

Einführung in moderne Simulationsmethoden: MATLAB/Simulink

Prüfungsaufgabe HT 2019: Quadrocopter



Abbildung 1: Beispiel eines Quadrocopters

In dieser Prüfungsaufgabe sollen die im Kursteil MATLAB/Simulink erlernten Fähigkeiten angewendet werden. Hierzu soll innerhalb der MATLAB-Toolbox Simulink ein Simulationsmodell für einen Quadrocopter erstellt werden.

Bewegungsgleichungen

Dargestellt sind die Bewegungsgleichungen eines Quadrocopter-Systems. Dabei werden sowohl das körperfeste (rotatorische) als auch das erdfeste (translatorische) Koordinatensystem verwendet. Als Stellgrößen werden der Schub in z -Richtung (u_1) sowie das Nick- (u_2), Roll- (u_3) und Giermoment (u_4) im körperfesten Koordinatensystem verwendet. Der Parameter L beschreibt den Abstand der Rotoren vom Schwerpunkt, mit I_x , I_y und I_z werden die Massenträgheiten um die jeweiligen Koordinaten bezeichnet und der Parameter J_R stellt die Trägheit der Motoren dar.

Rotatorische Dynamik

$$\ddot{\phi} = \dot{\theta}\dot{\psi} \left(\frac{I_y - I_z}{I_x} \right) - \dot{\theta} \frac{J_R}{I_x} \Omega_R + \frac{L}{I_x} u_2 \quad (1)$$

$$\ddot{\theta} = \dot{\phi}\dot{\psi} \left(\frac{I_z - I_x}{I_y} \right) - \dot{\phi} \frac{J_R}{I_y} \Omega_R + \frac{L}{I_y} u_3 \quad (2)$$

$$\ddot{\psi} = \frac{1}{I_z} u_4 \quad (3)$$

Translatorische Dynamik

$$\ddot{x} = -\frac{u_1}{m} (\cos(\phi) \sin(\theta) \cos(\psi) + \sin(\phi) \sin(\psi)) \quad (4)$$

$$\ddot{y} = -\frac{u_1}{m} (\cos(\phi) \sin(\theta) \sin(\psi) - \sin(\phi) \cos(\psi)) \quad (5)$$

$$\ddot{z} = \frac{u_1}{m} (\cos(\phi) \cos(\theta)) - g \quad (6)$$

Umrechnung Drehzahlen in Kräfte/Momente

Die Bewegungsgleichungen des Quadropters beinhalten den auf das System wirkenden Schub und die Momente. Diese werden jedoch bei einem Quadropter durch Anpassung der Drehzahlen Ω_i realisiert. Für die vollständige Modellierung wird deshalb zusätzlich der Zusammenhang zwischen Drehzahlen und den wirkenden Kräften bzw. Momenten benötigt. Die Parameter b und d bilden die aerodynamischen Eigenschaften der Propeller näherungsweise ab.

$$u_1 = b (\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \quad (7)$$

$$u_2 = b (-\Omega_2^2 + \Omega_4^2) \quad (8)$$

$$u_3 = b (\Omega_1^2 - \Omega_3^2) \quad (9)$$

$$u_4 = d (\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \quad (10)$$

$$\Omega_R = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4 \quad (11)$$

Parameter

Die für das Modell zu verwendende Parameter lauten wie folgt:

Trägheitsmomente	Gewicht	Geometrie	Rotorparameter
$I_x = 0.0081 \text{ kgm}^2$	$m = 1 \text{ kg}$	$L = 0.24 \text{ m}$	$b = 5.42 \cdot 10^{-5}$
$I_y = 0.0081 \text{ kgm}^2$	$g = 9.81 \text{ m/s}^2$		$d = 1.10 \cdot 10^{-6}$
$I_z = 0.0142 \text{ kgm}^2$			
$J_R = 0.0001 \text{ kgm}^2$			

Tabelle 1: Systemparameter

Aufgaben

Die Bewegungsgleichungen des Quadropters sollen in Form von zwei Subsystemen in einem Simulink-Modell realisiert werden.

- Das erste Subsystem für die rotatorische Dynamik, soll als Eingänge die drei Steuermomente Nick- (u_2), Roll- (u_3) und Giermoment (u_4) und die relative Drehzahl Ω_R haben. Als Ausgänge sind die sechs rotatorischen Zustände der Differentialgleichungen (Winkel ϕ , θ , ψ sowie Winkelgeschwindigkeiten $\dot{\phi}$, $\dot{\theta}$, $\dot{\psi}$) auszugeben. Innerhalb des rotatorischen Subsystems soll die Dynamik der Differentialgleichungen (1), (2) und (3) simuliert werden. Alle erforderlichen Parameter und

Startwerte sollen so implementiert werden, dass diese variabel sind und aus dem MATLAB-Workspace eingelesen werden. Sehen Sie dies in den Simulink-Blöcken daher entsprechend vor.

- b) Im zweiten Subsystem soll die translatorische Dynamik des Quadrocopters aus den Gleichungen (4), (5) und (6) simuliert werden. Als Eingänge benötigt dieses Subsystem den Schub u_1 sowie die drei Lagewinkel ϕ , θ und ψ des Quadrocopters. Als Ausgänge sollen sowohl die Positionen x , y , z , die Geschwindigkeiten \dot{x} , \dot{y} , \dot{z} , als auch die Beschleunigungen \ddot{x} , \ddot{y} , \ddot{z} ausgegeben werden. Analog zu Aufgabenteil a) sollen auch hier alle Parameter und Startwerte variabel gestaltet und aus dem MATLAB-Workspace eingelesen werden.

Wie oben im Text beschrieben, werden der Schub und die Steuermomente des Quadrocopters durch Variation der Drehzahlen Ω_1 , Ω_2 , Ω_3 und Ω_4 erzeugt.

- c) Erstellen Sie einen MATLAB-Funktionsblock in Simulink, welcher aus den vier Drehzahlen den Schub und die Momente nach den Gleichungen (7), (8), (9) und (10) erzeugt. Überlegen Sie hierzu, wie sich dies in MATLAB möglichst effizient mittels Matrix-Vektor-Multiplikation berechnen lässt.
- d) Sehen Sie außerdem eine Funktion vor, die aus den vier Drehzahlen die relative Drehzahl Ω_R entsprechend Gleichung (11) berechnet.

Nun sind alle Subsysteme, die für die Simulation der Quadrocopter-Dynamik erforderlich sind, implementiert. Um die Simulation in Betrieb zu nehmen müssen noch einige Aufgaben erledigt werden.

- e) Verbinden Sie alle notwendigen Ein- und Ausgänge der erstellten Subsysteme und Funktionen, um eine erfolgreiche Simulation aller zeitabhängiger Variablen in Abhängigkeit der vier Drehzahlen zu ermöglichen. Sehen Sie außerdem Möglichkeiten vor, um die Ausgänge der beiden erstellten Subsysteme mittels *Sinks* zu visualisieren.
- f) Legen sie im MATLAB-Workspace alle erforderlichen Parameter und Startwerte an und speichern Sie diese in einer MATLAB-Datei *init.mat*. Setzen Sie alle erforderlichen Startwerte außer der Höhe z auf Null. Für die Höhe soll ein Wert von $z = 10$ m gesetzt werden.
- g) Abschließend überlegen Sie sich, welchen Schub Sie aufbringen müssen um den Quadrocopter in der durch die Startwerte voreingestellten Lage zu halten, d.h. g zu kompensieren. Berechnen Sie hieraus die erforderlichen Drehzahlen, und bringen Sie diese als Eingänge in das erstellte Simulations-Modell. Simulieren Sie die Quadrocopter-Dynamik über einen Zeitraum von $0 \text{ s} \leq t \leq 20 \text{ s}$.

Um die Prüfungsaufgabe abzugeben, speichern Sie das lauffähige (!) Simulink-Modell, sowie die für den Betrieb des Modells erforderlichen Parameter aus dem MATLAB-Workspace (*init.mat*) in einem .zip-Archiv ab. Benennen Sie dieses mit Ihrem Vor-, Nachnamen sowie Matrikelnummer und senden sie das Archiv bis spätestens **29.10.2019** um 23:59 Uhr an die E-Mailadresse **carsten.herzog@directbox.com**.