalls in einer Form wie in Gleichung 6.4 angegeben werden.

Um die Schranken aus der Stellsignalbegrenzung in der Form von Gleichung 6.4 in Bezug auf die Zustandsgrößen anzugeben, werden sie zunächst entsprechend umgerechnet. Ursprünglich bezieht sich die obere Stellsignalbegrenzung auf diejenigen Signale, mit denen die vier Motoren angesteuert werden. Es gilt also

$$m_i \leq m_{max}, i = 1, 2, 3, 4, \quad (6.5)$$

wobei m_i die Motorsignale sind und m_{max} der numerische Wert für die obere Stellsignalbegrenzung, die für alle Motoren identisch ist. Diese Schranken sollen zunächst in die Form

$$g_{u,i}U \leq 1, i = 1, \dots, p_u \quad (6.6)$$

umgerechnet werden. Mit der Transformation des Eingangsvektors zu Motorsignalen $m = cT^{QT}U$, wobei T^{QT} die Transformationsmatrix und c die Umrechnungskonstante bezeichnet [Yan], lässt sich die Matrix

$$G_u = T^{QT} \frac{c}{m_{max}} \quad (6.7)$$

berechnen. Die Zeilen $g_{u,i}$ von G_u entsprechen dann den $p_u = 4$ einzelnen Schranken für U aus Gleichung 6.6.

Diese p_u Schranken in U können nun mithilfe des Regelgesetzes $U = Kx$ nach Gleichung 6.8 zu p_u Schranken im Zustandsfehler x umgerechnet werden.

$$a_{p_x+i} = g_{u,i}K, i = 1, \dots, p_u \quad (6.8)$$

Es ergeben sich damit aus den Schranken für die Zustände und die Stellsignale insgesamt $p = p_x + p_u$ Ungleichungen. Der zulässige Zustandsraum definiert sich wie folgt:

$$\mathcal{X} = \{x \in \mathbb{R}^n | a_i^T x \leq 1\}, i = 1, \dots, p \quad (6.9)$$

Die untere Stellsignalbegrenzung wird für die Bestimmung des sicheren Zustands-

raums zunächst nicht berücksichtigt, da ihr Erreichen keine Probleme herbeiführt, die den Kopter beschädigen könnten. In der Regel wird sie nur dann erreicht, wenn der Kopter die gewünschte Flughöhe überschreitet, und daher aus der Perspektive des linear entworfenen Reglers ein negativer Gesamtmotorschub nötig wäre. Dies entspricht keinem Verlassen des sicheren Zustandsraums. Da die Dynamik des Gesamtsystems sich jedoch ändert, muss dieser Sachverhalt in einer späteren Arbeit weiter untersucht werden. Ein Modell der Schwerkraft könnte hier Abhilfe schaffen.

Durch die Vernachlässigung der erweiterten Integralzustände kann es bei starker Modellunsicherheit dazu kommen, dass die Stellsignalbegrenzung tatsächlich schon erreicht wird, bevor der zulässige Zustandsraum verlassen wird. Dieses Problem muss ebenfalls in späteren Arbeiten adressiert werden.

6.3 Lyapunov-Stabilitätskriterium im zeitdiskreten Fall

Innerhalb von \mathcal{X} existieren nun allerdings noch Zustände, dessen weiterer Trajektorienverlauf \mathcal{X} verlassen würde. Unter Berücksichtigung der Einschränkungen, die \mathcal{X} an den Zustandsraum stellt, soll nun also nach [SS99] eine sogenannte *positiv invariante Teilmenge* \mathcal{W} ermittelt werden. Diese Teilmenge wird als *wiederherstellbarer Zustandsraum* bezeichnet. Die positive Invarianz von \mathcal{W} impliziert, dass jegliche Trajektorie, die innerhalb von \mathcal{W} startet für alle Zeitpunkte innerhalb dieser Menge verbleiben wird.

Im Folgenden soll der Begriff der positiven Invarianz geklärt werden. Außerdem wird hergeleitet, wie eine solche positiv invariante Teilmenge ermittelt werden kann. Dafür kommt das zeitdiskrete Lyapunov-Stabilitätskriterium nach [BCS18] zum Einsatz.

Lyapunov-Stabilitätskriterium

Es sei gegeben ein autonomes, zeitdiskretes System der Form

$$x_{k+1} = Ax_k, \tag{6.10}$$

Dann kann das sogenannte Lyapunov-Kriterium verwendet werden, um den folgenden Stabilitätsbegriff nachzuweisen:

Gegeben sei ein Gleichgewichtspunkt $x = 0$. Dieser Gleichgewichtspunkt ist asymptotisch stabil, wenn ein $\delta > 0$ definiert werden kann, sodass gilt:

$$\|x(0)\| < \delta \Rightarrow \lim_{t \rightarrow \infty} x(t) = 0 \quad (6.11)$$

Anschaulich bedeutet dies, dass wenn ein Gleichgewichtspunkt asymptotisch stabil ist, der Zustandsvektor gegen den Gleichgewichtspunkt strebt, unter der Voraussetzung, dass die Vektornorm der Anfangswerte von $x(t)$ sich innerhalb eines bestimmten Rahmens befindet.

Das Lyapunov-Kriterium besagt, dass ein Gleichgewichtspunkt dann asymptotisch stabil ist, wenn eine sogenannte Lyapunov-Funktion $V(x)$ existiert, die gewisse Bedingungen erfüllt, welche im Folgenden erörtert werden.

Es sei eine stetige Funktion $V : \mathbb{R}^n \rightarrow \mathbb{R}$ gegeben, wobei n die Systemordnung des zeitdiskreten Systems bezeichnet. Diese Funktion ist eine Lyapunov-Funktion und der Gleichgewichtspunkt ist asymptotisch stabil genau dann, wenn folgende Kriterien zutreffen:

- $V(0) = 0$
- $V(x_k) > 0, \forall x \in \mathbb{R}_{\neq 0}^n$
- $V(x_{k+1}) - V(x_k) < 0, \forall x \in \mathbb{R}_{\neq 0}^n$

Existiert eine solche Lyapunov-Funktion, so ist der Gleichgewichtspunkt asymptotisch stabil. Dieses Kriterium erinnert an die zeitkontinuierliche Entsprechung des Lyapunov-Kriteriums, bei dem die Ableitung einer Lyapunov-Funktion $\dot{V}(x)$ negativ definit sein muss.

Invarianzprinzip

Das Invarianzprinzip wird mithilfe der Lyapunov-Funktion formuliert. Es verhält sich im Zeitdiskreten analog zum Zeitkontinuierlichen. Die Bedeutung des Invarianzprinzips wird im Folgenden erörtert.

Es sei eine Menge M . Diese Menge M ist dann positiv invariant in Bezug auf ein System, wenn

$$x(0) \in M \Rightarrow x(t) \in M, \forall t \geq 0. \quad (6.12)$$

Das bedeutet, wenn der Zustand sich zu einem bestimmten Zeitpunkt in M befindet, dann verbleibt der Zustand für alle Zeiten in M . Diese Eigenschaft entspricht genau der Forderung, die an einen sicheren Zustandsbereich gestellt wurde; zusammen mit der asymptotischen Stabilität des Gleichgewichtspunktes wird ein Zustand innerhalb einer positiv invarianten Menge in jedem Fall zu $x = 0$ zurückkehren.

Existiert für ein dynamisches System nun eine Lyapunov-Funktion $V(x)$, so ist

$$M = \{x \in \mathbb{R}^n | V(x) < c\} \tag{6.13}$$

eine positiv invariante Menge für das System, auf welches sich die Lyapunov-Funktion bezieht. Dies bedeutet anschaulich, dass der Rand $V(x) < c$ von einer Zustandstrajektorie nicht überschritten werden kann.

Auf Basis dieser Erkenntnisse lässt sich nun die zulässige Zustandsmenge \mathcal{X} geeignet einschränken, sodass eine wiederherstellbare Zustandsmenge \mathcal{W} hervorgeht, welche eine Teilmenge von \mathcal{X} ist.

6.4 Berechnung des wiederherstellbaren Zustandsraums

Wahl der Lyapunov-Funktion

Um eine positiv invariante Menge zu erhalten, muss zunächst eine konkrete Lyapunov-Funktion für das betrachtete Zustandsfehlersystem aus Gleichung 6.10 ermittelt werden. Im Zeitdiskreten kann analog zum Zeitkontinuierlichen folgender Ansatz als Kandidat für eine Lyapunov-Funktion verwendet werden:

$$V(x) = x^T P x > 0, \forall x \neq 0. \tag{6.14}$$

Hierbei muss P eine symmetrische, positiv semidefinite Matrix sein. Um die letzte Bedingung an eine zeitdiskrete Lyapunov-Funktion zu erfüllen, muss gelten

$$V(x_{k+1}) - V(x_k) = x^T (A_g^T P A_g - P) x < 0. \tag{6.15}$$

D.h. $A_g^T P A_g - P$ muss als negativ definite Matrix gewählt werden können, damit der Gleichgewichtspunkt als asymptotisch stabil nachweisbar ist. Per Reglerentwurf ist der durch den LQR geschlossene Regelkreis asymptotisch stabil, demnach muss es P geben, sodass $V(x)$ eine Lyapunov-Funktion ist.

Wahl der Matrix P

Die Wahl der Matrix P zur Bestimmung der Lyapunov-Funktion soll derart erfolgen, dass für eine invariante Zustandsmenge

$$\mathcal{W} = \{x \in \mathbb{R}^n | V(x) < 1\} \tag{6.16}$$

gerade noch die für \mathcal{X} definierten Schranken respektiert werden. Dies stellt sicher, dass die wiederherstellbare Zustandsmenge \mathcal{W} die größtmögliche Teilmenge der zulässigen Zustandsmenge \mathcal{X} ist.

In [BEGFB94] ergibt sich, dass die Menge \mathcal{W} unter Def. 6.16 genau dann eine Teilmenge von \mathcal{X} ist, wenn

$$a_k^T P^{-1} a_k \leq 1, k = 1, \dots, p \tag{6.17}$$

gilt. Dabei sind a_k die für den zulässigen Zustandsraum definierten Schranken. Dieses Lemma wird in [SS99] für den zeitkontinuierlichen Entwurf etabliert, allerdings gilt es auch für die Betrachtung im zeitdiskreten Fall, da die Systemdynamik für diese Aussage nicht berücksichtigt wird.

Dies bedeutet anschaulich, dass für Funktionswerte $V(x) < 1$ der Lyapunov-Funktion mit der Matrix P gilt, dass sich der Zustandsvektor X innerhalb der *wiederherstellbaren Zustandsmenge* befindet. Um einen Sicherheitspuffer zu gewährleisten, wird später $V(x) < 0.9$ als Grenze für den *sicheren Zustandsraum* gewählt.

Die positiv invariante Zustandsmenge \mathcal{W} definiert im Zustandsraum einen Ellipsoiden. Die größtmögliche Menge \mathcal{W} maximiert also das Volumen dieses Ellipsoiden. Es wird sich außerdem zunutze gemacht, dass das Volumen dieses Ellipsoiden proportional zu $\sqrt{\det P^{-1}}$ ist. Das Optimierungsproblem ließe sich demnach, analog zu [SS99] vorläufig folgendermaßen formulieren:

$$\begin{aligned}
 &\text{minimiere} && \log(\det(P)) \\
 &\text{u.B.v} && P > 0 \quad (1) \\
 &&& A_g^T P A_g - P < 0 \quad (2) \\
 &&& a_k^T P^{-1} a_k \leq 1, k = 1, \dots, p \quad (3)
 \end{aligned}$$

Dieses Optimierungsproblem minimiert die Determinante von P , maximiert also das Volumen der resultierenden Menge \mathcal{W} . Bedingungen (1) und (2) müssen erfüllt sein, sodass $V(x) = x^T P x$ eine Lyapunov-Funktion für ein zeitdiskretes System, und somit \mathcal{W} eine positiv invariante Menge ist. Bedingung (3) stellt sicher, dass \mathcal{W} eine Teilmenge von \mathcal{X} ist, d.h. die definierten Zustandsschranken einhält.

Bedingung (3) bezieht sich allerdings auf die Inverse P^{-1} der gesuchten Matrix. Es liegt also keine lineare Matrixungleichung (LMI) vor. Numerische Methoden wie beispielsweise die *Interior Point Method* [BEGFB94] stellen gewisse Anforderungen an die Bedingungen. Das Auftreten der Inversen der gesuchten Variable ist dabei unzulässig. In dieser Form kann das Optimierungsproblem also nicht gelöst werden.

Ermitteln der Matrix P im Zeitkontinuierlichen

Um das Optimierungsproblem in eine lösbare Gestalt zu überführen, soll das System des geschlossenen Regelkreises ins Zeitkontinuierliche überführt werden. Dazu kommt der MATLAB-Befehl `d2c(sys)` zum Einsatz. Dieser erzeugt mittels Abtast-Halte-Methode ein zeitkontinuierliches System, das dem ursprünglichen zeitdiskreten System annäherungsweise äquivalent ist. Da es sich um ein autonomes System handelt, und lediglich die Dynamikmatrix A von Interesse ist, werden keine Parameter für die Matrizen B , C und D übergeben [Lun16].

Es ergibt sich für das zeitkontinuierliche System des geschlossenen Regelkreises folgende Zustandsgleichung mit der Dynamikmatrix A_c :

$$\dot{x} = A_c x. \tag{6.18}$$

Dabei beschreibt x weiterhin den Zustandsfehler des Quadroptersystems.

Die bisherigen Betrachtungen müssen lediglich in einem Punkt modifiziert werden; für die Lyapunov-Funktion muss nun analog zu Gleichung 6.15 gelten

$$\dot{V}(x) = x^T (P A_c + A_c^T P) x < 0, \tag{6.19}$$

also muss die Matrix $PA_c + A_c^T P$ negativ definit sein. Durch Multiplikation von P^{-1} von sowohl links als auch rechts ergibt sich nach $P^{-1}PA_cP^{-1} + P^{-1}A_cPP^{-1} < 0$ die Bedingung $A_cP^{-1} + P^{-1}A_c < 0$. Substitution von P^{-1} zu Q und Berücksichtigung der Bedingungen Gleichung 6.14 und Gleichung 6.17 resultieren in folgendem Optimierungsproblem von Q in lösbarer LMI-Form:

$$\text{maximiere} \quad \log(\det(Q))$$

$$\text{u.B.v} \quad Q > 0 \quad (1)$$

$$A_cQ + QA_c < 0 \quad (2)$$

$$a_k^T Q a_k \leq 1, k = 1, \dots, p \quad (3).$$

Das Optimierungsproblem ist analog zu dem in [Apf] und kann ebenfalls mit dem *CVX-Solver*, einer MATLAB-Bibliothek gelöst werden. Die Optimierungsvariable $Q = P^{-1}$ ergibt die gesuchte Lyapunov-Funktion $V(x) = x^T P x$. Die so ermittelte Lyapunov-Funktion kann dann auf die zeitdiskrete Systembetrachtung übertragen werden.

Anschaulich lässt sich dies erklären, indem man die Systemtrajektorien des jeweils zeitdiskreten bzw. zeitkontinuierlichen Systems für einen gegebenen Anfangswert x_0 betrachtet. Diese sind annäherungsweise identisch. Die für das zeitkontinuierliche System gültige Lyapunov-Funktion sollte daher beim Übergang zur zeitdiskreten Betrachtung bei hinreichend kleiner Abtastperiode ihre Gültigkeit behalten.

Die für den Quadrocopter *Parrot Mambo* vorläufig gewählten Schranken ergeben mit der in Kapitel 2 ermittelten Systembeschreibung und dem in Kapitel 3 entworfenen Regler sowie dem in Kapitel 5 entworfenen Beobachter somit eine Lyapunov-Funktion, für die numerisch gezeigt werden kann, dass die drei an die **zeitdiskrete Lyapunov-Funktion gestellten Bedingungen** in Gleichung 6.14, Gleichung 6.15 und Gleichung 6.17 **erfüllt** sind. Da die P-Matrix komplett besetzt ist, wird hier auf eine Darstellung verzichtet. Die Matrix wird im MATLAB-Skript `calculateInvariantSet` berechnet, welches sich in Anhang B befindet.

Um Fehler und Unsicherheiten weiter zu verringern und gegebenenfalls eine größere invariante Zustandsmenge zu erhalten, könnte das Optimierungsproblem direkt im Zeitdiskreten erfolgen. Dafür wäre eine Umformulierung des Optimierungsproblems aus Abschnitt 6.4 in zwei Sub-Probleme ein möglicher Ansatz [BEGFB94].

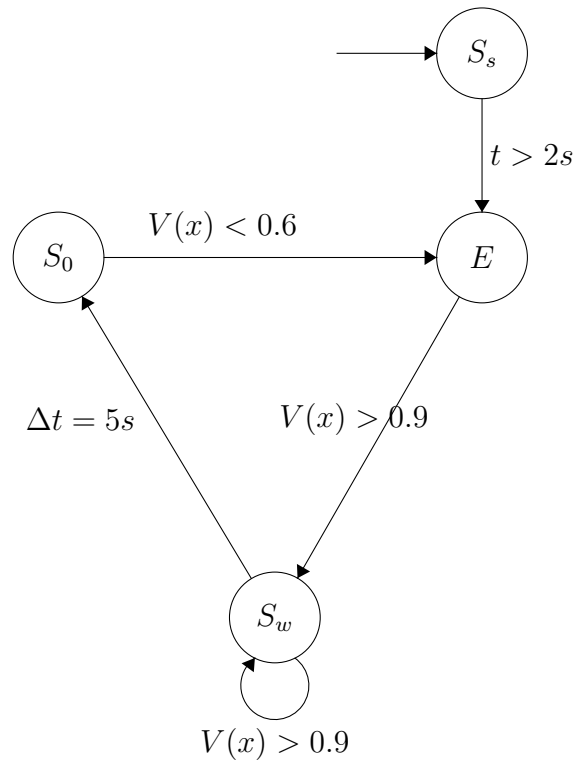


Bild 6.1: Zustandsautomat für die Umschaltung zwischen den Reglern

6.5 Umschaltlogik

Für diese Arbeit wurde eine zu der in [Apf] und [SS99] ähnliche Umschaltlogik implementiert. Diese soll zu jedem Zeitpunkt darüber entscheiden, ob der experimentelle Regler oder der sichere Regler aktiv sein soll. Es werden stets beide Regelgesetze berechnet. Die tatsächlichen Motorsignale werden allerdings nur durch den aktiven Regler vorgegeben.

Der Zustandsautomat in Abbildung 6.1 zeigt das Verhalten der Umschaltlogik. Die Zustände S repräsentieren den sicheren Regler, Zustand E den experimentellen Regler. t bezeichnet die Zeit seit Start des Kopters, Δt bezeichnet die Zeit seit der letzten Zustandstransition des Automaten. Die Implementierung wurde in Simulink mithilfe von Flip-Flops und Zählern realisiert. Beim Start des Kopters wird zwei Sekunden lang zunächst der sichere Regler verwendet. Dies ist nicht dringend notwendig, da die Lyapunov-Funktion während des Startvorgangs erwartungsgemäß ohnehin sehr große Werte annimmt, stellt aber einen zusätzlichen Sicherheitsmechanismus dar. Danach wird auf den experimentellen Regler umgeschaltet. Dieser wird solange betrieben,

bis die Lyapunov-Funktion einen Wert von 0.9 überschreitet. Es wird dann auf den sicheren Regler umgeschaltet, der das System in den stabilen Arbeitspunkt zurückbringt. Solange die Lyapunov-Funktion den Wert von 0.9 unterschreitet, werden fünf Sekunden hochgezählt. Hat nach Ablauf dieser Zeit die Lyapunov-Funktion den Wert 0.6 unterschritten, so wird erneut der experimentelle Regler aktiviert.

6.6 Simulative Erprobung

Ob die ermittelte Zustandsmenge und die Umschaltlogik das gewünschte Verhalten aufweisen, soll zunächst simulativ überprüft werden. Dafür wird die in Abschnitt 3.4 und Abschnitt 5.5 erarbeitete Simulation um die Umschaltung erweitert. Als sicherer Regler soll der in Kapitel 3 entworfene Regler verwendet werden. Der experimentelle Regler soll folgendes Paradigma realisieren:

$$U_{exp} = [\tau_{tot,fix} \quad \tau_x \quad \tau_y \quad \tau_z]^T = [-1.25mg \quad \tau_{x_b} \quad \tau_{x_b} \quad \tau_{x_b}]^T. \quad (6.20)$$

Das heißt, der Regler stabilisiert die Lage entsprechend des sicheren LQR, beschleunigt aber stur mit einem Schub von $\tau_{tot,fix} = -1.25mg$ nach oben. Erwartungsgemäß sollte dieses Verhalten rechtzeitig durch die Sicherheitsüberwachung abgefangen werden, sodass der Zustand z nicht davonläuft. Zunächst wird der Regelkreis ohne den Beobachter betrachtet, d.h. der Zustandsvektor wird direkt zurückgeführt. Bild 6.2 zeigt den Verlauf der z -Koordinate für den simulierten Flug mit den geschilderten Reglern. Bild 6.3 zeigt den zugehörigen Verlauf der Lyapunov-Funktion. Anhand dieses Experimentes kann gezeigt werden, dass die Methode zur Findung der Lyapunov-Funktion ein valider Ansatz ist. Veränderungen der Schranken des zulässigen Zustandsraums schlagen sich erwartungsgemäß auf die Form des beobachteten Zustandsverlaufs nieder, ebenso Änderungen an den LQR-Entwurfsparametern. Die Umschaltlogik zeigt ebenfalls das gewünschte Verhalten. Es lässt sich beobachten, dass alle 5 Sekunden der experimentelle, dysfunktionale Regler für eine Welle im Zustandsverlauf sorgt.

Es fällt beim Experimentieren auf, dass sich bei dem verwendeten Aufbau der Ausschlag der z -Trajektorie ab einer gewissen Lockerung der Zustandsschranken nicht mehr weiter ändert. Dies hängt mit der Stellsignalbegrenzung und der symmetrischen Natur der Lyapunov-Funktion zusammen. Da der Ellipsoid, den die invariante

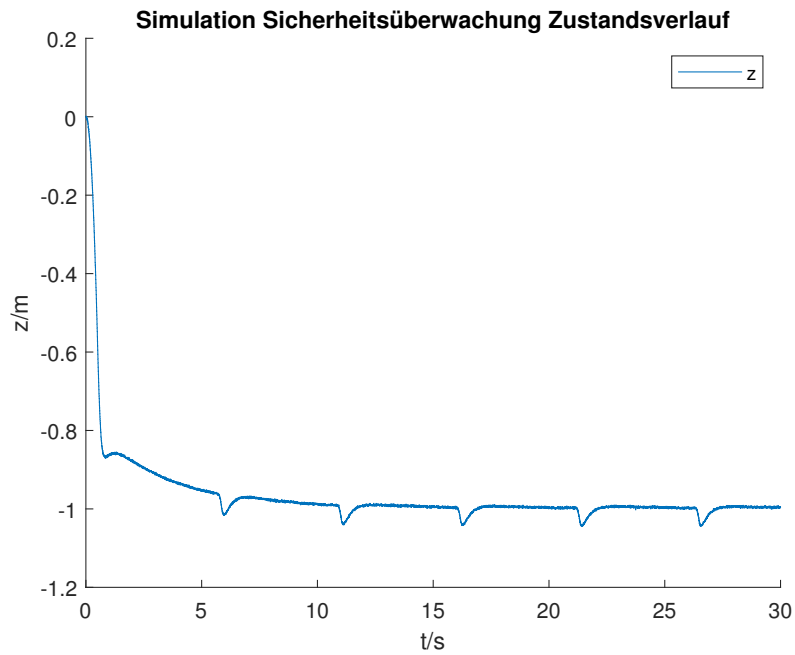


Bild 6.2: Simulation des Zustandsverlauf der z-Koordinate bei Sicherheitsüberwachung ohne Beobachter

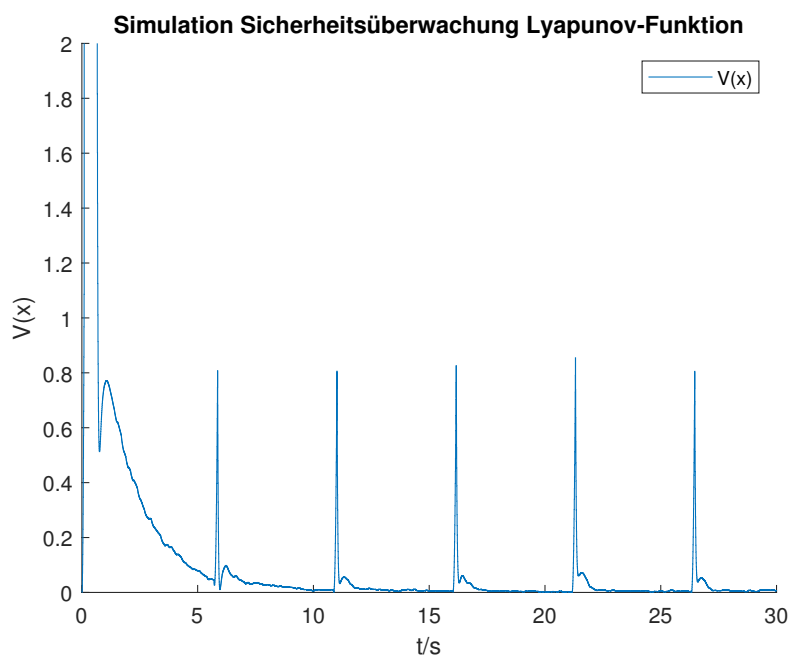


Bild 6.3: Simulation des zeitlichen Verlaufs der Lyapunov-Funktion ohne Beobachter

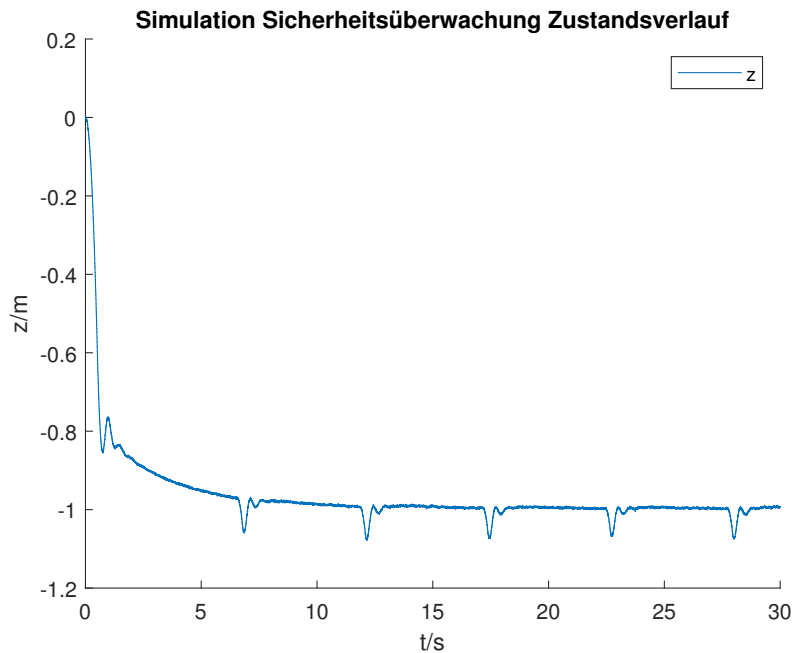


Bild 6.4: Simulation des Zustandsverlauf der z -Koordinate bei Sicherheitsüberwachung mit Beobachter

Zustandsmenge darstellt, symmetrisch zum Ursprung ist, spiegelt sich die Stellsignalbeschränkung des Schubs in negative z -Richtung ebenfalls als maximaler Schub in positive z -Richtung wider. Dadurch kommt es ab einer gewissen Abweichung vom Arbeitspunkt in negative z -Richtung dennoch zur Umschaltung. Dies muss in weiteren Betrachtungen adressiert werden.

Für Bild 6.3 hat sich gezeigt, dass die Lyapunov-Funktion tatsächlich nie den Wert $V(x) = 1$ erreicht. Nun wird der Regelkreis über den in Kapitel 5 entwickelten Beobachter geschlossen. Bild 6.4 zeigt den Trajektorienverlauf der z -Koordinate. Bild 6.5 zeigt den nun veränderten Verlauf der Lyapunov-Funktion. Der Funktionswert wächst über die 1 hinaus, d.h. der Zustand verlässt die wiederherstellbare Zustandsmenge. Dies deutet darauf hin, dass die Vernachlässigung der Beobachterdynamik bei der Berechnung der wiederherstellbaren Zustandsmenge unzulässig ist. Diese Problematik muss in späteren Arbeiten weiter beleuchtet werden.

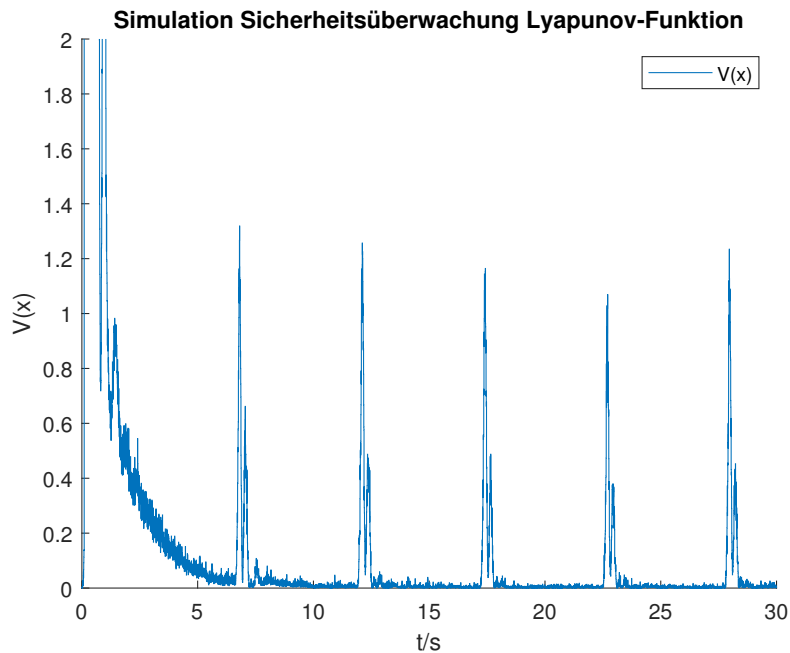


Bild 6.5: Simulation des zeitlichen Verlaufs der Lyapunov-Funktion mit Beobachter

6.7 Erprobung am Parrot Mambo

In dieser Arbeit wurden lediglich vorläufige Versuche durchgeführt, um die Grundfunktionalität der Sicherheitsüberwachung zu überprüfen. Der in Abschnitt 6.6 eingeführte experimentelle Regler soll dafür weiterhin verwendet werden. Eine Videoaufnahme des Flugtests befindet sich in Anhang D.

Es stellt sich heraus, dass der Quadrocopter das erwartete Verhalten aufweist. Die Position wird stabil gehalten, bis der experimentelle Regler am Zug ist. Dieser beschleunigt den Kopter nach oben, bis die Umschaltung ein Verlassen des sicheren Zustandsraums vorhersagt. Der sichere Regler führt das System dann in seinen stabilen Arbeitspunkt zurück. Nach zirka 5 Sekunden wiederholt sich der Vorgang.

Um die Parameter der Sicherheitsumschaltung auszuarbeiten, sind weitere Experimente nötig. So würde man beispielsweise unterschiedliche experimentelle Regler anwenden, die jeweils einen Zustand destabilisieren. Man würde dann iterativ die Schranken für diesen Zustand und dessen Ableitungen anpassen, bis sich das gewünschte Verhalten einstellt. Gegebenfalls müssen parallel die Entwurfsparameter des LQR abgeändert werden.

Kapitel 7

Zusammenfassung und Ausblick

In dieser Arbeit wurden die Grundlagen für eine Quadropterregelung als Demonstratorsystem für Echtzeit-Problemstellungen etabliert. Dafür wurde eine lineare Zustandsraumregelung in Form eines LQR entworfen. Um Integrationsdrift zu unterbinden und den Quadropter tauglicher für den Flug in Innenräumen zu machen, wurde eine Kameraauswertung auf Basis einer Bodenmarkierung ausgelegt und implementiert. Um die daraus gewonnenen Messdaten zu verarbeiten, wurde dann ein Zustandsbeobachter in Form eines Kalman-Filters entworfen. Abschließend wurde eine Sicherheitsüberwachung implementiert, die auf Basis einer invarianten Zustandsmenge intelligent zwischen einem sicheren und einem experimentellem Regler hin- und herschalten soll.

Die lineare Zustandsregelung durch den LQR aus Kapitel 3 war in der Lage, das Quadroptersystem zuverlässig zu stabilisieren und Sollgrößensprüngen zu folgen. Stationäre Genauigkeit konnte mithilfe der Integralzustände gewährleistet werden. Es hat sich ebenfalls gezeigt, dass ein Betrieb mit frei wählbarem Gierwinkel möglich ist.

Der entworfene Zustandsbeobachter auf Basis der Kameraauswertung aus Kapitel 4 und Kapitel 5 konnte die Bodenmarkierung zuverlässig erkennen und den Quadropter in die entsprechende Position überführen. Der vorliegende Integrationsdrift konnte hierdurch eliminiert werden. Der hierfür ausgelegte Kalman-Filter war auch für die anderen Systemzustände in der Lage, eine Zustandsschätzung zu liefern. Ein Betrieb des Quadropters bei beliebigem Gierwinkel war ebenfalls möglich.

Für die Sicherheitsüberwachung wurden in Kapitel 6 grundlegende theoretische Konzepte erarbeitet und erste simulative und physikalische Experimente durchgeführt. Es hat sich herausgestellt, dass die Ideen aus [Apf] und [SS99] tauglich sind, um den Quadrocopterflug zu überwachen. Die Sicherheitsüberwachung muss noch in manchen Aspekten weiterentwickelt werden. Es wurden einige Punkte angesprochen, die zur Weiterentwicklung der besprochenen Konzepte in Betracht gezogen werden müssen. Dazu gehören die Berücksichtigung der Beobachterdynamik, die genauere Analyse des Einflusses von Stellsignalbegrenzungen sowie ein anschließendes experimentelles Anpassen der Entwurfparameter.

Für einen möglichst verschleißfreien Betrieb des Quadrocopters wäre es zudem von Interesse, einen sicheren Landemechanismus zu implementieren. Ohne einen solchen Landemechanismus kommt es nach jedem Flug zu einem Sturz aus ca. 1m Flughöhe. Dies führt auf Dauer zu Verschleiß des Versuchsaufbaus. Ebenso wäre es denkbar, den Startvorgang weiter zu verbessern, beispielsweise durch eine Trajektorienplanung.

Literaturverzeichnis

- [Apf] APFEL, Quirin: *Sichere Regelung eines Quadropters*, Lehrstuhl für Regelungstechnik, FAU Erlangen-Nürnberg, Diplomarbeit
- [BCS18] BOF, Nicoletta ; CARLI, Ruggero ; SCHENATO, Luca: *Lyapunov Theory for Discrete Time Systems*. (2018)
- [BEGFB94] BOYD, Stephen ; EL GHAOU, Laurent ; FERON, Eric ; BALAKRISHNAN, Venkataramanan.: *Linear Matrix Inequalities in System and Control Theory*. SIAM, 1994
- [BNS04] BOUABDALLAH, Samir ; NORTH, André ; SIEGWART, Roland: *PID vs LQ Control Techniques Applied to an Indoor Micro Quadrotor*, 2004
- [Bri71] BRITTING, Kenneth R.: *Inertial navigation systems analysis*. New York : Wiley-Interscience, 1971
- [Cra] CRAIG, John J.: *Introduction to Robotics*. Pearson Prentice Hall
- [FKL⁺13] FÖLLINGER, Otto ; KONIGORSKI, Ulrich ; LOHMANN, Boris ; ROPPEN-ECKER, Günter ; TRÄCHTLER, Ansgar: *Regelungstechnik*. 11. Auflage. VDE-Verlag, 2013
- [Gre] GREIFF, Markus: *Modelling and Control of the Crazyflie Quadrotor for Aggressive and Autonomous Flight by Optical Flow Driven State Estimation*, Department of Automatic Control, University of Lund, Diplomarbeit
- [HS92] HARALICK, Robert M. ; SHAPIRO, Linda G.: *Computer and Robot Vision*. Addison-Wesley Longman Publishing Co., Inc., 1992

- [KFG⁺16] KLAUS, Tobias ; FRANZMANN, Florian ; GAUKLER, Maximilian ; MICHALKA, Andreas ; ULBRICH, Peter: Closing the Loop: Towards Control-aware Design of Adaptive Real-Time Systems. In: IEEE (Hrsg.): *Proceedings of the 37th Real-Time Systems Symposium Work-in-Progress Session (RTSS WiP '16)*, 2016, 1–4
- [Lan] LANDRY, Benoit: *Planning and Control for Quadrotor Flight through Cluttered Environments*, Massachusetts Institute of Technology, Diplomarbeit
- [Liu00] LIU, Jane W. S.: *Real-Time Systems*. Pearson, 2000
- [Lun16] LUNZE, Jan: *Regelungstechnik 2: Mehrgrößensysteme, Digitale Regelung*. 9. Berlin : Springer Vieweg, 2016 (Lehrbuch)
- [Mat] *Simulink Quadcopter Project Documentation*. <https://de.mathworks.com/help/aeroblks/quadcopter-project.html>, . – Accessed: 2020-02-02
- [May] MAYBECK, Peter S.: *Stochastic Models, Estimation, and Control, Vol.1*. Academic Press, Inc.
- [PKYK17] PHONG, Ha N. ; KI, Wan K. ; YOUNG, Won L. ; KANG, Ryoung P.: Remote Marker-Based Tracking for UAV Landing Using Visible-Light Camera Sensor. (2017)
- [SS99] SETO, Danbing ; SHA, Lui: An Engineering Method for Safety Region Development. (1999)
- [WB06] WELCH, Greg ; BISHOP, Gary: An Introduction to the Kalman Filter. (2006)
- [Yan] YANG, Runmu: *Modellierung und Regelung eines Quadropters*, Lehrstuhl für Regelungstechnik, FAU Erlangen-Nürnberg, Diplomarbeit

Anhang A

Verzeichnis der Formelzeichen

A	Dynamikmatrix
B	Eingangsmatrix
C	Ausgangsmatrix
K	Reglerverstärkung
L	Beobachterverstärkung
X	Zustandsvektor
U	Eingangsvektor
Y	Ausgangsvektor
x	kartesische Koordinate x (nach vorne)
y	kartesische Koordinate y (nach rechts)
z	kartesische Koordinate z (nach unten)
ϕ	Rollwinkel
θ	Nickwinkel
ψ	Gierwinkel
p	Winkelgeschwindigkeit um y -Achse im Weltkoordinatensystem
q	Winkelgeschwindigkeit um x -Achse im Weltkoordinatensystem
r	Winkelgeschwindigkeit um z -Achse im Weltkoordinatensystem
g	Beschleunigung durch Erdgravitationsfeld
m	Masse des Quadropters
J	Matrix der Massenträgheitsmomente des Quadropters
S	Vektor der Integralzustände
T_s	Abtastperiode
Q_R	Kostenmatrix für Zustandsgrößen beim LQR
R_R	Kostenmatrix für Eingangsgrößen beim LQR
P	Berechnungsmatrix für Riccati-Gleichung / Lyapunov-Funktion
E	Zustandsfehler
T	Transformationsmatrix
G	RGB-Aufnahme der Bauchkamera in Tensorform

f	Farbpunkt im RGB-Raum, welches der Farbe der Bodenmarkierung entspricht
d	Toleranzvektor der Farberkennung im RGB-Raum
μ	Mittelwert einer Messgröße
σ	Standardabweichung einer Messgröße
Q_B	Kovarianzmatrix für Systemrauschen beim Kalman-Filter
R_B	Kovarianzmatrix für Messrauschen beim Kalman-Filter
\hat{X}	geschätzter Zustand
$V(x)$	Lyapunov-Funktion des Zustandsfehlers
a	Schranke für zulässige Zustandsmenge
Q	inverse Matrix von P für Berechnung der Lyapunov-Funktion
\mathcal{X}	zulässige Zustandsmenge
\mathcal{W}	wiederherstellbare Zustandsmenge

Anhang B

Simulationsmodell

Dieses Simulink-Modell enthält alle erarbeiteten Systemkomponenten und eine Simulation des linearisierten Quadroptersystems.

Folgende MATLAB-Skripte sind enthalten:

- ModelValues: Initialisiert allgemeine Werte für das Modell
- LQRValues: Initialisiert Werte für den Linear Quadratic Regulator
- KalmanFilterValues: Initialisiert Werte für den Beobachter

Folgende Simulink-Modelle sind enthalten:

- flightController: Flugregelung mit LQR und Sicherheitsüberwachung
- MAINSimulationModel: Modell für die Simulation des Gesamtsystems mit Strecke und Beobachter

Eine grobe Darstellung des Modells zeigt Bild B.1. K ist dabei der in Kapitel 3 entworfene Regler mit Sicherheitsumschaltung, S eine lineare Simulation des in Kapitel 2 entwickelten Systemmodells mit Stellsignalbegrenzung und Systemrauschen sowie Modellunsicherheit und L der in Kapitel 5 ausgelegte Beobachter.

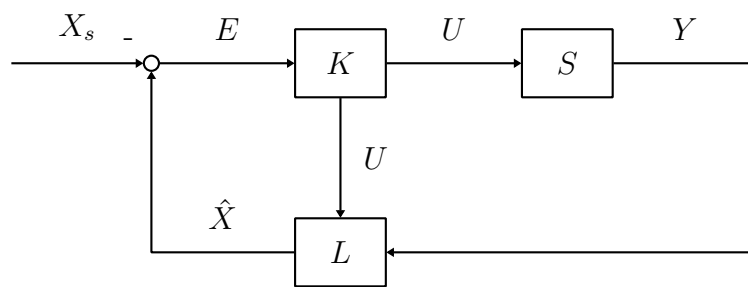


Bild B.1: Blockschaltbild für das Simulationsmodell

Anhang C

Versuchsaufbau für den Flugtest mit dem Parrot Mambo

Im Folgenden wird einmalig der allgemeine Versuchsaufbau für die Erprobung jeder einzelnen Komponente in dieser Arbeit geschildert.

Etablieren der Schnittstelle zum Parrot Mambo

Praktischerweise stellt Simulink eine Support Package zur Verfügung, die es erlaubt, relativ unkompliziert eine eigene Flugregelung auf den Parrot Mambo zu laden. Hierfür muss pro Parrot Mambo und PC ein Netzwerktreiber installiert werden, der eine Übertragung mittels TCP/IP-Protokoll über Bluetooth ermöglicht. Dafür gibt es einen Installationsassistenten von MATLAB, der sich im folgenden Menü findet: *MATLAB Home* → *AddOns* → *Manage AddOns* → *Simulink Support Package for Parrot Minidrones* → *Zahnrad-Icon*.

Nach der Installation kann die Verbindung zu einem Parrot mit wenigen Mausklicks hergestellt werden. Zuerst muss im Windows-Menü für Bluetooth der gewünschte Kopter mit dem PC gekoppelt werden. Dann kann die TCP/IP-Verbindung im Menü *Systemsteuerung* → *Geräte und Drucker* wie in Bild C.1 hergestellt werden:

Mittels Simulink kann dann ein Modell für den Flugregler ausgelegt werden, welches dann unter Verwendung der *Simulink Coder* Package in C-Code umgewandelt wird. Dieser Code wird dann auf den Quadrocopter geladen und ersetzt dort den bestehenden Flugregler. Anhang Abschnitt C.1 enthält das finale Simulink-Modell für den Kopterflug.

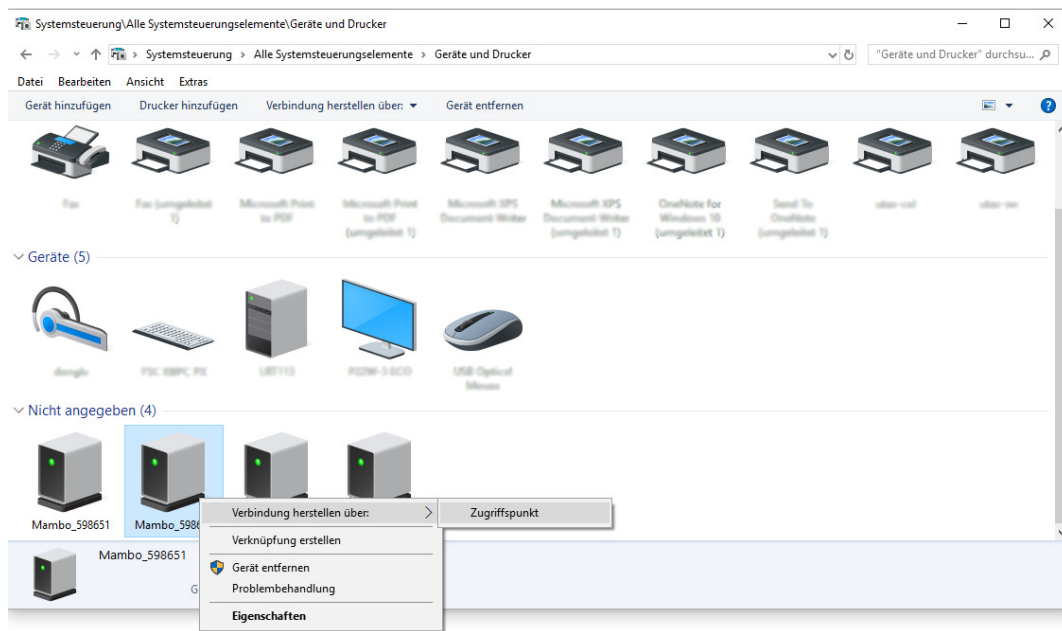


Bild C.1: Verbindungsaufbau zum Parrot Mambo

Vorbereitung der Versuchsumgebung

Für den Kopterflug muss die Testumgebung gewissen Anforderungen gerecht werden. Die Tests in dieser Arbeit wurden in einem Innenraum ausgeführt. Bild C.2 zeigt den Versuchsaufbau.

In der Mitte des Aufbaus befindet sich die Bodenmarkierung, die durch die Gewichtskraft von zwei Metallstäben fixiert wurde. Der umliegende Bereich wurde mit Kügelchen aus Klebmasse versehen. Diese dienen dazu, die optische Flussmessung zu unterstützen. Eine intensive Belichtung des Bodens ist ebenfalls hilfreich, um zu gewährleisten, dass die Kameraaufnahme allgemein brauchbare Werte liefert. Dies gilt insbesondere wegen der dunklen Farbe des Bodens. Fällt die Funktion des optischen Flusses aus, so kann es vor allem während des Startvorgangs schnell passieren, dass der Kopter seine Lineargeschwindigkeit falsch einschätzt und gegen die Wand fliegt. Dies führt schnell zu Schäden an Rotoren und Fluggestell.

Ein optimaler Versuchsaufbau sollte in alle Himmelsrichtungen ca. zwei Meter freies Feld haben. Nahe Gegenstände oder Wände können durch Reflexion der Ultraschallwellen die Höhenschätzung stören. Dadurch kann es passieren, dass der Kopter gegen die Decke fliegt und beschädigt wird. Harte Kanten in der Umgebung sollten ebenso



Bild C.2: Versuchsaufbau

möglichst vermieden werden, da ein Sturz auf eine solche Kante die Sensorik am Bauch des Kopters beschädigen kann.

Es gilt außerdem zu beachten, dass rapide Druckschwankungen durch bspw. das Öffnen einer Tür oder eines Fensters ebenfalls die Höhenmessung temporär stören können.

Vorbereitung des Parrot Mambo für ein Flugexperiment

Bevor ein Flugexperiment durchgeführt werden kann, gilt es, einige Problemquellen im Vorraus zu eliminieren. Zunächst muss überprüft werden, ob der Akku zureichend geladen ist. Dies ist essentiell, da der Kopter andernfalls einen Vorwärtssalto macht und dann ausschaltet. Dies kann die Rotoren beschädigen, daher empfiehlt es sich, zunächst einen Flug mit gedrosselter Motorleistung auszuführen, um zu testen ob der Akku zureichend geladen ist. Um den Akkustand zu überprüfen kann außerdem die *Flight Log*-Funktionalität der Simulink-Schnittstelle zu Rate gezogen werden. Es empfiehlt sich grundsätzlich, einen Zusatzakku parat zu haben, der parallel zu den Flugtests geladen werden kann, da der Kopterakku relativ schnell zuneige geht. So können Totzeiten beim Experimentieren vermieden werden. Es hat sich außerdem

herausgestellt, dass sich bei einem niedrigem Akkustand das Reglertiming ändert und die Lagestabilisierung an Qualität verliert.

Nach einer Bruchlandung muss dringend überprüft werden, ob die Rotoren des Kopters alle weiterhin reibungsfrei drehen können. Ansonsten muss der entsprechende Rotor manuell von seiner Welle entfernt und dann wieder aufgebracht werden.

Für jedes einzelne Flugexperiment muss die Flugregelung komplett neu geladen werden. Wird dies nicht befolgt, so kann die Lage des Kopters ab dem zweiten Flug nicht mehr richtig stabilisiert werden. Der genaue Grund hierfür ist unklar geblieben. Womöglich hängt dieses Phänomen mit zwischengespeicherten Werten aus der Inertial Measurement Unit (IMU) zusammen, die zwischen den Flügen nicht zurückgesetzt werden.

C.1 Modell für den Kopterflug

Das Modell enthält folgende Komponenten für den experimentellen Kopterflug, welche in dieser Arbeit entwickelt wurden:

- `flightController`: Flugregler
- `stateEstimator`: Zustandsbeobachter
- `VisualDataHandling`: Verarbeitung der Kameradaten
- Weitere Komponenten, die bereits in Anhang B beschrieben wurden.

Das Modell enthält außerdem weitere Komponenten, die aus dem *Parrot-Mambo* Beispielmmodell *asbQuadcopter* übernommen wurden.

Anhang D

Videos Flugexperimente

Dateiname	Beschreibung
01LQRTrajectVid	LQR Test
02DriftVid	Nachweis Integrationsdrift
03estimDemoVid	Test Beobachter/Kameraauswertung
04safetyCtrlPrelim	Test Sicherheitsüberwachung

Tabelle D.1: Videoaufzeichnungen des Kopterflugs

Anhang E

Kameraauswertung

E.1 Offline-Farbanalyse

Dieser Anhang enthält das MATLAB-Skript `analyzePixelData`. Dieses Skript dient dazu, die Farbdaten einer Bodenmarkierung zu analysieren.

E.2 Kamerasimulation zur Erprobung des Auswertealgorithmus

Dieser Anhang enthält folgende Dateien zur Simulation und Erprobung des Kameraauswertealgorithmus:

- `pixelPlotFromXyzrpy`: Dieses Skript erlaubt es, mittels Schieberegler für die Zustandsgrößen ein Kamerabild zu simulieren. Dieses kann dann verwendet werden, um den Auswertealgorithmus simulativ zu erproben.
- `LRTBoden`: Eine Aufnahme des Fußbodens, um dem simulierten Bild Hintergrundrauschen hinzuzufügen.

E.3 Algorithmus zur Erkennung der Bodenmarkierung

Dieser Anhang enthält folgende MATLAB-Funktionen, die beim Quadropterflug zum Einsatz kommen, um die x - y -Position sowie ψ zu messen:

- `detectColor`: Diese Funktion nimmt als Eingang eine RGB-Aufnahme sowie Farbparameter und gibt eine Binärmaske derselben Größe zurück, die für jeden Pixel aussagt, ob die gesetzte Farbbedingung erfüllt ist.
- `XYPsiFromVis`: Diese Funktion nimmt als Eingang eine RGB-Aufnahme, einen Schätzwert für die Flughöhe sowie einen Schätzwert für ψ und gibt eine Messung für x , y und ψ sowie eine Validitätsbeurteilung für die jeweiligen Messwerte zurück.