OSTBAYERISCHE
TECHNISCHE HOCHSCHULE
REGENSBURG

Ostbayerische Technische Hochschule Regensburg
Fakultät für Informatik und Mathematik

Projektarbeit

in der Veranstaltung
Datenverarbeitung in der Technik

Quadkoptersimulation

Vorgelegt von: Markus Heimerl
Matrikelnummer: -
Studiengang: Technische Informatik

Erstgutachter: -
Zweitgutachter: -

Abgabedatum: 19.01.2022

Inhaltsverzeichnis

1 Simulation **1**
1.1 Motivation . . . 1
1.2 Zweidimensionale Simulation . . . 1
1.2.1 Zustand . . . 1
1.2.2 Dynamik . . . 2
1.2.3 Regelung . . . 4
1.3 Dreidimensionale Simulation . . . 8
1.3.1 Zustand . . . 8
1.3.2 Dynamik . . . 9
1.3.3 Regelung . . . 12

2 Ermittlung der Simulationsparameter **16**

Literatur **19**

1 Simulation

Markus Heimerl

1.1 Motivation

Um die Regelung für die Drohne entwerfen zu können und die dafür notwendigen Parameter experimentell zu ermitteln, ist es notwendig eine Simulation zu entwerfen, da es keinen systematischen oder analytischen Weg die Regelungsparameter zu ermitteln gibt, was Abstürze unvermeidlich macht. Diese sollten unbedingt virtuell stattfinden, sodass keine Schäden an Hardware, Körper der Drohne oder Laborequipment entsteht und die Sicherheit aller Mitarbeiter und Studierenden in den Laboren gewährleistet ist.

1.2 Zweidimensionale Simulation

1.2.1 Zustand

Eine Simulation ist dann als vollständig anzusehen, wenn sie den Zustand X der Drohne im nächsten diskreten Zeitabschnitt vorhersagen kann.

$$X = [z, y, \phi, \dot{z}, \dot{y}, \dot{\phi}] \tag{1}$$

Der Zustand X ist zum einen aufgebaut aus den Koordinaten der z-Achse, die positiv in Richtung Erdmitte zeigt, der y-Achse und dem Vektor ϕ, der den Drehwinkel um die x-Achse beschreibt. Zum anderen sind die Geschwindigkeiten, also Änderungsraten, $\dot{z}, \dot{y}, \dot{\phi}$ all dieser Größen ebenfalls teil des Vektors.

Der Schub F_1 und F_2 der beiden Rotoren, zusammen mit Umwelteinflüssen, ergeben alle Kräfte, die Änderungen am Zustandsvektor X zufolge haben.

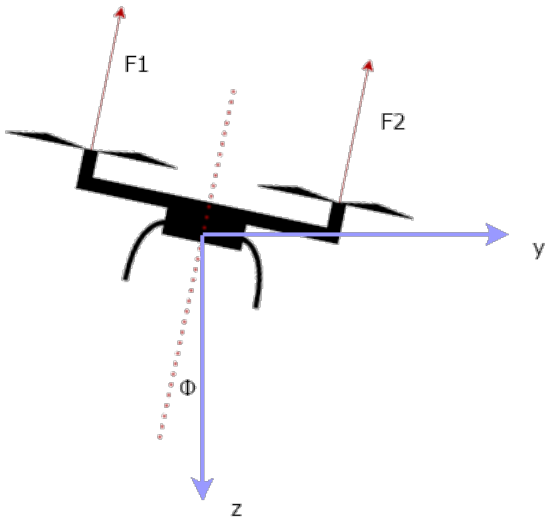

Abbildung 1: Drohne in zwei Dimensionen

Die Richtung der Schübe F_1 und F_2 stehen im Winkel ϕ zur z-Achse.

1.2.2 Dynamik

Die Dynamik beschreibt wie der Zustand X sich anhand von F_1 und F_2 ändert. Um mit fundamentalen Größen zu arbeiten, werden F_1 und F_2 im folgenden durch ihre zugrundeliegenden Rotorgeschwindigkeiten ω_1 und ω_2 dargestellt. Das Verhältnis k_f zwischen Rotorgeschwindigkeit und Kraft ist quadratisch [1, S. 4]. Somit ergibt sich:

$$F_i = k_f \omega_i^2 \tag{2}$$

Dieses Verhältnis k_f wird experimentell ermittelt wie in Kapitel 2 erläutert. Die Zustandsänderungen in den Positionen z, y und ϕ ergibt sich durch Integration der Geschwindigkeiten $\dot{z}, \dot{y}$ und $\dot{\phi}$. Die Zustandsänderungen dieser ist Resultat einer Integration der Beschleunigungen $\ddot{z}, \ddot{y}$ und $\dot{\phi}$. Diese ergeben sich aus $F = ma$ und der Erdanziehung $F_g = mg$ mit $g = 9.81 \frac{m}{s^2}$:

$$\ddot{z} = \frac{F_g - F_z}{m} = g - F_z \qquad \ddot{y} = \frac{F_y}{m} \tag{3}$$

F_z liegt parallel zur z-Achse und F_y liegt parallel zur y-Achse. Somit befindet sich ein rechter Winkel zwischen F_z und F_y. Es gilt zusätzlich

$$F_{gesamt} = \sum_{1}^{i} F_i = F_1 + F_2 \tag{4}$$

Die gesamte Kraft der beiden Rotoren F_{gesamt} steht im Winkel ϕ zur z-Achse.

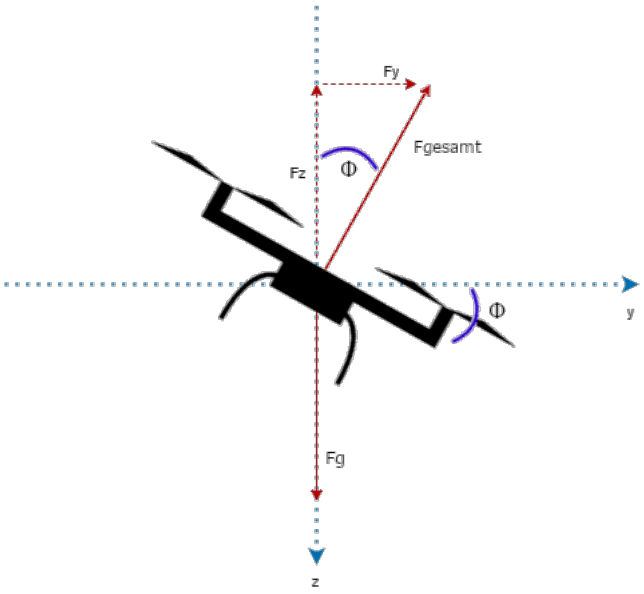

Abbildung 2: Kräfte einer 2D Drohne

Somit ergeben sich die Kräfte F_z und F_y durch die Trigonometrie im Dreieck als

$$F_z = F_{gesamt} \cos\phi \qquad F_y = F_{gesamt} \sin\phi \tag{5}$$

und die Beschleunigungen in entlang der z- und y-Achse durch Kombination von Gleichungen 3, 4 und 5:

$$\ddot{z} = g - \frac{(F_1 + F_2)\cos\phi}{m} \qquad \ddot{y} = \frac{(F_1 + F_2)\sin\phi}{m} \tag{6}$$

$\ddot{\phi}$ ist abgeleitet durch die Kombination der Gleichungen für den Drehmoment M. l ist der Abstand vom Mittelpunkt der Masse des Dualkopters zu einem der Motoren.

$$M_x = F_\phi l \qquad M_x = I_x \ddot{\phi} \tag{7}$$

Das Trägheitsmoment I_x muss experimentell, wie in Kapitel 2 beschrieben, ermittelt werden. Die Kraft um die x-Achse herum F_ϕ ist die Differenz der beiden Motorkräfte

$F1$ und $F2$ aus Abbildung 1. Positives F_ϕ bedeutet eine Drehung im Uhrzeigersinn und eine negativer Wert steht für Drehung gegen den Uhrzeigersinn. Zusammenfassend gilt also:

$$F_\phi = F_1 - F_2 \tag{8}$$

Somit lässt sich die Winkelbeschleunigung $\ddot{\phi}$ beschreiben als

$$\ddot{\phi} = \frac{(F_1 - F_2) \cdot l}{I_x} \tag{9}$$

Also ist der Zustand des Dualkopters zum Zeitpunkt t bestimmbar durch die Summe des Zustands $X(t)$ und dem Integral der Ableitung des Zustands $\dot{X}$.

$$X(t + \Delta t) = X(t) + \int_t^{t+\Delta t} \dot{X}(\tau) d\tau \tag{10}$$

Diese Gleichung beschreibt die Quintessenz der Dynamik des Dualkopters und ist deshalb hier nochmal in Python aufgeführt. Der Zustandsvektor X ist hier ein Array aus dem Paket numpy und genau so aufgebaut wie in Gleichung 1 beschrieben.

```
1        import numpy as np
2
3        def bringeZustandVoran(X, z_ddot, y_ddot, phi_ddot, dt):
4        # Bewegt den Zustand des Dualkopters vorwaerts durch die
             Zeit
5        X_dot = np.array([X[3],X[4],X[5],z_ddot,y_ddot,phi_ddot])
6        X = X + X_dot * dt
7        return X
```

Abbildung 3: Python Code für das Voranbringen des Zustands durch die Zeit in 2D

1.2.3 Regelung

Strecke Die Regelung bestimmt die Rotorgeschwindigkeiten ω_1 und ω_2 des Dualkopters so, dass eine Strecke geflogen wird. Es lässt sich jede gewünschte Strecke s, die der Dualkopter abfliegen soll, durch sechs Funktionen beschreiben. Für jeden Zeitpunkt t liefert

$$\begin{aligned} s_z(t) \quad & \textit{die gewünschte z-Koordinate,} \\ s_y(t) \quad & \textit{die gewünschte y-Koordinate.} \end{aligned} \tag{11}$$

Die Funktionen für dafür nötigen Geschwindigkeiten zu jedem Zeitpunkt t ergeben sich durch die Ableitungen der Streckenfunktionen

$$s_{\dot{z}}(t) = \frac{d}{dt} s_z(t) \quad s_{\dot{y}}(t) = \frac{d}{dt} s_y(t) \tag{12}$$

und die Ableitung wiederum dieser resultieren in den Funktionen der notwendigen Beschleunigungen zu jedem Zeitpunkt t.

$$s_{\ddot{z}}(t) = \frac{d}{dt} s_{\dot{z}}(t) \quad s_{\ddot{y}}(t) = \frac{d}{dt} s_{\dot{y}}(t) \tag{13}$$

Es gibt keine Streckeninformation über den Winkel ϕ, da dieser benutzt wird um die Änderung in y-Richtung zu realisieren. Der Dualkopter kann nur indirekt über ϕ-Änderungen seine y-Position verändern.

Regelgleichungen Es werden die Größen $F_{gesamt,ziel}$ und $M_{x,ziel}$ durch die drei PD-Regler kontrolliert. Es gibt den Höhenregler für z, den Seitenregler für y und dem Haltungsregler für ϕ. Da Ergebnisse eines Reglers von anderen verwendet wird, nämlich die des Höhenreglers im Seitenregler und die des Seitenreglers im Haltungsregler, spricht man von einer kaskadierenden Regelarchitektur.

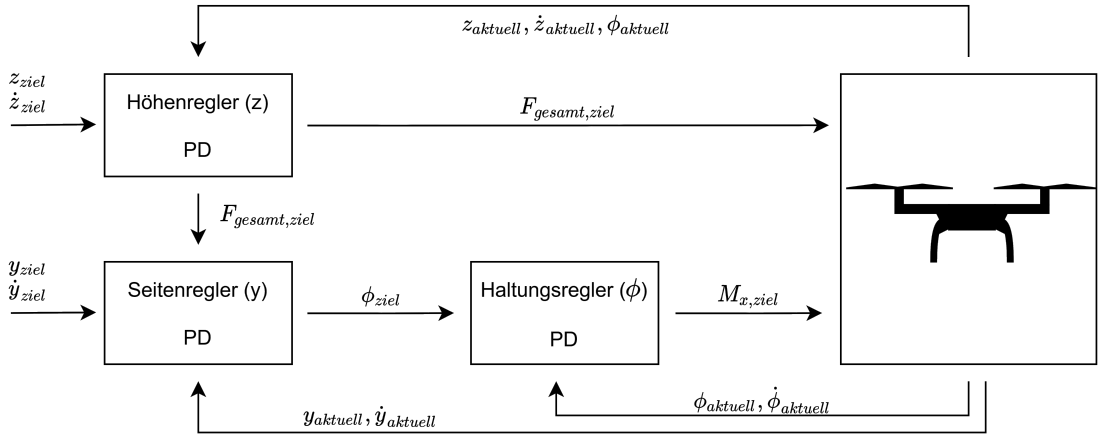

Abbildung 4: Kaskadierende Regelarchitektur in zwei Dimensionen

Höhenregler Die notwendige Zielbeschleunigung $\ddot{z}_{ziel}$, was oft in Fachliteratur als u_1 gekennzeichnet wird, um den Positionsfehler zur gewünschten z-Koordinate $s_z(t) = z_{ziel}$ zum Zeitpunkt t exponentiell zu verringern, ergibt sich somit aus folgender Differentialgleichung zweiter Ordnung

$$\ddot{z}_{ziel} = k_{z,p}(z_{ziel} - z_{aktuell}) + k_{z,d}(\dot{z}_{ziel} - \dot{z}_{aktuell}) \tag{14}$$

Alle Parameter k werden durch Raten in der Simulation ermittelt. Es gilt außerdem $\dot{s}_z(t) = \dot{z}_{ziel}$ Aus dem Resultat kann, durch Umstellen der Gleichung 6 links, der notwendige Gesamtschub $F_{gesamt,ziel}$ errechnet werden.

$$F_{gesamt,ziel} = \frac{m(g - \ddot{z}_{ziel})}{\cos \phi_{aktuell}} \tag{15}$$

Seitenregler Analog dazu ergibt sich für die Zielbeschleunigung $\ddot{y}_{ziel}$ unter Berücksichtigung von $s_y(t) = y_{ziel}$ und $\dot{s}_y(t) = \dot{y}_{ziel}$

$$\ddot{y}_{ziel} = k_{y,p}(y_{ziel} - y_{aktuell}) + k_{y,d}(\dot{y}_{ziel} - \dot{y}_{aktuell}) \tag{16}$$

Und hieraus kann, durch umstellen der Gleichung 6 rechts und Verwendung des Resultats des ersten PD-Reglers $F_{gesamt,ziel}$ aus Gleichung 15, das notwendige ϕ_{ziel} ermittelt werden.

$$\phi_{ziel} = \sin^{-1}(\frac{m\ddot{y}_{ziel}}{F_{gesamt,ziel}}) \tag{17}$$

ϕ_{ziel} ist somit der Winkel in dem die Drohne stehen muss, um zu $s_y(t)$ zu translatierten.

Haltungsregler Durch dieses Ergebnis kann dann die Fehlerberechnung im letzten PD-Regler festgelegt werden als

$$\ddot{\phi}_{ziel} = k_{\phi,p}(\phi_{ziel} - \phi_{aktuell}) + k_{\phi,d}(\dot{\phi}_{ziel} - \dot{\phi}_{aktuell}) \tag{18}$$

wo immer $\dot{\phi}_{ziel} = 0$.

Hieraus erschließt sich dann das nötige Drehmoment $M_{x,ziel}$, oft als u_2 benannt, durch Verwendung von Gleichung 7 rechts.

$$M_{x,ziel} = I_x \ddot{\phi}_{ziel} \tag{19}$$

Resultierende Rotorgeschwindigkeiten Aus den beiden errechneten Zielgrößen $F_{gesamt,ziel}$ und $M_{x,ziel}$ lassen sich die beiden Rotorgeschwindigkeiten $\omega_{1,ziel}$ und $\omega_{2,ziel}$ durch Verwendung der Gleichungen 7 links, 4 und 2 ausmachen.

$$\begin{aligned}
M_{x,ziel} &= F_{\phi_{ziel}} l = (F_{1,ziel} - F_{2,ziel}) l \\
F_{gesamt,ziel} &= F_{1,ziel} + F_{2,ziel} \\
F_{1,ziel} &= \frac{F_{gesamt,ziel}}{2} + \frac{M_{x,ziel}}{2l} \\
F_{2,ziel} &= \frac{F_{gesamt,ziel}}{2} - \frac{M_{x,ziel}}{2l} \\
\omega_{1,ziel} &= \sqrt{\frac{F_{1,ziel}}{k_f}} \\
\omega_{2,ziel} &= \sqrt{\frac{F_{2,ziel}}{k_f}}
\end{aligned} \tag{20}$$

1.3 Dreidimensionale Simulation

1.3.1 Zustand

Der Zustandsvektor X besteht im dreidimensionalen Fall aus zwölf Größen. Formal ist er definiert als

$$X = [x,y,z,\phi,\theta,\psi,\dot{x},\dot{y},\dot{z},p,q,r] \tag{21}$$

wo

x, y, z die Positionen darstellen,

$\dot{x}, \dot{y}, \dot{z}$ die Geschwindigkeiten in der jeweiligen Achse vertreten,

ϕ, θ, ψ, auch Eulerwinkel genannt, die Drehungen um die x-Achse, y-Achse und z-Achse repräsentieren und

p, q, r die Geschwindigkeiten um die Achsen kx, ky und kz im Körperkoordinatensystem sind.

Bis auf die letzten drei Parameter befinden sich alle Informationen im Weltkoordinatensystem, welches das einzige in Verwendung war im zweidimensionalen Fall aus Kapitel 1.2. Das Körperkoordinatensystem hingegen ist ein fixes, an dem Quadrokopter befestigtes Koordinatensystem.

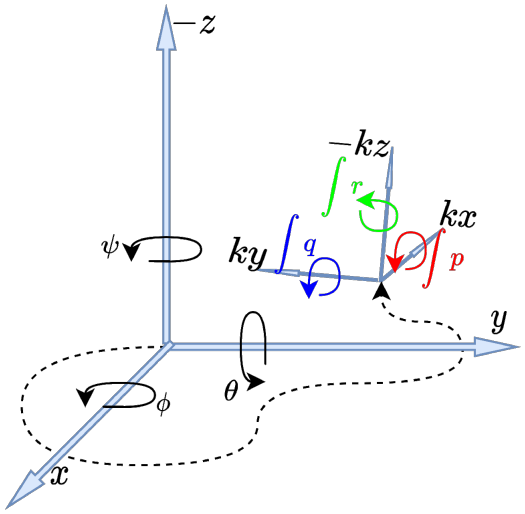

Abbildung 5: Welt- und Körperkoordinatensystem

1.3.2 Dynamik

Auch hier gilt Gleichung (2) nach wie vor. Analog zu Formel 10 benötigen wir auch im dreidimensionalen Fall die Ableitung $\dot{X}$ des Zustandvektors, um den Zustand vorwärts durch die Zeit zu bewegen. Gleichungen 3 gelten nach wie vor aber hinzu kommt, analog zu $\ddot{y}$, die Formel für $\ddot{x}$. Zusammengefasst, wiederholt und immer noch abgeleitet aus Newtons $F = ma$ gilt also

$$\ddot{z} = \frac{F_g - F_z}{m} = g - F_z \qquad \ddot{y} = \frac{F_y}{m} \qquad \ddot{x} = \frac{F_x}{m} \tag{22}$$

Anders sind im dreidimensionalen Fall aber die Formeln für F_x, F_y und F_z. Hier wird nun eine Rotationsmatrix [2, S. 12] definiert, die jeden Vektor aus dem Körperkoordinatensystem in einen Vektor im Weltkoordinatensystem durch Multiplikation transformiert. In Essenz baut sich diese immer noch aus der Trigonometrie im Dreieck auf. Die Matrix lautet

$$R^{Welt}_{Körper} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} =$$

$$= \begin{bmatrix} c(\psi)c(\theta) & c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi) & s(\psi)s(\phi) + c(\psi)s(\theta)c(\phi) \\ s(\psi)c(\theta) & c(\psi)c(\phi) + s(\psi)s(\theta)s(\phi) & s(\psi)s(\theta)c(\phi) - c(\psi)s(\phi) \\ -s(\theta) & c(\theta)s(\phi) & c(\theta)c(\phi) \end{bmatrix} \tag{23}$$

Somit ergeben sich aus Gleichungen 22 und 23 die Formeln für die Achsenbeschleunigungen im Weltkoordinatensystem als

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} + \frac{1}{m} R^{Welt}_{Körper} \begin{bmatrix} 0 \\ 0 \\ -F_{gesamt} \end{bmatrix} = \begin{bmatrix} R_{13}\frac{1}{m}(-F_{gesamt}) \\ R_{23}\frac{1}{m}(-F_{gesamt}) \\ g + R_{33}\frac{1}{m}(-F_{gesamt}) \end{bmatrix} \tag{24}$$

Hier zeigt sich außerdem, dass die Propeller Schub in Richtung $-kz$ erzeugen. Auf der Erde gilt $g = 9{,}81\frac{m}{s^2}$. Und die gesamte Kraft ergibt sich nun aus der Summe aller vier Rotoren, sodass

$$F_{gesamt} = \sum_{1}^{i} F_i = F_1 + F_2 + F_3 + F_4 \tag{25}$$

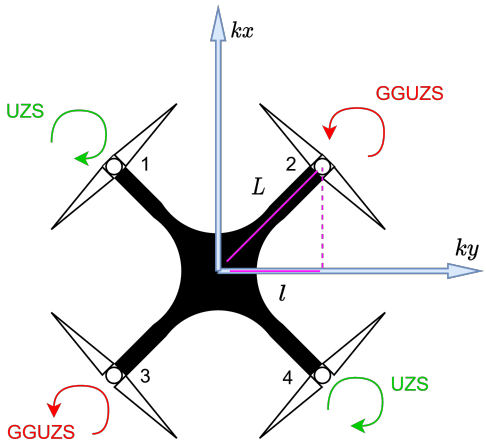

Abbildung 6: Rotationsrichtungen und Nummern der Propeller. kz geht in die Zeichenebene hinein.

Die Beschleunigungen um die Körperachsen $\dot{p}$, $\dot{q}$ und $\dot{r}$ werden durch die Bewegungsgleichung von Euler berechnet, welche lautet

$$\begin{aligned} M &= I\dot{\omega} + \omega \times (I\omega) = \\ \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} &= \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \left(\begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right) \end{aligned} \tag{26}$$

Die Trägheitsmomente I_x, I_y und I_z werden durch Experimente, wie in Kapitel 2 erläutert, ermittelt. Umstellung ergeben sich dann die Beschleunigungen als

$$\begin{aligned} \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} &= \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} qI_xr - rI_yq \\ rI_xp - pI_zr \\ pI_yq - qI_xp \end{bmatrix} \\ \begin{bmatrix} M_x - (qI_xr - rI_yq) \\ M_y - (rI_xp - pI_zr) \\ M_z - (pI_yq - qI_xp) \end{bmatrix} &= \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \\ \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} &= \begin{bmatrix} \frac{1}{I_x} & 0 & 0 \\ 0 & \frac{1}{I_y} & 0 \\ 0 & 0 & \frac{1}{I_z} \end{bmatrix} \begin{bmatrix} M_x - (qI_xr - rI_yq) \\ M_y - (rI_xp - pI_zr) \\ M_z - (pI_yq - qI_xp) \end{bmatrix} \\ \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} &= \begin{bmatrix} \frac{M_x - rq(I_x - I_y)}{I_x} \\ \frac{M_y - rp(I_x - I_z)}{I_y} \\ \frac{M_z - pq(I_y - I_x)}{I_z} \end{bmatrix} \end{aligned} \tag{27}$$

Als verbleibende notwendige Größen, um $\dot{X}$ aufzustellen, was das Ziel in der Dynamik ist, verweilen die Geschwindigkeiten um die Weltkoordinatensystemachsen $\dot{\phi}$, $\dot{\theta}$ und $\dot{\psi}$. Diese ergeben sich, wie die linearen Beschleunigungen in Gleichung 24 mithilfe einer Rotationsmatrix speziell für die Eulerwinkelgeschwindigkeiten [3, S. 12, Gl. 79]

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{28}$$

Also ist der Zustand des Quadrokopters zum Zeitpunkt t bestimmbar durch die Summe des Zustands $X(t)$ und dem Integral der Ableitung des Zustands $\dot{X}$.

$$X(t+\Delta t) = X(t) + \int_t^{t+\Delta t} \dot{X}(\tau)d\tau \tag{29}$$

Diese Gleichung ist hier in Python aufgeführt. Der Zustandsvektor X ist hier ein Array aus dem Paket numpy und genau so aufgebaut wie in Gleichung 21 beschrieben.

```
1        import numpy as np
2
3        def bringeZustandVoran(X, phi_dot, theta_dot, psi_dot,
             z_ddot, y_ddot, phi_ddot, p_dot, q_dot, r_dot dt):
4        # Bewegt den Zustand des Quadrokopters um dt Sekunden in
             positive Richtung durch die Zeit.
5        X_dot = np.array([X[6],X[7],X[8],phi_dot,theta_dot,psi_dot
             ,z_ddot,y_ddot,phi_ddot,p_dot,q_dot,r_dot])
6        X = X + X_dot * dt
7        return X
```

Abbildung 7: Python Code für das Voranbringen des Zustands durch die Zeit in 3D

1.3.3 Regelung

Strecke Die Regelung bestimmt die Rotorgeschwindigkeiten ω_1, ω_2, ω_3 und ω_4 des Quadrokopters so, dass eine Strecke geflogen wird. Es lässt sich jede gewünschte Strecke s, die der Quadrokopter abfliegen soll, durch zwölf Funktionen beschreiben. Für jeden Zeitpunkt t liefert

$$\begin{aligned} s_x(t) &\quad \textit{die gewünschte x-Koordinate,} \\ s_y(t) &\quad \textit{die gewünschte y-Koordinate,} \\ s_z(t) &\quad \textit{die gewünschte z-Koordinate,} \\ s_\psi(t) &\quad \textit{der gewünschte } \psi\textit{-Winkel.} \end{aligned} \tag{30}$$

Die Funktionen für dafür nötigen Geschwindigkeiten zu jedem Zeitpunkt t ergeben sich durch die Ableitungen der Streckenfunktionen

$$s_{\dot{x}}(t) = \frac{d}{dt}s_x(t) \quad s_{\dot{y}}(t) = \frac{d}{dt}s_y(t) \quad s_{\dot{z}}(t) = \frac{d}{dt}s_z(t) \tag{31}$$

und die Ableitung wiederum dieser resultieren in den Funktionen der notwendigen Beschleunigungen zu jedem Zeitpunkt t.

$$s_{\ddot{x}}(t) = \frac{d}{dt}s_{\dot{x}}(t) \quad s_{\ddot{y}}(t) = \frac{d}{dt}s_{\dot{y}}(t) \quad s_{\ddot{z}}(t) = \frac{d}{dt}s_{\dot{z}}(t) \tag{32}$$

Es gibt keine Streckeninformation über die Winkel ϕ und θ, da diese benutzt werden, um die Änderung in y- und x-Richtung zu realisieren. Der Quadrokopter kann nur indirekt über diese Winkeländerungen seine laterale Position verändern.

Regelgleichungen Es werden die Größen $F_{gesamt,ziel}$, $M_{x,ziel}$, $M_{y,ziel}$ und $M_{z,ziel}$ durch zwei PD-Regler und drei P-Regler kontrolliert. Es gibt den Höhenregler für z, den Seitenregler für x und y, den Roll-Nick-Regler für ϕ und θ, den Gier-Regler für ψ und den Körper-Raten-Regler für p, q und r. Da Ergebnisse eines Reglers von anderen verwendet wird, nämlich die des Seitenreglers im Roll-Nick-Regler, die des Roll-Nick-Regler im Körper-Raten-Regler und die des Gier-Regler im Körper-Raten-Regler spricht man von einer kaskadierenden Regelarchitektur.

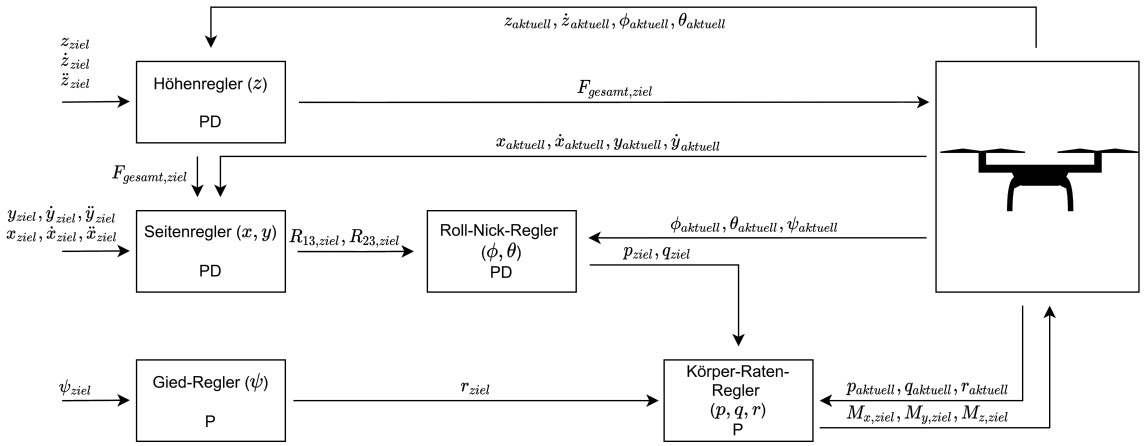

Abbildung 8: Kaskadierende Regelarchitektur in drei Dimensionen

Höhenregler Die notwendige Zielbeschleunigung $\ddot{z}_{ziel}$, was oft in Fachliteratur als u_1 gekennzeichnet wird, um den Positionsfehler zur gewünschten z-Koordinate $s_z(t) = z_{ziel}$ zum Zeitpunkt t exponentiell zu verringern, ergibt sich somit aus folgender Differentialgleichung zweiter Ordnung

$$\ddot{z}_{ziel} = k_{z,p}(z_{ziel} - z_{aktuell}) + k_{z,d}(\dot{z}_{ziel} - \dot{z}_{aktuell}) + \ddot{z}_{vorschub} \tag{33}$$

Alle Parameter k werden durch Ausprobieren in der Simulation ermittelt. Es gilt außerdem $\dot{s}_z(t) = \dot{z}_{ziel}$, $\ddot{s}_z(t) = \ddot{z}_{vorschub}$. Aus dem Resultat kann, durch Umstellen der untersten Zeile von Gleichung 24, der notwendige Gesamtschub $F_{gesamt,ziel}$ errechnet werden.

$$F_{gesamt,ziel} = -\frac{m}{R_{33}}(\ddot{z}_{ziel} - g) = -\frac{m}{\cos\phi\cos\theta}(\ddot{z}_{ziel} - g) \tag{34}$$

Seitenregler Analog dazu ergibt sich für die Zielbeschleunigung $\ddot{x}_{ziel}$ und $\ddot{y}_{ziel}$ unter Berücksichtigung von $s_x(t) = x_{ziel}$, $\dot{s}_x(t) = \dot{x}_{ziel}$, $\ddot{s}_x(t) = \ddot{x}_{vorschub}$, $s_y(t) = y_{ziel}$, $\dot{s}_y(t) = \dot{y}_{ziel}$ und $\ddot{s}_y(t) = \ddot{y}_{vorschub}$.

$$\begin{aligned} \ddot{x}_{ziel} &= k_{x,p}(x_{ziel} - x_{aktuell}) + k_{x,d}(\dot{y}_{ziel} - \dot{x}_{aktuell}) + \ddot{x}_{vorschub} \\ \ddot{y}_{ziel} &= k_{y,p}(y_{ziel} - y_{aktuell}) + k_{y,d}(\dot{y}_{ziel} - \dot{y}_{aktuell}) + \ddot{y}_{vorschub} \end{aligned} \tag{35}$$

Und hieraus kann, durch umstellen der ersten beiden Zeilen aus Gleichung 24 und Verwendung des Resultats des ersten PD-Reglers $F_{gesamt,ziel}$ aus Gleichung 34, das notwendige $R_{13,ziel}$ und $R_{23,ziel}$ ermittelt werden.

$$R_{13,ziel} = -\frac{m\ddot{x}_{ziel}}{F_{gesamt,ziel}} \qquad R_{23,ziel} = -\frac{m\ddot{y}_{ziel}}{F_{gesamt,ziel}} \tag{36}$$

$R_{13,ziel}$ und $R_{23,ziel}$ sind die Werte, die in der Rotationsmatrix an ihren Indexen auftreten müssen, sodass die Drohne nach $s_x(t)$ und $s_y(t)$ translatiert.

Roll-Nick-Regler Durch dieses Ergebnis kann dann die Fehlerberechnung im Roll-Nick-Regler festgelegt werden als

$$\begin{aligned} \dot{R_{13,ziel}} &= k_{roll,p}(R_{13,ziel} - R_{13,aktuell}) \\ \dot{R_{23,ziel}} &= k_{nick,p}(R_{23,ziel} - R_{23,aktuell}) \end{aligned} \tag{37}$$

Es lässt sich jedoch nicht auf trivialen Weg aus bisher etablierten Gleichungen die nötigen Größen p_{ziel} und q_{ziel}. Hingegen wird folgende Formel aus Publikation [4, S. 4315, Gl. 6] entnommen und ohne Herleitung verwendet:

$$\begin{bmatrix} p_{ziel} \\ q_{ziel} \end{bmatrix} = \frac{1}{R_{33}} \begin{bmatrix} R_{21} & -R_{11} \\ R_{22} & -R_{12} \end{bmatrix} \begin{bmatrix} \dot{R_{13,ziel}} \\ \dot{R_{23,ziel}} \end{bmatrix} \tag{38}$$

Gier-Regler Hier wird nur noch der ψ-Fehler durch einen P-Regler minimiert und die Vereinfachung angenommen, dass $\dot{\psi}_{ziel} = r_{ziel}$, so dass

$$r_{ziel} = k_{gier,p}(\psi_{ziel} - \psi_{aktuell}) \tag{39}$$

Körper-Raten-Regler Zuletzt sind $\dot{p}_{ziel}$, $\dot{q}_{ziel}$ und $\dot{r}_{ziel}$ definiert als

$$\begin{aligned} \dot{p}_{ziel} &= k_{p,p}(p_{ziel} - p_{aktuell}) \\ \dot{q}_{ziel} &= k_{q,p}(q_{ziel} - q_{aktuell}) \\ \dot{r}_{ziel} &= k_{r,p}(r_{ziel} - r_{aktuell}) \end{aligned} \tag{40}$$

Durch Anpassen der Gleichung 26 für das Weltkoordinatensystem (vergleichbar nun mit z.B. Formel 19), ergibt sich für die zu steuernden Momente:

$$M = I\dot{\omega} = \begin{bmatrix} M_{x,ziel} \\ M_{y,ziel} \\ M_{z,ziel} \end{bmatrix} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \begin{bmatrix} \dot{p}_{ziel} \\ \dot{q}_{ziel} \\ \dot{r}_{ziel} \end{bmatrix} \tag{41}$$

Resultierende Rotorgeschwindigkeiten Aus den vier errechneten Zielgrößen $F_{gesamt,ziel}$, $M_{x,ziel}$, $M_{y,ziel}$, $M_{z,ziel}$ lassen sich die vier Rotorgeschwindigkeiten $\omega_{1,ziel}$, $\omega_{2,ziel}$, $\omega_{3,ziel}$ und $\omega_{4,ziel}$ durch Verwendung der Gleichungen 2, 25, 42 und 43 ausmachen. Mit $l = \frac{L\sqrt{2}}{2}$, L gleich der Länge von der Mitte des Quadrokopters zu einem der Motoren (vgl. Abbildung 6) und k_m entsprechend dem Verhältnis zwischen Drehgeschwindigkeit und Moment der Motoren. Die experimentelle Ermittlung davon ist in Kapitel 2 aufgeführt. Gleichung 42 lässt sich mit ähnlichen Überlegungen wie für Rechnung 8 aufstellen.

$$\begin{aligned} M_x &= l((F_1 + F_3) - (F_2 + F_4)) \\ M_y &= l((F_1 + F_2) - (F_3 + F_4)) \\ M_z &= (M_1 + M_4) - (M_2 + M_3) \end{aligned} \tag{42}$$

$$M_i = k_m \omega_i^2 \tag{43}$$

$$\begin{bmatrix} \frac{F_{gesamt,ziel}}{k_f} \\ \frac{M_{x,ziel}}{lk_f} \\ \frac{M_{y,ziel}}{lk_f} \\ \frac{M_{z,ziel}}{k_m} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} \omega_{1,ziel}^2 \\ \omega_{2,ziel}^2 \\ \omega_{3,ziel}^2 \\ \omega_{4,ziel}^2 \end{bmatrix}$$
$$\begin{bmatrix} \omega_{1,ziel}^2 \\ \omega_{2,ziel}^2 \\ \omega_{3,ziel}^2 \\ \omega_{4,ziel}^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} & \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & -\frac{1}{4} & \frac{1}{4} & -\frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} \\ \frac{1}{4} & -\frac{1}{4} & -\frac{1}{4} & \frac{1}{4} \end{bmatrix} \begin{bmatrix} \frac{F_{gesamt,ziel}}{k_f} \\ \frac{M_{x,ziel}}{lk_f} \\ \frac{M_{y,ziel}}{lk_f} \\ \frac{M_{z,ziel}}{k_m} \end{bmatrix} \tag{44}$$

2 Ermittlung der Simulationsparameter

Markus Heimerl

Neben der Gesetze der Dynamik bestimmen noch einige zentrale Größen das Verhalten des Quadrokopters. Diese sind

m die Masse,

k_f die Konstante für das quadratische Verhältnis zwischen Rotorgeschwindigkeit und Schub,

k_m die Konstante für das quadratische Verhältnis zwischen Rotorgeschwindigkeit und Moment,

I_x, I_y, I_z die Trägheitsmomente um jede Achse.

Diese Konstanten müssen durch Experimente ermittelt werden.
Die Masse wird über eine Badezimmerwaage gemessen. Ein Projektmitglied wiegt sich selbst und dann während er die Drohne nahe seines Oberkörper hält. Die Differenz ergibt die Masse m.
k_f und k_m werden mittels einer Wippe gemessen.

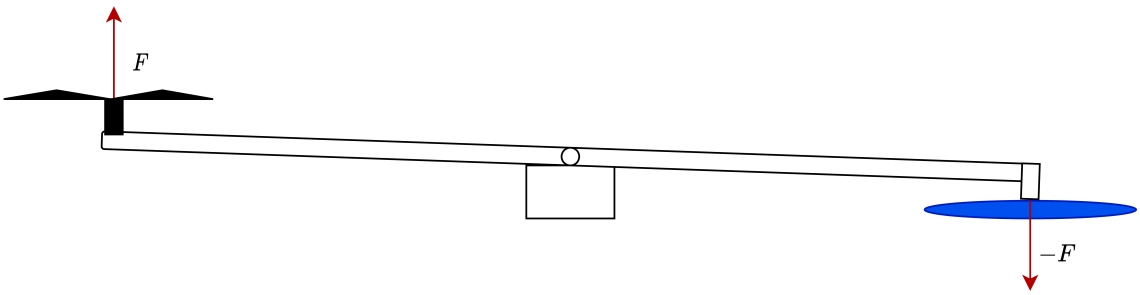

Abbildung 9: Messung von k_f. Die Küchenwaage ist blau eingefärbt.

An einem Ende wird der Propeller befestigt und am anderen Ende drückt ein Stift die Kraft des Schubes auf eine Küchenwaage. Durch Ablesen des Gewichts auf der Waage kann die resultierende Kraft ermittelt werden.

Abbildung 10: Messung von k_f

k_m kann durch eine leichte Modifikation der Wippe ebenfalls gemessen werden. Der Propeller muss lediglich in einem 90° Winkel zur Wippe stehen.

Abbildung 11: Messung von k_m

Die Drehzahl wird durch ein Tachometer oder Drehzahlmessgerät festgehalten. Dazu wird ein Reflexionsstreifen am Propeller befestigt. Das Tachometer wirft einen Laser und zählt die Häufigkeit an Reflexionen pro Minute.

Abbildung 12: Messung von ω mit einem Tachometer

Drehanzahl [rads/s]	**Messung** k_f [N]	**Messung** k_m [N]
20.00	0.56898	0.11772
26.70	0.99081	0.30411
35.08	1.74618	0.37278
41.26	2.43288	0.6867
48.17	3.2373	0.981
53.20	4.08096	1.12815
58.85	4.93443	1.30473
62.41	5.4936	1.52055
65.97	6.0822	1.6677

Tabelle 1: Ergebnisse der Experimente

Die Messreihen haben ein $k_f = 0.00141446535$ und ein $k_m = 0.0004215641$ ergeben. Die Trägheitsmomente sind abhängig vom Gewicht und den Dimensionen des Drohnenkörpers. In dieser Arbeit werden Ergebnisse einer Publikation, in der ein ähnlicher Rahmen verwendet wird, angenommen.[5, S. 7] Es ergeben sich also $I_x = 0.0121$, $I_y = 0.0119$ und $I_z = 0.0223$.

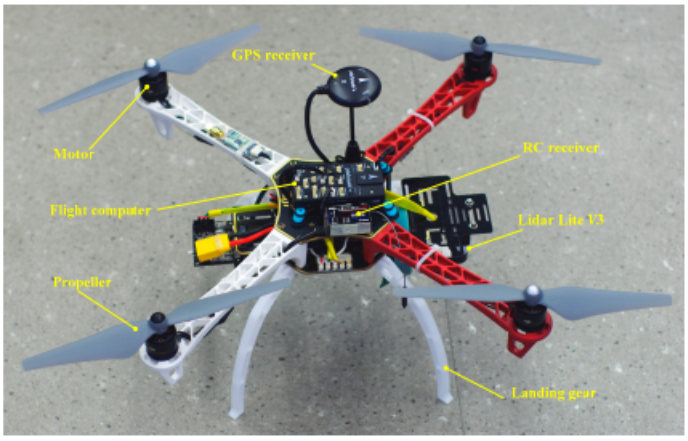

Figure 5. Quadcopter platform used for the experimental demonstration.

Table 1. System parameters used for the simulations.

Parameter	Value	Unit
m	2.0	kg
J_x	0.0121	kg.m^2
J_y	0.0119	kg.m^2
J_z	0.0223	kg.m^2
l	0.23	m
g	9.81	m/s^2
C_t	7.732 (10^{-6})	Ns^2
C_b	1.275 (10^{-7})	Nms^2

Abbildung 13: Messung von I_x, I_y und I_z

Literatur

[1] F. I. T. Johansen Tor and S. Berge, "Constrained Nonlinear Control Allocation with Singularity Avoidance using Sequential Quadratic Programming," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 1, pp. 211–216, 2004.

[2] L. F. L. Johnson Eric N. and S. B. L, *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*. New Jersey: Wiley-Blackwell, 2015.

[3] D. James, "Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors," *self*, 2006.

[4] A. Schoellig, C. Wiltsche, and R. D'Andrea, "Feed-forward parameter identification for precise periodic quadrocopter motions," *Proceedings of the American Control Conference*, pp. 4313–4318, 06 2012.

[5] N. Xuan-Mung and S.-K. Hong, "Improved altitude control algorithm for quadcopter unmanned aerial vehicles," *Applied Sciences*, vol. 9, no. 10, 2019.