

Universität Bielefeld
Arbeitsgruppe Kognitronik & Sensorik
Studiengang BioMechatronik

Dokumentation

Masterprojekt - AMiRoFly

Thema:	Zustandsvorhersage und Positionskontrolle einer Quadrotordrohne auf Hardwarebasis des AMiRo
Autor:	Philipp Jünemann pjuenemann@techfak.uni-bielefeld.de
Version vom:	7. November 2017
Betreuer:	Timo Korthals, M.Sc.

Inhaltsverzeichnis

1	Einleitung	3
2	Grundlagen	4
2.1	Flight Controller	4
2.2	AMiRo	4
2.3	Telewerkbank	5
2.4	Kommunikation der Systeme	5
3	Modellbeschreibung	6
3.1	Extended Kalman Filter	6
3.1.1	Modell der Zustandsschätzung	7
3.1.2	Sensormodell	8
3.1.3	Ermittlung der Kovarianz des Messrauschens	9
3.2	Positionsregelung	10
4	Modellierung mit MatLab/Simulink	10
4.1	Extended Kalman-Filter in Simulink	10
5	Zusammenfassung	11
	Abbildungsverzeichnis	12
	Tabellenverzeichnis	12
	Literaturverzeichnis	13
	Anhang	14

1 Einleitung

Der Bedarf nach autonom agierenden Systemen steigt kontinuierlich an. Dadurch werden Forschung und Industrie angetrieben innovative, ressourceneffiziente Systeme zu entwickeln, die sich durch autonomes Handeln in der Umwelt auszeichnen.

Der AMiRo ist solch ein System, das in der Arbeitsgruppe Kognitronik & Sensorik der Universität Bielefeld entwickelt wird. Dabei handelt es sich um einen zylinderförmigen Miniroboter mit zwei Rädern. Mithilfe von acht Infrarot-Sensoren und anderen Modulen, wie Kamera oder Laserscanner, die wahlweise angebracht werden können, kann er sich autonom bewegen und orientieren[?].

In dem Simulationsprogramm Gazebo gibt es ebenfalls eine Nachbildung des AMiRo. Damit können jegliche Applikationen in einer realitätsnahen Umgebung auf dem AMiRo simuliert werden. Da bislang noch kein Modell mit Infrarot-Sensoren in Gazebo vorhanden ist, soll innerhalb dieses Projekts solch ein AMiRo entwickelt werden. Als Grundlage dafür dient ein Sensor vom Typ VCNL4020, der bereits im herkömmlichen Roboter verbaut ist. Das Modell in Gazebo soll ebenso mit acht dieser Sensoren erweitert werden.

Diese Dokumentation beschreibt das Vorgehen bei der Entwicklung des Simulationsmodells in vier Schritten. Zuerst wird die Aufnahme der originalen Sensordaten und deren Auswertung beschrieben. Anschließend wird der Entwurf und die Implementierung eines Näherungssensors in Gazebo sowie die des dazugehörigen Plugins erläutert. In Kapitel 4 wird das Sensormodell, das im Hintergrund realitätsnahe Sensorwerte im 16 Bit-Format erzeugt, entwickelt und beschrieben. Die Evaluation des Modells wird in Kapitel 5 mithilfe eines Mapping-Programms durchgeführt und ausgewertet. Abschließend werden die erzielten Ergebnisse zusammengefasst.

2 Grundlagen

Dieses Kapitel enthält die Beschreibungen der in diesem Projekt verwendeten Systeme. Darunter fallen der AMiRo, der die Hardware zur Berechnung der Zustandsschätzung sowie der Positions- und Lageregelung bereitstellt. Ebenso gehören zu den Systemen die Hardware der Drohne mitsamt des Flight Controllers, der geregelt werden soll sowie die Telewerkbank, innerhalb der die Position der Drohne erfasst wird.

2.1 Flight Controller

Als zentrale Steuerungshardware wurde für die AMiRoFly-Drohne der Seriously Pro Racing F3 EVO (SPRacingF3EVO) ausgewählt. Dieser Flight Controller stammt aus dem mittleren Kostensegment und besitzt eine mit 72 MHz getaktete CPU¹ vom Typ STM32F303. Daneben sind darauf sämtliche Komponenten verbaut, die dem aktuellen Standard entsprechen. Darunter fallen ein 9-Achsen erfassender Bewegungssensor vom Typ InvenSense MPU-9250. Dieser besteht aus einer IMU² mit einem 3-Achsen Beschleunigungssensor (Accelerometer) und einem 3-Achsen Drehratensensor (Gyroskop). Ersterer besitzt in dem verwendeten Modus einen Wertebereich von $\pm 16\text{ g}$ und ist mit einem Faktor von 2048 LSB/g skaliert. Das Gyroskop ist mit einem Wertebereich von $\pm 2000 \frac{\circ}{\text{s}}$ und einem Skalierungsfaktor von $16,4 \frac{\text{LSB}}{\frac{\circ}{\text{s}}}$ spezifiziert. Außerdem ist auf der MPU-9250 ein 3-Achsen Digitalkompass (Magnetometer) verbaut, dessen Wertebereich von $\pm 4800 \mu\text{T}$ reicht und mit $0,6 \frac{\mu\text{T}}{\text{LSB}}$ skaliert ist. Sämtliche Skalierungsfaktoren werden in dem später beschriebenen Modell berücksichtigt. Die Kommunikation mit der CPU wird über eine SPI³-Schnittstelle vollzogen. Weiterhin besitzt der SPRacingF3EVO ein Barometer vom Typ BMP280 sowie sämtliche weitere Schnittstellen.

2.2 AMiRo

Der AMiRo ist ein Robotersystem, das im Rahmen von Forschungsprojekten innerhalb der Arbeitsgruppe Kognitronik & Sensorik entwickelt wurde. Dabei handelt es sich um einen zylinderförmigen Miniroboter, dessen Komponenten sich bedingt durch seine modulare Bauweise verschieden variieren lassen. Dadurch setzt sich der AMiRo aus verschiedenen Platinen zusammen, die jeweils einen bestimmten Zweck erfüllen.

Für dieses Projekt soll der AMiRo als Steuerungshardware für eine Drohne dienen. Dafür wurde ein AMiRo bestehend aus einem *PowerBoard*, einem *CognitionBoard* und einem *LightRingBoard*. Das *PowerBoard* ist dabei für die Stromversorgung und das gesamte elektronische Verhalten des Systems zuständig (siehe AMiRo-Paper). Dafür

¹Central Processing Unit

²Inertial Measurement Unit

³Serial Peripheral Interface

ist es mit einem STM32F405 Mikrocontroller ausgestattet, der einen ARM Cortex-M4 als CPU besitzt. Das *CognitionBoard* hingegen besitzt einen ARM Cortex-A8, der innerhalb des Projekts, die Zustandsschätzung und Positions- sowie Lageregelung des Systems durchführt. Dafür arbeitet auf dem AMiRo ein Linux-System. Abschließend besitzt die verwendete Variante des AMiRo ein *LightRingBoard*, dessen LEDs⁴ zu Debugging-Zwecken verwendet werden können. Die Kommunikation mit peripheren Systemen wird hier über eine serielle sowie eine Wi-Fi-Schnittstelle betrieben.

2.3 Telewerkbank

Die Telewerkbank(TWB) ist ein für Kamera-Tracking ausgelegtes Labor der Arbeitsgruppe Kognitronik & Sensorik. Mit Hilfe von vier an der Decke platzierten Kameras, die vertikal zum Boden gerichtet sind, kann eine Fläche mit einer Größe von etwa 6m mal 6m erfasst werden. Für die TWB spezifizierte Marker können so auf dieser Fläche erkannt und ihre Position getrackt werden. Die Kommunikation innerhalb der TWB geschieht dabei über RSB⁵-Ressourcen. Um die TWB auch für das Tracking von Drohnen auszulegen, wurde das Kamerasystem modifiziert, sodass es nun möglich ist, Marker in einer Höhe von bis zu 1,5m zu erfassen. Das Tracking ist somit auch für den dreidimensionalen Raum ausgelegt.

2.4 Kommunikation der Systeme

Damit die in diesem Projekt erarbeitete Zustandsschätzung sowie Positions- und Lageregelung der Drohne durchgeführt werden kann, muss die Kommunikation der drei zuvor beschriebenen Systeme untereinander gewährleistet sein. Der AMiRo fungiert dabei als Mittelpunkt des Systems, da auf ihm die Zustandsschätzung und anschließende Regelung berechnet werden.

Wie in Kapitel Einleitung beschrieben, stellt die TWB dafür die getrackte Position der Drohne zur Verfügung. Dies geschieht über ein Wi-Fi-Netzwerk, über das der AMiRo mit der Telewerkbank kommuniziert. Über das RSB können die Daten anschließend auf dem AMiRo zur Berechnung verwendet werden.

Die bidirektionale Kommunikation zwischen dem AMiRo und dem SPRacingF3EVO, der als Hardwarebasis der Drohne fungiert, wird über eine serielle Schnittstelle hergestellt. Der AMiRo liest dabei die Sensordaten, die vom Flight Controller ausgegeben werden. In diesem Fall sind das Beschleunigungswerte, Drehraten, Magnetfeld- und Luftdruckwerte. Entgegengesetzt schreibt der AMiRo Steueranweisungen an den SPRacingF3EVO.

⁴Light-Emitting Diode

⁵Robotics Service Bus

Das verwendete Protokoll wurde in einem verwandten Projekt entwickelt und basiert auf dem Base85-Kodiervorgahren. Für einen Lesezugriff auf den Flight Controller Binärdaten dekodiert und für einen Schreibzugriff entsprechend kodiert.

3 Modellbeschreibung

In diesem Kapitel wird das Modell für die Zustandsschätzung und Positionskontrolle einer Drohne im dreidimensionalen Raum nach Santana et al. [1] beschrieben. Es handelt sich dabei um ein Modell, das mit Hilfe eines Extended Kalman Filters (EKF) eine Schätzung für die inneren Zustände des Drohnensystems vornimmt und diese mit Messdaten der Telewerkbank und des Flight Controllers korrigiert. Anschließend können die Steuersignale der Drohne so geregelt, dass diese in ihrer Position kontrolliert werden kann. Dies geschieht mittels vier PD-Reglern, die jeweils an ein Steuersignal gekoppelt sind. Das gesamte Modell ist in der Mathworks Entwicklungsumgebung Simulink beschrieben. Zudem wurde MatLab verwendet, um den EKF zu parametrisieren.

3.1 Extended Kalman Filter

Der Extended Kalman-Filter ist eine Erweiterung für nichtlineare Modelle, die aus dem ursprünglichen Kalman-Filter hervorgeht. Dieser ist ein mathematisches Modell von Rudolf E. Kalman, das zur Zustandsschätzung beziehungsweise -vorhersage dient (Zitat Daniel Rudolph). Hinter dem Filter verbirgt sich ein Algorithmus, der mit Hilfe eines Prädiktor-Schritts eine Zustandsschätzung für den nächsten Zeitschritt ermittelt. In dem darauffolgenden Korrektor-Schritt wird diese Schätzung mit aktuellen Sensordaten korrigiert und verbessert. Die sequentielle Abfolge des Prädiktor- und Korrektor-Schritts mitsamt der dazugehörigen Formeln ist in Abb 4.3 dargestellt.

Abbildung 4.3

Da es sich bei Santana et al. um ein nichtlineares Modell handelt, wird der Extended Kalman-Filter verwendet. Dieser basiert auf den Zustandsgleichung x_k für die Zustandsschätzung und z_k für den Messterm (Daniel).

$$x_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + w_{k-1} \quad (1)$$

$$z_k = h(\mathbf{x}_k) + v_k \quad (2)$$

Dabei repräsentieren f und h differenzierbare, nichtlineare Funktionen für den Zustandsübergang sowie das Sensormodell zu einem diskreten Zeitpunkt k . w und v beschreiben das gaußsche System- beziehungsweise Messrauschen.

3.1.1 Modell der Zustandsschätzung

Als Zustandsvektor des Systems ist hier

$$x = [x \ y \ z \ \dot{x} \ \dot{y} \ \dot{z} \ v_x \ v_y \ \phi \ \theta \ \psi \ \dot{\psi}], \quad (3)$$

gewählt, wobei $x, y, z, \dot{x}, \dot{y}$ und \dot{z} die linearen Positionen (in m) und Geschwindigkeiten (in $\frac{\text{m}}{\text{s}}$) im dreidimensionalen Referenzkoordinatensystem sind. v_x und v_y repräsentieren die linearen Geschwindigkeiten der Drohne (in $\frac{\text{m}}{\text{s}}$) in ihrem eigenen Referenzsystem. ϕ, θ, ψ und $\dot{\psi}$ sind die Rotationen (in $^\circ$) um die Achsen des Systems beziehungsweise die Rotationsgeschwindigkeit um die z -Achse (in $\frac{^\circ}{\text{s}}$).

Als Gleichung für den Zustandsübergang ist

$$\begin{pmatrix} x_{k+1} \\ y_{k+1} \\ z_{k+1} \\ \dot{x}_{k+1} \\ \dot{y}_{k+1} \\ \dot{z}_{k+1} \\ v_{x_{k+1}} \\ v_{y_{k+1}} \\ \phi_{k+1} \\ \theta_{k+1} \\ \psi_{k+1} \\ \dot{\psi}_{k+1} \end{pmatrix} = \begin{pmatrix} x_k + \delta t \cdot \dot{x}_k \\ y_k + \delta t \cdot \dot{y}_k \\ z_k + \delta t \cdot \dot{z}_k \\ c_{\psi_k} \cdot v_{x_k} - s_{\psi_k} \cdot v_{y_k} \\ s_{\psi_k} \cdot v_{x_k} + c_{\psi_k} \cdot v_{y_k} \\ \dot{z} + \delta t \cdot \ddot{z}_k(\mathbf{x}, \mathbf{u}) \\ v_{x_k} + \delta t \cdot \dot{v}_{x_k}(\mathbf{x}) \\ v_{y_k} + \delta t \cdot \dot{v}_{y_k}(\mathbf{x}) \\ \phi_k + \delta t \cdot \dot{\phi}_k(\mathbf{x}, \mathbf{u}) \\ \theta_k + \delta t \cdot \dot{\theta}_k(\mathbf{x}, \mathbf{u}) \\ \psi_k + \delta t \cdot \dot{\psi}_k \\ \dot{\psi}_k + \delta t \cdot \ddot{\psi}_k(\mathbf{x}, \mathbf{u}) \end{pmatrix}, \quad (4)$$

wobei c_ψ für $\cos(\psi)$ und s_ψ für $\sin(\psi)$ sowie δt für die Dauer eines diskreten Zeitschritts (in s) stehen. Des Weiteren werden \dot{v}_{x_k} und \dot{v}_{y_k} mit Hilfe der Funktionen

$$\dot{v}_x = K_1(s_\psi s_\phi c_\theta + c_\psi s_\theta) - K_2 v_x \quad (5)$$

$$\dot{v}_y = K_3(-c_\psi s_\phi c_\theta + s_\psi s_\theta) - K_4 v_y, \quad (6)$$

3.1.3 Ermittlung der Kovarianz des Messrauschens

Neben der Kovarianz des Prozessrauschens, die hier experimentell für das Verhalten des Filters angepasst werden kann, muss die Kovarianz des Messrauschens eingebracht werden. Sowohl für die Telewerkbank als auch für die Sensoren auf dem SPRacingF3EVO mussten diese zuvor ermittelt werden. Für Erstere wurde die Fläche unter einer Kamera der TWB vermessen. Dies geschah, indem die Position zweier unterschiedlicher Marker an 6 x 6 x 6 Stellen im dreidimensionalen Raum getrackt wurden.

Anschließend wurde die minimale Anzahl an berechneten Markerpositionen je Stelle ermittelt und insgesamt 216 Punktwolken mit dieser Anzahl an Datensätzen gebildet. Zu jedem Datenpunkt wurde daraufhin die mittlere Abweichung innerhalb seiner Punktwolke bestimmt. Mit Hilfe der MatLab-Funktion `cov()` wurde die Kovarianzmatrix über die mittleren Abweichungen aller Datenpunkte berechnet. Für den 99 mm mal 99 mm großen Marker, der sich innerhalb der Untersuchungen als der geeignetste erwies, ergab sich folgende Kovarianzmatrix:

$$Q_{TWB} = \begin{bmatrix} 1,1404 \cdot 10^{-4} & -2,8209 \cdot 10^{-6} & 1,4042 \cdot 10^{-5} \\ -2,8209 \cdot 10^{-6} & 1,1566 \cdot 10^{-4} & 1,0895 \cdot 10^{-6} \\ 1,4042 \cdot 10^{-5} & 1,0895 \cdot 10^{-6} & 0,0014 \end{bmatrix} \quad (15)$$

Um die zweite geforderte Kovarianzmatrix, die das Messrauschen der Sensoren auf dem Flight Controller beschreibt, zu erhalten, wurde ein Datensatz mit Sensormesswerten des Beschleunigungs- und Drehratensensors sowie des Kompass und des Barometers aufgenommen. Ebenso wurde mit der MatLab-Funktion `cov()` die Kovarianzmatrix berechnet.

$$Q_{Drone} = \begin{bmatrix} 0,1285 & -0,0034 & 0,0008 & -1,3570 \cdot 10^{-5} & 7,9558 \cdot 10^{-6} & 6, \\ -0,0034 & 0,1086 & -0,0011 & 1,7921 \cdot 10^{-5} & -3,9995 \cdot 10^{-6} & -1, \\ 0,0008 & -0,0011 & 0,05915 & 5,1793 \cdot 10^{-6} & -1,4508 \cdot 10^{-5} & 6, \\ -1,3570 \cdot 10^{-5} & 1,7921 \cdot 10^{-5} & 5,1793 \cdot 10^{-6} & 7,4049 \cdot 10^{-7} & 4,0087 \cdot 10^{-8} & -1, \\ 7,9558 \cdot 10^{-6} & -3,9995 \cdot 10^{-6} & -1,4508 \cdot 10^{-5} & 4,0087 \cdot 10^{-8} & 1,1241 \cdot 10^{-6} & 7, \\ 0,0001 & -0,0001 & 6,1779 \cdot 10^{-6} & -1,3290 \cdot 10^{-7} & 7,1049 \cdot 10^{-8} & 3, \\ -5,9173 \cdot 10^{-6} & -8,3804 \cdot 10^{-5} & -8,5141 \cdot 10^{-5} & 2,6485 \cdot 10^{-7} & -8,6540 \cdot 10^{-8} & 4, \\ 1,8504 \cdot 10^{-5} & -8,7701 \cdot 10^{-5} & -0,0001 & 4,8178 \cdot 10^{-8} & -1,9957 \cdot 10^{-7} & -4, \\ -6,2684 \cdot 10^{-5} & 3,7285 \cdot 10^{-5} & -2,0711 \cdot 10^{-5} & 1,7061 \cdot 10^{-7} & 2,1289 \cdot 10^{-7} & 6, \\ 0,0022 & 0,0007 & -0,0019 & -2,5173 \cdot 10^{-5} & 1,2967 \cdot 10^{-5} & 2, \end{bmatrix} \quad (16)$$

3.2 Positionsregelung

Neben der Zustandsvorhersage mit Hilfe des Extended Kalman-Filters wird innerhalb des beschriebenen Modells auch die Position und Rotation um die z -Achse der Drohne kontrolliert. Die dazu verwendete Regelung ist ebenfalls Santana et al. entnommen. Basierend auf vier einzeln konfigurierten PD-Reglern, lässt sich die Drohne an eine vorgegebene Position $P_d = [x_d \ y_d \ z_d \ \psi_d]$ steuern. Dies geschieht über den Steuervektor $\mathbf{u} = [u_z \ u_\psi \ u_\phi \ u_\theta]$ mit Hilfe der folgenden Gleichungen:

$$u_z = K_{p1}(z_d - z) - K_{d1}\dot{z} \quad (17)$$

$$u_\psi = K_{p2}(\psi_d - \psi) - K_{d2}\dot{\psi} \quad (18)$$

$$u'_\phi = K_{p3}(y_d - y) - K_{d3}\dot{y} \quad (19)$$

$$u'_\theta = K_{p4}(x_d - x) - K_{d4}\dot{x} \quad (20)$$

$$u_\phi = u'_\phi s_\psi - u'_\theta c_\psi \quad (21)$$

$$u_\theta = u'_\phi c_\psi + u'_\theta s_\psi \quad (22)$$

K_{p1} bis K_{p4} und K_{d1} bis K_{d4} stellen dabei Konstanten für den P-Anteil beziehungsweise den D-Anteil der Regler dar, die experimentell ermittelt werden müssen.

4 Modellierung mit MatLab/Simulink

Dieses Kapitel beschreibt die Modellierung des in 3 präsentierten Systems mithilfe der Entwicklungssoftware Simulink von The Mathworks. Dabei wird insbesondere auf die Verwendung eines bereits existierenden Extended Kalman-Filters aus der Simulink Bibliothek eingegangen. Zudem wird erläutert, wie MatLab zur Konfiguration der Systemparameter dient.

Das gesamte System ist als zeit-diskretes Modell aufgebaut, das mit δt in s parametrisiert werden kann. Dieser Parameter wird ebenso wie übrigen Konstanten über die *textInitFcn* in den *Callbacks* der *Block Properties* definiert. Weitere Konstanten sind beispielsweise die Kovarianzmatrizen des Prozess- und Messrauschens sowie die Beobachtungsmatrix H des Messdaten.

4.1 Extended Kalman-Filter in Simulink

Die *Control System Toolbox* aus Simulink Bibliothek stellt einen ausgereiften Block zur Zustandsvorhersage für nichtlineare Systeme mithilfe des Extended Kalman-Filters bereits zur Verfügung. Dieser Filterblock zeichnet sich durch ein großes Konfiguri-

onsspektrum aus. So können bis zu fünf Messgrößeneingänge eingestellt werden, die separat über eine Signalleitung aktiviert werden können, sobald Messdaten zur Verfügung stehen und in den Korrektor-Schritt mit eingehen sollen. Außerdem kann für jeden Eingang eine eigene Messfunktion $h(\mathbf{x}_k)$ und die Kovarianz des Messrauschens definiert werden.

Neben der genannten Messfunktion kann auch die in 3.1.1 beschriebene Zustandsübergangsfunktion $f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1})$ samt der Kovarianz des Prozessrauschens an den Filter übergeben werden. Hierbei kann konfiguriert werden, ob es sich um ein zeit-variantes beziehungsweise -invariantes Prozessrauschen handelt. In diesem Modell ist dieses als additives, zeit-invariantes Rauschen gewählt und mit einer Matrix $Q^{12 \times 12}$, deren Elemente alle den Wert 0,5 haben, geschrieben.

Des Weiteren kann zur Initialisierung des Filters der Systemzustand wie auch die Kovarianzvorhersage vorgegeben werden. !!! Womit initialisiert? !!! Diese kann bei Bedarf als Ausgang des Filters konfiguriert werden. In diesem Modell findet sie allerdings keine Verwendung. Standardmäßig ist hingegen aber der vorhergesagte Zustand ein Ausgang des Systems. Entsprechend liefert dieser den Zustandsvektor \mathbf{x}_k .

4.2 title

5 Zusammenfassung

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

Anhang

