

Cube-Balancing: Der selbst balancierende Würfel

Florian Roth, Michael Meindl, Alexander Sarici, Alexander Schleicher

16. September 2016

Vorwort

In der folgenden Dokumentation wird das Cube-Balancing (CuBa)-Projekt vorgestellt, welches im Rahmen eines Entwicklungsprojektes an der Hochschule Karlsruhe durchgeführt wurde. Die Idee für dieses Projekt stammt von dem s.g. Cubli der ETH Zürich. Hierbei handelt es sich um einen Würfel, welcher in der Lage ist selbständig auf seine Ecken und Kanten zu springen und dort zu balancieren. Die Hauptaufgaben dieser Arbeiten bestehen darin einen solchen Würfel zu konstruieren, passende Sensorik bzw. Aktorik auszuwählen und in Betrieb zu nehmen und auf der Basis einer Systemanalyse einen Regelkreis zu entwerfen.

Inhaltsverzeichnis

1	Cube-Balance (CuBa) - Der selbstbalancierende Würfel	1
2	1D-Prototyp	2
2.1	Aktorik und Sensorik	3
2.2	Modellierung der Systemdynamik	4
2.3	Sensorik	7
2.3.1	Winkelschätzung	7
2.3.2	Kalibrierung und Justierung	7
2.3.2.1	Umrechnung der Beschleunigungswerte	8
2.3.2.2	Umrechnung der Winkelgeschwindigkeiten	8
2.3.3	Auswertung der Radgeschwindigkeit $\dot{\psi}$	9
2.3.4	Filterung der Sensordaten	10
2.3.4.1	Komplementärfilter	10
2.3.4.2	Kalman-Filter	10
2.3.4.3	Anwendung des Kalman-Filters	11
2.4	Modellbildung und Bestimmung der Systemgrößen	13
2.4.1	Identifikation der Parameter	13
2.4.1.1	Ermittlung des Reibwertes C_φ	13
2.4.1.2	Ermittlung des Reibwertes C_ψ	14
2.4.1.3	Resultate der Systemidentifikation	15
2.4.2	Entwurf des Simulink-Modelles	16
2.4.2.1	Simulation des Motors	16
2.4.2.2	Simulation der Würfelseite	16
2.4.2.3	Simulation der Schwungmasse	17
2.5	Reglerentwurf	18
2.5.1	Analytische Bestimmung der Reglerparameter	18
2.5.2	Verifizierung des Reglers an dem 1D-Prototyp	19
2.6	Aufspringen	21
2.7	Software	22
2.7.1	Hardware-Schnittstelle	22
2.7.2	TCP/IP-Verbindung	23
2.7.3	Komponentenarchitektur	23
2.7.4	Allgemeiner Ablauf der Versuchsapplikationen	24
3	3D-Modell	25
3.1	Modellierung der Systemdynamik	26
3.1.1	Potential des Systems	27
3.1.2	Kinetische Energie des Systems	27
3.1.3	Generalisierte Kraftkomponenten	28
3.1.3.1	Test	29

Abkürzungsverzeichnis

CuBa Cube-Balancing

BBB BeagleBone Black

SHM Shared Memory

1 Cube-Balance (CuBa) - Der selbstbalancierende Würfel

In der folgenden Dokumentation wird das CuBa-Projekt vorgestellt. Die Idee für dieses Projekt stammt von dem s.g. Cubli der ETH Zürich. Hierbei handelt es sich um einen Würfel, welcher in der Lage ist selbstständig auf seine Ecken und Kanten zu springen und dort zu balancieren. Hierfür werden Motoren in dem Würfel fixiert, an welchen Schwungräder befestigt sind. Die Motormomente dienen einerseits als Stellgröße um den Würfel zu balancieren. Andererseits können die Schwungmassen über Bremsen abrupt zum Stillstand gebracht werden. Dadurch wird der Drehimpuls der Räder auf den Würfel übertragen. Somit ist es möglich den Würfel aus einer beliebigen Ruhelage aufzurichten.

Das Projekt ist in zwei Abschnitte unterteilt. Zuerst wird eine einzelne Würfelseite konzipiert, an welcher ein Motor mit einer Schwungmasse angebracht ist. Über eine Achse wird die Würfelseite gelagert und ist somit auf einen einzelnen rotatorischen Freiheitsgrad beschränkt. Dieser Prototyp dient als erstes Versuchsobjekt (1D-Modell) um die Systemeigenschaften zu untersuchen und Rückschlüsse auf den Entwurf des kompletten Würfels (3D-Modell) zu ziehen. Im zweiten Teil wird der letztendliche Würfel entwickelt, welcher über drei Motoren und Schwungmassen verfügt.

Der Aufbau der Dokumentation ist an den Projektverlauf angelehnt, so wird zu Beginn der Aufbau des 1D-Modells näher erläutert. Hierbei werden zuerst der mechanische Aufbau und die elektrischen Komponenten diskutiert. Im Anschluss werden mit Hilfe des Lagrange-Formalismus die Bewegungsgleichungen hergeleitet. Damit kann eine Zustandsraumdarstellung des Systems gewonnen werden, welche wiederum zu dem Entwurf eines zeitdiskreten Zustandsreglers verwendet wird. Außerdem wird auf die Auswertung der Sensoren eingegangen um mit Hilfe von Filtern und Datenfusionen möglichst genaue Schätzwerte des aktuellen Zustandsvektors zu erhalten. Zuletzt wird das Aufspringen der Würfelseite näher untersucht. Hierbei wird ein Lernalgorithmus vorgestellt, welcher die optimale Radgeschwindigkeit zum Aufspringen unter realen Bedingungen findet.

Der zweite Abschnitt beschäftigt sich mit der Entwicklung des 3D-Modells, wobei vor allem auf die Unterschiede zu dem 1D-Modell eingegangen wird. Durch die zusätzlichen Freiheitsgrade ergibt sich eine deutlich komplexere Systemdynamik. Da die verschiedenen Rotationen nicht voneinander entkoppelt werden können, führt dies auch zu einem umfangreicheren Regelsystem.

Der letzte Teil des Berichtes stellt eine Bewertung und Fazit des Projektes dar. Wobei die verschiedenen Ansätze kritisch beurteilt werden und mögliche Verbesserungen aufgezeigt werden. Zusätzlich werden mögliche Erweiterungen und Optimierungen vorgestellt.

2 1D-Prototyp

In diesem Teil wird der erste Prototyp vorgestellt. Hierbei handelt es sich um eine einzelne Würfelseite, welche mit Hilfe einer Achse gelagert ist. Dadurch wird die Bewegung des Systems auf zwei rotatorische Freiheitsgrade beschränkt, nämlich die Rotation um die Achse und die Bewegung der Schwungmasse relativ zu der Würfelseite. Mit Hilfe dieses Entwurfes kann die Dynamik und Anforderungen an die Komponenten an einem vereinfachten Modell untersucht werden. Aus diesen Ergebnisse können dann Rückschlüsse auf den Entwurf des endgültigen Würfels gezogen werden. Hier ein vernünftiges Bild von dem endgültigen Aufbau?

2.1 Aktorik und Sensorik

Der folgenden Abschnitt beschreibt die verwendeten elektrischen Bauteile, um einerseits die benötigten physikalischen Größen zu messen, und andererseits die verwendete Aktorik, um das Aufspringen und Balancieren der Würfelseite zu ermöglichen.

Die Aufgabe der Sensorik besteht darin die Zustandsgrößen des Systemes zu bestimmen. Hierfür werden zwei *GYR-521*-Platinen verwendet, die mit einem *MPU6050*-IC der Firma *InvenSense* bestückt sind. Diese bieten jeweils einen dreiachsigen Beschleunigungssensor und Gyroskop. Mit Hilfe dieser Messwerte können die Zustandsgrößen φ und $\dot{\varphi}$ berechnet werden. Die Sensoren bieten die zusätzliche Möglichkeit einen variablen Tiefpassfilter zu verwenden um eine erste Glättung der Messwerte durchzuführen. Dieser Tiefpassfilter wird auf eine Grenzfrequenz von 44Hz eingestellt. Dieser Wert hat sich empirisch als optimaler Kompromiss zwischen Filterung der Rauschsignale und Verzögerung des eigentlichen Signals. Die Konfiguration und Auswertung der Sensoren erfolgt über eine *I²C*-Schnittstelle. Die Justierung und Auswertung der Sensoren wird näher in Abschnitt ?? beschrieben.

Abschnitt 2.2 zeigt den Einfluss eines Motormomentes auf die Position und Geschwindigkeit der Würfelseite. Um diese Moment zu erzeugen wird ein bürstenloser DC-Motor der Firma *MaxonMotor* verwendet (EC 45 flat, 50 Watt). Die Kriterien zur Auswahl des Motors sind einerseits die maximale Drehzahl und Drehmoment, andererseits die mechanische Zeitkonstante. Für das Aufspringen des Würfels ist die maximale Drehzahl des Motors von Bedeutung, die 10000 Umdrehung pro Minute des gewählten Motor reichen hierbei aus um eine ausreichend hohe kinetische Energie der Schwungmasse zu ermöglichen. Die Robustheit der Regelung wird durch das maximale Drehmoment limitiert, welches in diesem Fall bei 83.4 mNm liegt. Von besondere Bedeutung für die Regelung ist die mechanische Zeitkonstante des Motors, da diese eine Verzögerung der Stellgröße bewirkt und somit den geschlossenen Regelkreis negativ beeinflussen kann. Die mechanische Zeitkonstante des gewählten Motors ist mit 13.3ms im Vergleich zu anderen Kandidaten sehr niedrig. Die Ansteuerung des Motors erfolgt über den Treiberbaustein *ESCON 36/3 EC*, welcher ebenfalls von der Firma *Maxon Motor* vertrieben wird. Dieser ermöglicht die Steuerung des Drehmoments über ein PWM-Signal und die Auswertung der Winkelgeschwindigkeit $\dot{\psi}$ über ein analoges Signal.

Mit Hilfe einer mechanischen Bremse kann die Schwungmasse stoßartig zum Stillstand gebracht werden. Dadurch wird die kinetische Energie der Schwungmasse teilweise auf das Gesamtsystem übertragen und ermöglicht somit das Aufspringen. Die Bremsbacken werden über einen Servomotor betätigt, welcher mit Hilfe eines PWM-Signales kontrolliert wird.

Zur Ansteuerung der Aktorik und Sensorik wird ein BeagleBone Black verwendet, auf welchem eine Linux-Distribution ausgeführt wird. Die Programmierung erfolgt über eine, auf Eclipse basierende, Toolkette. Um die Auswertung der Sensordaten und den Entwurf der Regelung zu erleichtern, werden Sensoren- und Regelungsdaten an MATLAB übertragen, wo weitere Auswertungen stattfinden.

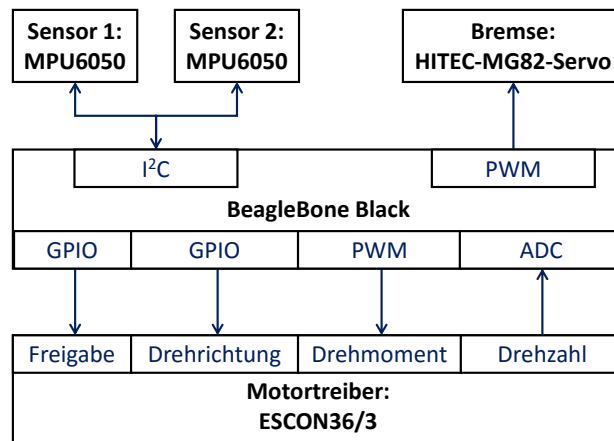


Abbildung 1: Übersicht Elektrik, Quelle: eigene Darstellung

2.2 Modellierung der Systemdynamik

In dem folgenden Abschnitt werden die Bewegungsgleichungen mit Hilfe des Lagrange Formalismus hergeleitet. Aus diesen Gleichung kann im Anschluss eine Zustandsraumdarstellung aufgestellt werden, welche als Grundlage für den Reglerentwurf dient.

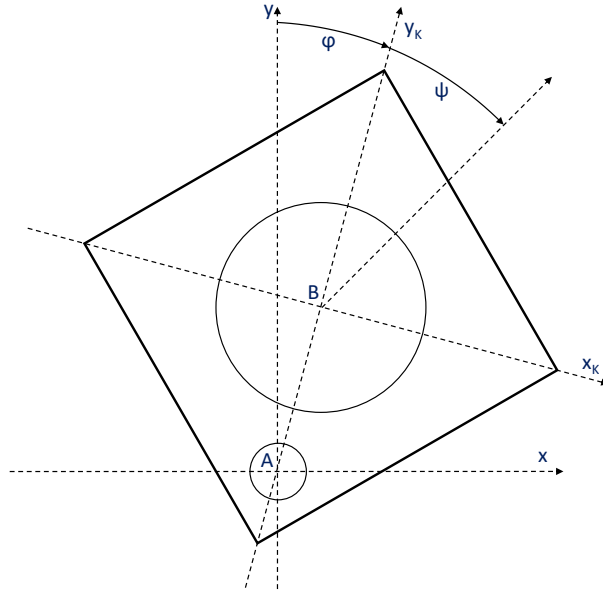


Abbildung 2: Mechanischer Aufbau, Quelle: eigene Darstellung

Der Prototyp besteht aus einem starren Körper der in A auf einer Achse gelagert ist. In B ist eine Schwungmasse über einen Motor mit dem Körper verbunden. Somit verfügt das Gesamtsystem über zwei Freiheitsgrade, welche durch die generalisierten Koordinaten

$$q_1 = \varphi \quad q_2 = \psi \quad (1)$$

beschrieben werden. Der Winkel φ wird von den Achsen y und y_K eingeschlossen. Der Winkel beschreibt die rotatorische Verschiebung der Schwungmasse zu dem Körper. Die folgenden Größen beschreiben die weiteren physikalischen Gegebenheiten des Systems.

Variable	Erklärung
$q_1 = \varphi$	Ausfallwinkel des Körpers
$q_2 = \psi$	Winkel zwischen Schwungmasse und Körper
A	Drehpunkt des Körpers
B	Drehpunkt des Schwungrades
l_{AB}	Abstand zwischen A und B
l_{AC}	Abstand zwischen A und dem Schwerpunkt des Körpers
m_K	Masse des Körpers
m_R	Masse des Schwungrades
θ_K^A	Massenträgheitsmoment des Körper um A
θ_R^B	Massenträgheitsmoment der Schwungmasse um B
C_φ	Dynamischer Reibkoeffizient des Körpers in A
C_ψ	Dynamischer Reibkoeffizient des Schwungrades in B
T_M	Drehmoment des Motor

Um die Bewegungsgleichungen des Systems zu ermitteln wird der Lagrange Formalismus verwendet. Dieser basiert auf der Lagrange-Funktion L , welche die Differenz der kinetischen Energie T und der potenziellen Energie V des Systems beschreibt.

$$T = \frac{1}{2}[(\theta_K^A + m_R \cdot l_{AB}^2)\dot{\varphi}^2 + \theta_R^B(\dot{\varphi} + \dot{\psi})^2] \quad (2)$$

$$V = g(m_R \cdot l_{AB} + m_K \cdot l_{AC})\cos(\varphi) \quad (3)$$

$$L = T - V = \frac{1}{2}[(\theta_K^A + m_R \cdot l_{AB}^2)\dot{\varphi}^2 + \theta_R^B(\dot{\varphi} + \dot{\psi})^2] - g(m_R \cdot l_{AB} + m_K \cdot l_{AC})\cos(\varphi) \quad (4)$$

In dem System wirken unterschiedliche Kräfte und Momente, welche nach dem Lagrange-Formalismus als generalisierte Kraftkomponenten Q_i dargestellt werden können. Das Ziel hiervon ist es, die Summe der Kräfte, bzw. deren verrichtete Arbeit, in Komponenten zu zerteilen, welche in Richtung der generalisierten Koordinaten wirken. Die generalisierten Kraftkomponenten können nach der folgenden Formel berechnet werden, wobei F_j die verschiedenen Kräfte bzw. Momente und r_j eine Verrückung in ihrer Wirkungsrichtung darstellt.

$$Q_i = \sum F_j \cdot \frac{\partial r_j}{\partial q_i} \quad (5)$$

Durch die Gravitation wirkt eine konstante Kraft entgegen der raumfesten y-Koordinate auf den Körper.

$$F_1 \cdot r_1 = -g \cdot (m_K \cdot y_K + m_R \cdot y_R) = -g \cdot (m_K \cdot l_{AC} + m_R \cdot l_{AB}) \cdot \cos(\varphi) \quad (6)$$

Durch den Motor und die Reibung in den beiden Lagern entstehen die Drehmomente T_M , M_{R_∞} und M_{R_ψ} , welche entlang der generalisierten Koordinaten wirken und somit nicht umgeformt werden müssen. Die Reibmomente werden als linear proportional zu der momentanen Winkelgeschwindigkeit modelliert.

$$F_2 \cdot r_2 = T_M \cdot \psi \quad (7)$$

$$F_3 \cdot r_3 = M_{R_\varphi} \cdot \varphi = (-C_\varphi \cdot \dot{\varphi}) \cdot \varphi \quad (8)$$

$$F_4 \cdot r_4 = M_{R_\psi} \cdot \psi = (-C_\psi \cdot \dot{\psi}) \cdot \psi \quad (9)$$

Somit können nun die Summe der Kräfte und Momente gebildet werden und daraufhin die generalisierten Kraftkomponenten bestimmt werden.

$$\begin{aligned} Q_\varphi &= -g \cdot (m_K \cdot l_{AC} + m_R \cdot l_{AB}) \cdot \frac{\partial \cos(\varphi)}{\partial \varphi} + T_M \cdot \frac{\partial \psi}{\partial \varphi} + C_\varphi \cdot \dot{\varphi} \cdot \frac{\partial \varphi}{\partial \varphi} + C_\psi \cdot \dot{\psi} \cdot \frac{\partial \psi}{\partial \varphi} \\ &= g \cdot (m_K \cdot l_{AC} + m_R \cdot l_{AB}) \cdot \sin(\varphi) - C_\varphi \cdot \dot{\varphi} \end{aligned} \quad (10)$$

$$\begin{aligned} Q_\psi &= -g \cdot (m_K \cdot l_{AC} + m_R \cdot l_{AB}) \cdot \frac{\partial \cos(\varphi)}{\partial \psi} + T_M \cdot \frac{\partial \psi}{\partial \psi} + C_\varphi \cdot \dot{\varphi} \cdot \frac{\partial \varphi}{\partial \psi} + C_\psi \cdot \dot{\psi} \cdot \frac{\partial \psi}{\partial \psi} \\ &= T_M - C_\psi \cdot \dot{\psi} \end{aligned} \quad (11)$$

Bei dem Prototyp handelt es sich um ein nicht konservatives System, da durch die Reibung mechanische Energie verloren geht und der Motor dem System mechanische Energie zuführt. Da die beiden generalisierten Koordinaten φ und ψ voneinander unabhängig sind können aus dem d'Alembert'schen Prinzip zwei Bewegungsgleichungen abgeleitet werden.

$$\frac{d}{dt} \frac{\partial T}{\partial \dot{q}_i} - \frac{\partial T}{\partial q_i} = Q_i \quad (12)$$

$$\frac{d}{dt} \frac{\partial T}{\partial \dot{\varphi}} - \frac{\partial T}{\partial \varphi} = Q_\varphi \quad (13)$$

$$(\theta_K^A + \theta_R^B + m_R \cdot l_{AB}^2)\ddot{\varphi} + \theta_R^B \cdot \ddot{\psi} - g(m_R \cdot l_{AB} + m_K \cdot l_{AC})\sin(\varphi) + C_\varphi \cdot \dot{\varphi} = 0 \quad (14)$$

$$\frac{d}{dt} \frac{\partial T}{\partial \dot{\psi}} - \frac{\partial T}{\partial \psi} = Q_\psi \quad (15)$$

$$\theta_R^B \cdot \ddot{\psi} = T_M - C_\psi \cdot \dot{\psi} - \theta_R^B \cdot \ddot{\varphi} \quad (16)$$

Durch Einsetzen von (16) in (14) ergibt sich die folgende Bewegungsgleichung für die Würfelseite.

$$\ddot{\varphi} = \frac{g(m_R \cdot l_{AB} + m_K \cdot l_{AC})\sin(\varphi) - C_\varphi \cdot \dot{\varphi} + C_\psi \cdot \dot{\psi} - T_M}{\theta_K^A + m_R \cdot l_{AB}^2} \quad (17)$$

Die Bewegungsgleichung für die Schwungmasse ergibt sich durch Einsetzen von (17) in (16).

$$\ddot{\psi} = \frac{(\theta_K^A + m_R \cdot l_{AB}^2 + \theta_R^B)(T_M - C_\psi \cdot \dot{\psi})}{(\theta_K^A + m_R \cdot l_{AB}^2)\theta_R^B} + \frac{C_\varphi \cdot \dot{\varphi} - g(m_R \cdot l_{AB} + m_K \cdot l_{AC})\sin(\varphi)}{\theta_K^A + m_R \cdot l_{AB}^2} \quad (18)$$

2.3 Sensorik

Die Aufgabe der verwendeten Sensorik liegt darin die Werte für φ , und $\dot{\varphi}$ zu bestimmen. Hierfür wurden zwei MPU6050 IC's verwendet. Diese verfügen jeweils über einen Beschleunigungssensor und Gyroskop, welche Werte für drei Achsen ausgeben. Der Tiefpass der Sensoren wird auf eine Grenzfrequenz von 44Hz eingestellt, da hier einerseits eine erste Glättung der Daten erfolgt, andererseits aber keine zu große Verzögerung ergibt, welche sich wiederum negativ auf die Regelung auswirken könnte. Die Position und Ausrichtung der Sensoren ist in 3 dargestellt.

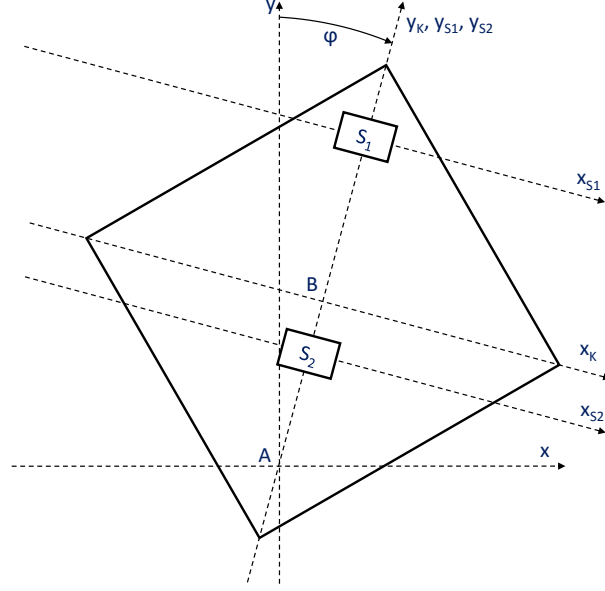


Abbildung 3: Position der Sensoren, Quelle: eigene Darstellung

2.3.1 Winkelschätzung

Die Sensoren keine Wege bzw. Winkel. Somit muss der Winkel φ berechnet werden. Die gemessenen Sensorwerte hängen von r_{S1} bzw. r_{S2} ab, welche den Abstand zwischen den Sensoren und dem Drehpunkt A beschreiben. Zusätzlich beeinflussen neben dem Winkel φ auch dessen beiden Ableitungen $\dot{\varphi}$ und $\ddot{\varphi}$ die Sensorausgabe. Allerdings lassen sich aus den Beschleunigungswerten der beiden Sensoren nach [1] wie folgt der aktuelle Wert von φ berechnen.

$$\ddot{S}_i = \begin{pmatrix} \ddot{x}_i \\ \ddot{y}_i \\ \ddot{z}_i \end{pmatrix} = \begin{pmatrix} r_{Si} \cdot \ddot{\varphi} + \sin(\varphi) \cdot g \\ -r_{Si} \cdot \dot{\varphi}^2 - \cos(\varphi) \cdot g \\ 0 \end{pmatrix} \quad i \in [1; 2] \quad (19)$$

$$\alpha = \frac{r_{S1}}{r_{S2}} \quad (20)$$

$$\ddot{x}_1 - \alpha \cdot \ddot{x}_2 = g(1 - \alpha)\sin(\varphi) \quad (21)$$

$$\ddot{y}_1 - \alpha \cdot \ddot{y}_2 = -g(1 - \alpha)\cos(\varphi) \quad (22)$$

$$\frac{\ddot{x}_1 - \alpha \cdot \ddot{x}_2}{\ddot{y}_1 - \alpha \cdot \ddot{y}_2} = -\tan(\varphi) \quad (23)$$

2.3.2 Kalibrierung und Justierung

Die Sensoren geben die Beschleunigungs- und Geschwindigkeitswerte als 16 Bit Werte im Zweierkomplement aus. Diese Rohwerte müssen in die mit Hilfe eines Ausgleichspolynoms in die jeweilige SI-Einheit umgerechnet werden.

2.3.2.1 Umrechnung der Beschleunigungswerte

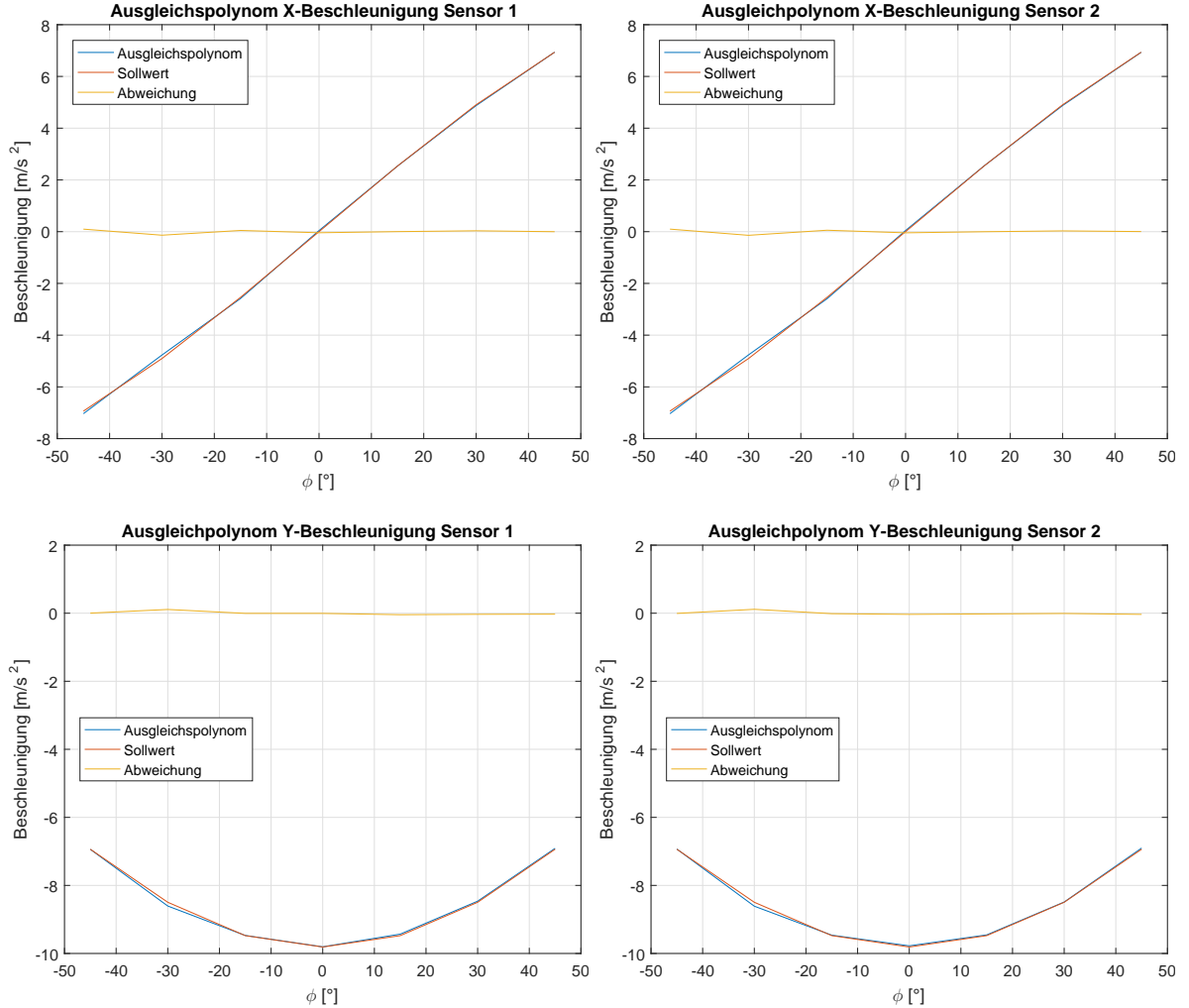
Um das Polynom zur Umrechnung der Beschleunigungswerte zu ermitteln werden sieben Messungen in den fixen Ausfallpositionen $\phi \in [-45, -30, -15, 0, 15, 30, 45]$ durchgeführt. Pro Position werden $m = 10000$ Messwerte aufgenommen. Da in der Ruhelage die Beschleunigung lediglich von dem aktuellen Ausfallwinkel abhängt ist der Sollwert für jede Position bekannt. Somit kann ein Polynom erster Ordnung approximiert werden um Mittelwerte der sieben Positionen in die entsprechenden Beschleunigungswerte umzurechnen.

$$\begin{aligned} \ddot{x}_n &\equiv \text{X-Beschleunigung Sensor } n & \ddot{x}_n^R &\equiv \text{X-Rohwert Sensor } n \\ \ddot{y}_n &\equiv \text{Y-Beschleunigung Sensor } n & \ddot{y}_n^R &\equiv \text{Y-Rohwert Sensor } n \end{aligned}$$

$$\ddot{x}_n = p_{x_n}^1 \cdot \ddot{x}_n^R + p_{x_n}^2 \quad | \quad n \in \{1, 2\} \quad (24)$$

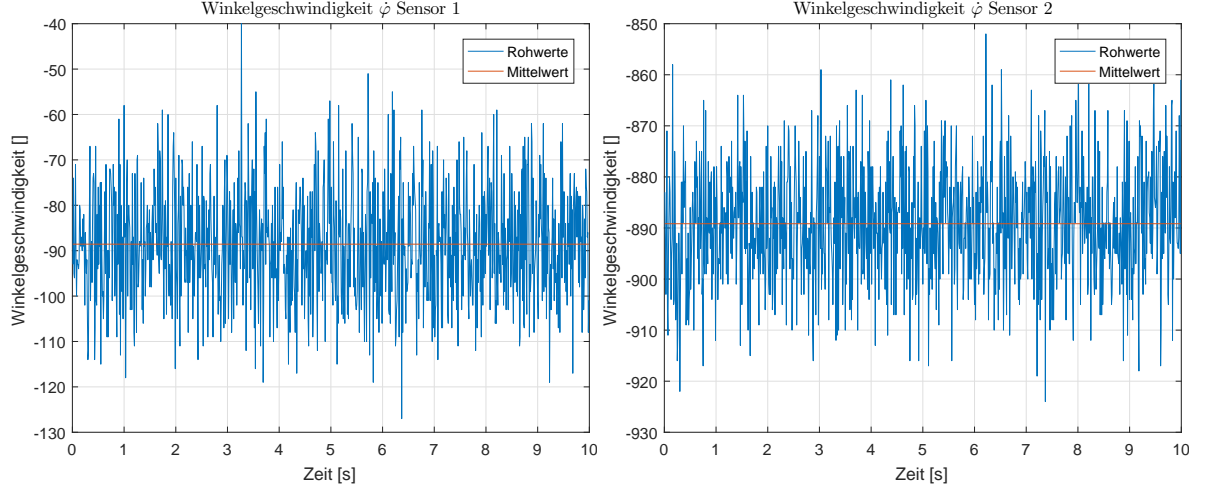
$$\ddot{y}_n = p_{y_n}^1 \cdot \ddot{y}_n^R + p_{y_n}^2 \quad | \quad n \in \{1, 2\} \quad (25)$$

$$\begin{aligned} p_{x_1}^1 &= -6.082 \cdot 10^{-4} & p_{x_1}^2 &= 0.4412 \\ p_{x_2}^1 &= -6.07 \cdot 10^{-4} & p_{x_2}^2 &= 0.2804 \\ p_{y_1}^1 &= -6.064 \cdot 10^{-4} & p_{y_1}^2 &= 0.1248 \\ p_{y_2}^1 &= -6.089 \cdot 10^{-4} & p_{y_2}^2 &= 0.09182 \end{aligned}$$



2.3.2.2 Umrechnung der Winkelgeschwindigkeiten

Um die Rohwerte der Gyroskope in Winkelgeschwindigkeiten umzurechnen wird die Würfelseite fixiert und die Winkelgeschwindigkeitswerte der beiden Sensoren aufgenommen. Hierbei werden jeweils



$$\dot{\varphi}_n \equiv \varphi\text{-Geschwindigkeit Sensor } n \quad \dot{\varphi}_n^R \equiv \dot{\varphi}\text{-Rohwert Sensor } n$$

$m = 1000$ Werte aufgenommen. Da der Sollwert $\dot{\varphi} = 0 \frac{m}{s}$ bekannt ist kann die systematische Messabweichung der Sensoren über den Mittelwert bestimmt werden. Der proportionale Umrechnungsfaktor von Rohdaten zu Winkelgeschwindigkeiten wird dem Datenblatt des Herstellers entnommen.

$$\dot{\varphi}_n = p_{\dot{\varphi}_n^R}^1 \cdot (\dot{\varphi}_n^R + p_{\dot{\varphi}_n^R}^2) \quad (26)$$

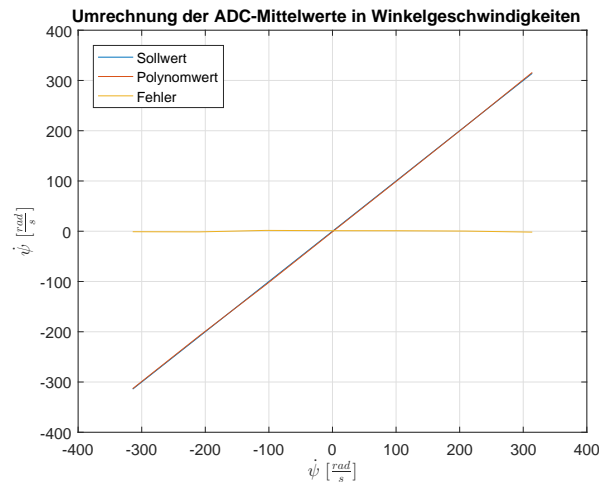
$$\begin{aligned} p_{\dot{\varphi}_1}^1 &= -0.0076 & p_{\dot{\varphi}_1}^2 &= 89 \\ p_{\dot{\varphi}_2}^1 &= -0.0076 & p_{\dot{\varphi}_2}^2 &= 889 \end{aligned}$$

2.3.3 Auswertung der Radgeschwindigkeit $\dot{\psi}$

Der Motortreiber liefert ein analoges Spannungssignal, welches die aktuelle Motorgeschwindigkeit wiedergibt. Um die ADC-Werte in SI-Einheiten umzurechnen wird ein Polynom erster Ordnung benötigt. Hierfür werden mit Hilfe der ESCON-Studio konstante Motorgeschwindigkeiten ($\dot{\psi} \in \{-3000, -2000, -1000, 0, 1000, 2000, 3000\} [rpm]$) gefahren und pro Durchlauf $m = 500$ ADC-Werte aufgenommen. Über die Mittelwerte der Messungen und die vorgegebenen Radgeschwindigkeiten wird anschließend ein Polynom erster Ordnung approximiert.

$$\dot{\psi} \equiv \text{Geschwindigkeit der Schwungmasse} \quad \dot{\psi}_{ADC} \equiv \text{ADC-Wert}$$

$$\dot{\psi} = -0.5176 \cdot \dot{\psi}_{ADC} + 1017 \quad (27)$$



2.3.4 Filterung der Sensordaten

In der Regel werden Sensoren von Störungen unterschiedlichster Art beeinflusst. In diesem Abschnitt werden das Komplementär- und Kalman-Filter vorgestellt. Hierbei handelt es sich um Methoden der Datenfusion um möglichst präzise Schätzungen der Zustandsgrößen zu erreichen.

Der Ausfallwinkel φ kann über die Auswertung der Beschleunigungswerte berechnet werden. Allerdings weisen diese Sensoren hochfrequente Störanteile auf, welche sich negativ auf die Winkelschätzung auswirken. Alternativ kann die Zustandsgröße φ über die Integration der Winkelgeschwindigkeit $\dot{\varphi}$ gewonnen werden. Die Geschwindigkeit $\dot{\varphi}$ wird mit Hilfe der beiden Gyroskope gemessen. Allerdings sind diese Messungen von einer systematischen Messabweichung betroffen, welche in die Integration einfließt und somit zu einem langfristigen Drift des Winkels φ führt.

2.3.4.1 Komplementärfilter

Eine der simpelsten Methoden der Signalverarbeitung stellt das s.g. Komplementärfilter dar. Die Grundidee dieses Prinzips besteht darin die einzelnen Sensorsignale mit einem Hoch- oder Tiefpass zu filtern und anschließend zu fusionieren.

In diesem Anwendungsfall wird der Winkel φ_{Acc} , welcher über die Beschleunigungswerte berechnet wird, mit einem Tiefpass gefiltert. Dadurch werden die hochfrequenten Störsignale entfernt. Der zweite Winkelwert φ_{Gyr} , welcher durch die Integration der Winkelgeschwindigkeit $\dot{\varphi}$ gewonnen wird, kann mit Hilfe eines Hochpasses von dem niederfrequenten Störsignal befreit werden.

Dieses Fusionsprinzip führt zu den folgenden Berechnungsschritten.

1. Integration der Winkelgeschwindigkeit $\dot{\varphi}_n$ nach der Trapezregel über das jeweilige Abtastintervall t_a .

$$\Delta\varphi_n = \frac{t_a}{2} \cdot (\dot{\varphi}_n + \dot{\varphi}_{n-1}) \quad (28)$$

2. Summation des vorherigen Winkels φ_{n-1} mit der Winkeländerung $\Delta\varphi_n$.

$$\varphi_{Gyr,n} = \varphi_{n-1} + \Delta\varphi_{Int,n} \quad (29)$$

3. Berechnung des aktuellen Winkels φ_n mit einem gewählten Gewichtungsfaktor α und den Winkelschätzungen $\varphi_{Gyr,n}$ und $\varphi_{Acc,n}$.

$$\varphi_n = \alpha \cdot \varphi_{Gyr,n} + (1 - \alpha) \cdot \varphi_{Acc,n} \quad (30)$$

2.3.4.2 Kalman-Filter

Das Kalman-Filter beruht auf dem Prinzip der Zustandsschätzung. Diese Schätzung erfolgt mittels der Messwerte aus Gyroskopen, Beschleunigungssensoren und der darauf folgenden Winkelschätzung. Hierbei greift der Algorithmus auf Methoden der Wahrscheinlichkeitsrechnung zurück. Ist die Varianz eines Messfehlers gegeben, so kann eine Schätzung des tatsächlichen Zustandes vorgenommen werden. In diesem Fall wird mit den linearisierten Bewegungsgleichungen gearbeitet. Dadurch kann das gewöhnliche Kalman-Filter verwendet werden. Für nichtlineare Systeme müssen unterschiedliche Filteralgorithmen, wie beispielsweise das Extended-Kalman-Filter, verwendet werden.

Bezeichnung	Erklärung
\mathbf{x}_n^*	Systemzustände
$\hat{\mathbf{x}}_n$	Geschätzte Systemzustände
\mathbf{u}_n	Messbare Eingangsgrößen
\mathbf{v}_n	Störgrößen
\mathbf{A}_d	Systemmatrix, welche den Systemzustand $\hat{\mathbf{x}}_n$ auf den folgenden Zeitschritt abbildet.
\mathbf{B}_d	Eingangsmatrix
\mathbf{C}_d	Ausgangsmatrix
\mathbf{P}_n^*	Kovarianzmatrix von \mathbf{x}_n^* . Legt die Sicherheit der Schätzung fest.
$\hat{\mathbf{P}}_n$	Filterkovarianzmatrix von $\hat{\mathbf{x}}_n$. Gibt die Gewichtung der Messwerte an.
\mathbf{Q}_n	Kovarianzmatrix des Systemrauschens
\mathbf{R}_n	Kovarianzmatrix des Messrauschens

Die Berechnung des Kalman-Filters verluft in zwei Schritten, welche in insgesamt funf Teile gliedert ist. Zuerst wird der Predikationsschritt durchgefuhrt, hierbei wird der Predikationsschatzwert \mathbf{x}_{n+1}^* berechnet, welcher den Systemzustand im folgenden Zeitschritt darstellt. Zusatzlich wird die Predikationskovarianzmatrix \mathbf{P}_{n+1}^* berechnet, welche angibt wie sicher die Vorhersage im Verhaltnis zu dem wahren Systemzustand ist. Im zweiten Abschnitt, der s.g. Korrektur- bzw. Filterschritt, wird die Vorhersage mit Hilfe der neuen Messwerte korrigiert. Hierfur wird zuerst die Verstarkungsmatrix \mathbf{K}_{n+1} berechnet, welche den Ruckkopplungsfaktor der Messwerte wiedergibt. Im Anschluss wird der Endschatzwertes des Systemzustandes mit Hilfe des ersten Predikationswertes \mathbf{x}_{n+1}^* und der Verstarkungsmatrix \mathbf{K}_{n+1} bestimmt. Zuletzt wird die Varianz des geschatzten Systemzustandes $\hat{\mathbf{P}}_{n+1}$ berechnet.

1. Predikationsschatzwert

Der Systemzustand zum Zeitpunkt n ist beschreibbar durch die zeitdiskrete, lineare, stochastische Differenzengleichung

$$\mathbf{x}_{n+1}^* = \mathbf{A}_d \cdot \hat{\mathbf{x}}_n + \mathbf{B} \cdot \mathbf{u}_n + \mathbf{v}_n \quad (31)$$

Die Systemmatrix \mathbf{A} bildet den Systemzustand $\hat{\mathbf{x}}$ von Zeitschritt n auf den folgenden Zeitschritt $n + 1$ ab. Die Eingangsgroen \mathbf{u}_n werden durch die Matrix \mathbf{B} auf den Systemzustand \mathbf{x}_{n+1}^* abgebildet. Das Raschen bzw. die aueren Storeinflusse werden durch den additiven Term \mathbf{v}_n representiert.

2. Predikationskovarianzmatrix

Die Kovarianzmatrix \mathbf{P}_{n+1}^* gibt an, wie sicher die Predikation im Verhaltnis zu dem wahren Systemzustand ist.

$$\mathbf{P}_{n+1}^* = \mathbf{A} \cdot \hat{\mathbf{P}}_n \cdot \mathbf{A}^T + \mathbf{Q}_n \quad (32)$$

3. Verstarkungsmatrix

Die Verstarkungsmatrix \mathbf{K}_{n+1} wird fur die Korrektur der Vorhersage verwendet. Sie bestimmt, mit welcher Verstarkung der Messvektor ruckgekoppelt wird.

$$\mathbf{K}_{n+1} = (\mathbf{P}_{n+1}^* \cdot \mathbf{C}^T)(\mathbf{C} \cdot \mathbf{P}_{n+1}^* \cdot \mathbf{C}^T + \mathbf{R}_{n+1})^{-1} \quad (33)$$

4. Filterschatzwert

Mit Hilfe der Verstarkungsmatrix \mathbf{K}_{n+1} und des Predikationswertes \mathbf{x}_{n+1}^* wird der finale Schatzwert des Systemzustandes zu dem Zeitpunkt $n + 1$ bestimmt.

$$\hat{\mathbf{x}}_{n+1} = \mathbf{x}_{n+1}^* + \mathbf{K}_{n+1}(\mathbf{y}_{n+1} - \mathbf{C} \cdot \mathbf{x}_{n+1}^*) \quad (34)$$

5. Filterkovarianzmatrix

Im letzten Schritt wird die Varianz des geschatzten Systemzustandes berechnet.

$$\hat{\mathbf{P}}_{n+1} = \mathbf{P}_{n+1}^* - \mathbf{K}_{n+1} \cdot \mathbf{C} \cdot \mathbf{P}_{n+1}^* \quad (35)$$

2.3.4.3 Anwendung des Kalman-Filters

Um die zuvor angesprochenen Nachteile der einzelnen Sensoren auszugleichen, wird nun die Umsetzung eines Kalman-Filters zur Sensorfusion vorgestellt.

Das Kalman-Filter fusioniert den Ausfallwinkel φ_{Acc} aus der Winkelschatzung mit der Winkelanderung $\Delta\varphi_{Gyr}$, welche durch Integration der Gyroskopwerte gewonnen wurde. Daraus ergibt sich der gefilterte Winkel $\hat{\varphi}_n$. Dieser gefilterte Zustandswert wird fur die Berechnung des Regelkreises verwendet.

Als absolute Eingangsgroen stehen hier zum einen die Winkelschatzung φ_{Acc} und die Winkelanderung $\Delta\varphi_{Gyr}$ zur Verfugung. Somit kann das fehlerbehaftete Teilsystem durch folgende Gleichungen beschrieben werden, wobei $v_{Gyr,n}$ $w_{Acc,n}$ die storbehafteten Additionsterme darstellen.

$$\varphi_{n+1} = \varphi_n + \Delta\varphi_{Gyr,n} + v_{Gyr,n} \quad (36)$$

$$\varphi_{Acc,n} = \varphi_n + w_{Acc,n} \quad (37)$$

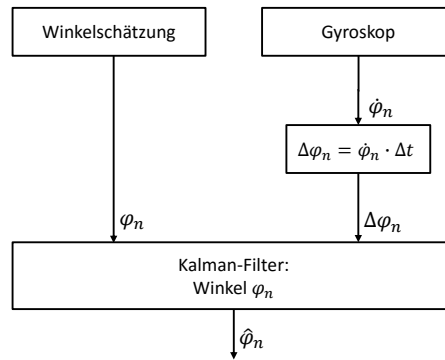


Abbildung 4: Übersicht Kalman-Filter, Quelle: eigene Darstellung

Die bereits vorgestellten Berechnungsschritte können nun auf den spezifischen Anwendungsfall angepasst werden. Da lediglich eine einzelne Zustandsgröße berechnet wird, werden anstelle von Vektoren und Matrizen skalare Größen verwendet.

1. Prädikationsschätzwert

$$\varphi_{n+1}^* = \hat{\varphi}_n + \Delta\varphi_{Gyr,n} \quad (38)$$

2. Prädikationsvarianz

$$P_{n+1}^* = \hat{P}_{n+1} + \sigma^2(\Delta\varphi_{Gyr,n}) \quad (39)$$

3. Verstärkungsfaktor

$$K_{n+1} = \frac{P_{n+1}^*}{P_{n+1}^* + \sigma^2(\varphi_{Acc,n})} \quad (40)$$

4. Filterschätzwert

$$\hat{\varphi}_{n+1} = \varphi_{n+1}^* + K_{n+1} \cdot (\varphi_{Acc,n} - \varphi_{n+1}^*) \quad (41)$$

5. Filtervarianz

$$\hat{P}_{n+1} = (1 - K_{n+1}) \cdot P_{n+1}^* \quad (42)$$

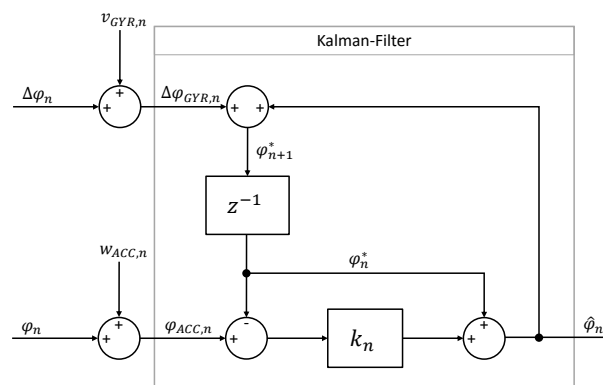


Abbildung 5: Berechnungsablauf Kalman-Filter, Quelle: eigene Darstellung

2.4 Modellbildung und Bestimmung der Systemgrößen

Mit Hilfe der Bewegungsgleichungen aus Abschnitt 2.2 kann nun eine Zustandsraumdarstellung aufgestellt werden. Hierfür werden die nichtlinearen Terme entsprechend linearisiert. Mit Hilfe der Bewegungsgleichungen bzw. Zustandsraumdarstellung kann ein Simulink-Modell implementiert werden um das Systemverhalten zu simulieren. Mit Hilfe der Zustandsraumdarstellung wird ein Zustandsregler entworfen, welcher an dem Modell erprobt werden kann. Zusätzlich über die Simulation der Einfluss der einzelnen Parameter, Sensorrauschen und Störungen untersucht werden.

$$\mathbf{x} = \begin{pmatrix} \varphi \\ \dot{\varphi} \\ \psi \end{pmatrix} \quad \mathbf{y} = \begin{pmatrix} \varphi \\ \dot{\varphi} \\ \dot{\psi} \end{pmatrix} \quad u = T_M \quad (43)$$

$$\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x} + \mathbf{B} \cdot u \quad (44)$$

$$\mathbf{y} = \mathbf{C} \cdot \mathbf{x} + \mathbf{D} \cdot u \quad (45)$$

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 \\ \frac{g(m_K \cdot l_{AC} + m_R \cdot l_{AB})}{\theta_K^A + m_R \cdot l_{AB}^2} & \frac{-C_\varphi}{\theta_K^A + m_R \cdot l_{AB}^2} & \frac{C_\psi}{\theta_K^A + m_R \cdot l_{AB}^2} \\ \frac{-g(m_K \cdot l_{AC} + m_R \cdot l_{AB})}{\theta_K^A + m_R \cdot l_{AB}^2} & \frac{C_\varphi}{\theta_K^A + m_R \cdot l_{AB}^2} & \frac{-C_\psi(\theta_K^A + \theta_R^B + m_R \cdot l_{AB}^2)}{\theta_R^B(\theta_K^A + m_R \cdot l_{AB}^2)} \end{pmatrix} \quad (46)$$

$$\mathbf{B} = \begin{pmatrix} 0 \\ \frac{-1}{\theta_K^A + m_R \cdot l_{AB}^2} \\ \frac{\theta_K^A + \theta_R^B + m_R \cdot l_{AB}^2}{\theta_R^B(\theta_K^A + m_R \cdot l_{AB}^2)} \end{pmatrix} \quad \mathbf{C} = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix} \quad \mathbf{D} = \begin{pmatrix} 0 \end{pmatrix}$$

2.4.1 Identifikation der Parameter

Der Reglerentwurf und die Simulation erfordern eine möglichst präzise Bestimmung der Systemparameter, wie z.B. Längen, Massen, Massenträgheitsmomente und Reibwerte. Die Bestimmung der Längen l_{AB} und l_{AC} , der Massen m_K , m_R und m_G , der Massenträgheitsmomente θ_K^A und θ_R^B erfolgt über das CAD-Modell. Hierfür werden Bauteile mit einer nicht homogenen Massenverteilung, wie z.B. die Motoren, in separate Baugruppen mit homogener Massenverteilung unterteilt.

2.4.1.1 Ermittlung des Reibwertes C_φ

In dem die Schwungmasse fest mit der Würfelseite verbunden wird ergibt sich die folgende Bewegungsgleichung für das Gesamtsystem.

$$(\theta_K^A + \theta_R^B + m_R \cdot l_{AB}^2) \ddot{\varphi} = g(m_K \cdot l_{AC} + m_R \cdot l_{AB}) \sin(\varphi) - C_\varphi \cdot \dot{\varphi} \quad (47)$$

In dem Versuchsaufbau wird das Gesamtsystem nun von einem Startwinkel φ_0 losgelassen, woraufhin eine gedämpfte Schwingung entsteht. Mit Hilfe der Sensoren können die Größen φ , $\dot{\varphi}$ und $\ddot{\varphi}$ gemessen werden.

Über die n Messpunkte ergeben sich die folgenden Vektoren.

$$\boldsymbol{\varphi} = \begin{pmatrix} \varphi_1 \\ \varphi_2 \\ \vdots \\ \varphi_n \end{pmatrix} \quad \dot{\boldsymbol{\varphi}} = \begin{pmatrix} \dot{\varphi}_1 \\ \dot{\varphi}_2 \\ \vdots \\ \dot{\varphi}_n \end{pmatrix} \quad \ddot{\boldsymbol{\varphi}} = \begin{pmatrix} \ddot{\varphi}_1 \\ \ddot{\varphi}_2 \\ \vdots \\ \ddot{\varphi}_n \end{pmatrix} \quad (48)$$

Damit ergibt sich durch Umstellen von 47 die folgende Gleichung.

$$C_\varphi \cdot \dot{\boldsymbol{\varphi}} = g(m_K \cdot l_{AC} + m_R \cdot l_{AB}) \sin(\boldsymbol{\varphi}) - (\theta_K^A + \theta_R^B + m_R \cdot l_{AB}^2) \ddot{\boldsymbol{\varphi}} \quad (49)$$

Mit Hilfe der Methode der kleinsten Fehlerquadrate kann nun der Reibwert C_φ bestimmt werden.

$$C_\varphi = 6.2 \cdot 10^{-3} \cdot \text{kg} \cdot \text{m}^2 \cdot \text{s}^{-1} \quad (50)$$

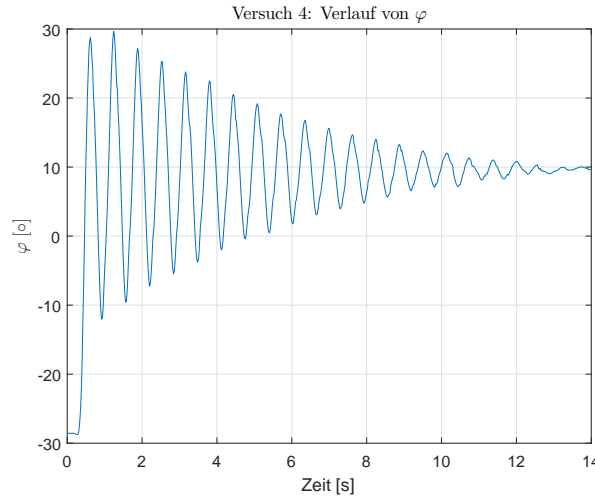


Abbildung 6: Ausfallwinkel der Würfelseite bei Versuch 4, Quelle: eigene Darstellung

2.4.1.2 Ermittlung des Reibwertes C_ψ

Im nächsten Versuchsaufbau wird die Würfelseite fixiert ($\dot{\varphi} = 0$). Hierbei beschleunigt der Motor die Schwungmasse mit einem konstanten Drehmoment $T_M = 10 \text{ mNm}$. T_M ist so zu wählen, dass sich die Radgeschwindigkeit $\dot{\psi}$ in einem Bereich bewegt, welcher dem Arbeitsbereich des geschlossenen Regelkreises entspricht.

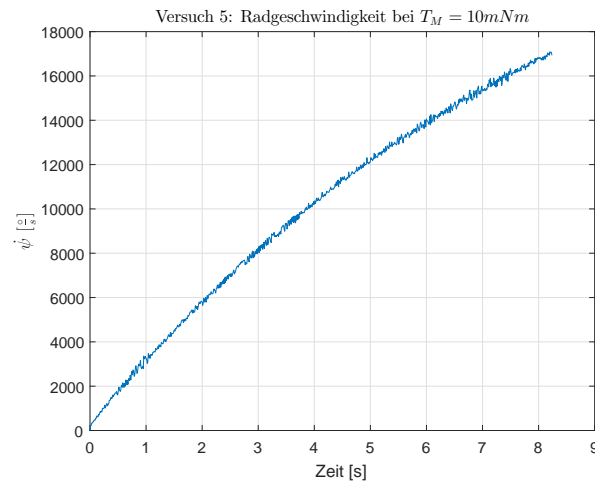


Abbildung 7: Versuch 5: Verlauf der Radgeschwindigkeit, Quelle: eigene Darstellung

Da die Bewegung auf einen Freiheitsgrad beschränkt wurde vereinfacht sich das Modell des Systems auf die folgende Bewegungsgleichung.

$$\theta_R^B \cdot \ddot{\psi} = T_M - C_\psi \cdot \dot{\psi} \quad (51)$$

Im Versuchsverlauf werden bei n Stützstellen die Werte von ψ , $\dot{\psi}$ und $\ddot{\psi}$ gemessen. Daraus ergeben sich die folgenden Vektoren.

$$\psi = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \vdots \\ \psi_n \end{pmatrix} \quad \dot{\psi} = \begin{pmatrix} \dot{\psi}_1 \\ \dot{\psi}_2 \\ \vdots \\ \dot{\psi}_n \end{pmatrix} \quad \ddot{\psi} = \begin{pmatrix} \ddot{\psi}_1 \\ \ddot{\psi}_2 \\ \vdots \\ \ddot{\psi}_n \end{pmatrix} \quad (52)$$

Durch Einsetzen von 52 in 51 kann über die Methode der kleinsten Fehlerquadrate wiederum der Reibwert C_ψ bestimmt werden.

$$C_\psi = 3.1176 \cdot 10^{-5} \cdot \text{kg} \cdot \text{m}^2 \cdot \text{s}^{-1} \quad (53)$$

2.4.1.3 Resultate der Systemidentifikation

An Hand der beschriebenen Versuche und Methoden wurden die folgenden Werte für die Parameter des Gesamtsystems ermittelt.

Parameter	Wert
l_{AB}	$0.084m$
l_{AC}	$0.087m$
m_K	$0.221kg$
m_R	$0.09kg$
θ_K^A	$2.8 \cdot 10^{-3} kg \cdot m^2$
θ_R^B	$1.1683 \cdot 10^{-4} \cdot kg \cdot m^2$
C_φ	$6.2 \cdot 10^{-3} \cdot kg \cdot m^2 \cdot s^{-1}$
C_ψ	$3.1176 \cdot 10^{-5} \cdot kg \cdot m^2 \cdot s^{-1}$
r_{S1}	$0.14m$
r_{S2}	$0.061m$

2.4.2 Entwurf des Simulink-Modelles

Dieser Abschnitt erklärt den Aufbau des Simulink-Modelles zur Simulation des Systems. Die oberste Modellschicht besteht aus drei Subsystemen zur Simulation des Motor, der Würfelseite und der Schwungmasse.

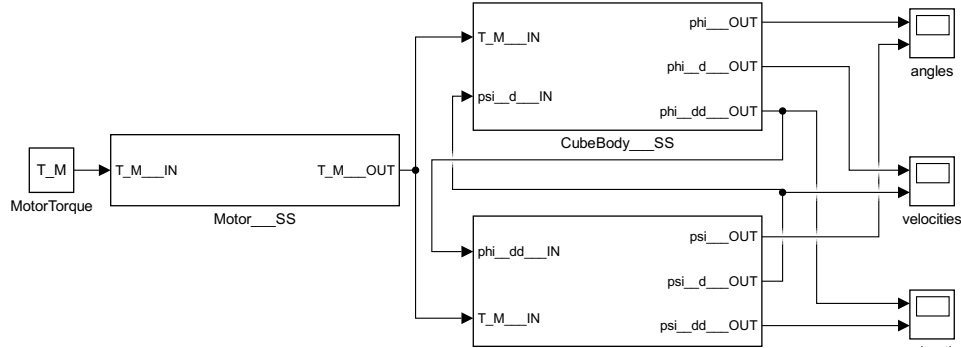


Abbildung 8: Simulink-Modell Übersicht, Quelle: eigene Darstellung

2.4.2.1 Simulation des Motors

Der Motor wird als zwei in Reihe geschaltete PT1-Glieder simuliert. Da der Regler als Stellgröße ein Motormoment berechnet, beträgt die Verstärkung des Motor K_M in der Simulation den Wert eins. Die Zeitkonstanten der PT1-Glieder sind einerseits die elektrische Zeitkonstante T_e und die mechanische Zeitkonstante T_m , wessen Werte dem Datenblatt des Herstellers entnommen werden.

$$K_M = 1 \quad T_e = 0.55ms \quad T_m = 12.4ms \quad (54)$$

2.4.2.2 Simulation der Würfelseite

Die Dynamik der Würfelseite wird von 17 beschrieben.

$$\ddot{\varphi} = \frac{g(m_R \cdot l_{AB}^2 + m_K \cdot l_{AC}) \sin(\varphi) - C_\varphi \cdot \dot{\varphi} + C_\psi \cdot \dot{\psi} - T_M}{\theta_K^A + m_R \cdot l_{AB}^2} \quad (17)$$

Somit ist die Winkelbeschleunigung gleich der Summe der Drehmomente geteilt durch die betroffenen Massenträgheitsmomente. Durch Integration und Rückführung können die einzelnen Drehmomente berechnet werden. Das folgende Modell zeigt die Umsetzung dieser Berechnungsvorschrift in Simulink.

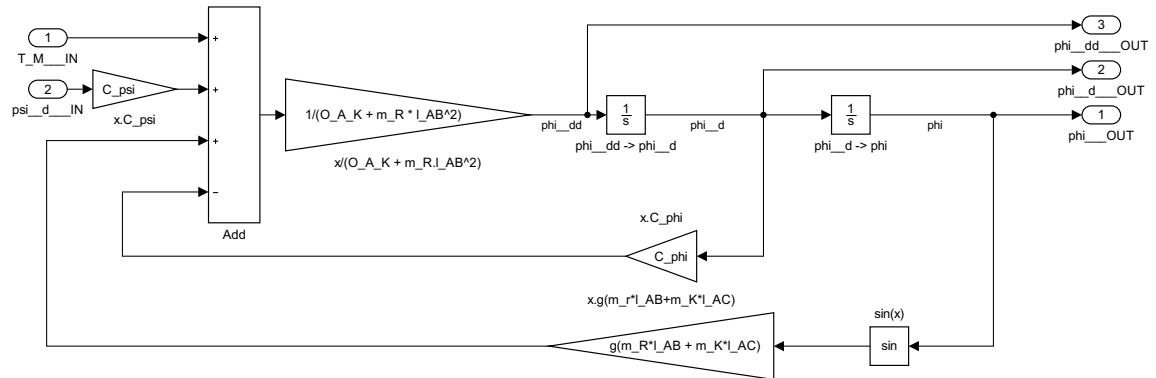


Abbildung 9: Subsystem Würfelseite, Quelle: eigene Darstellung

2.4.2.3 Simulation der Schwungmasse

Die Dynamik der Schwungmasse wird von 18 beschrieben, allerdings wird das Modell vereinfacht indem $\ddot{\phi}$ nicht substituiert wird.

$$\theta_B^R \cdot \ddot{\psi} = T_M - C_\psi \cdot \dot{\psi} - \theta_R^B \cdot \ddot{\phi} \quad (18)$$

Das Simulink-Modell folgt dem selben Schema wie das Subsystem zur Simulation der Bewegung des Würfelskörpers.

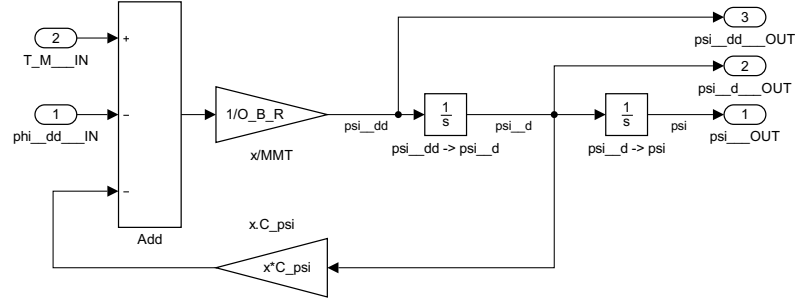


Abbildung 10: Subsystem Schwungmasse, Quelle: eigene Darstellung

2.5 Reglerentwurf

In dem folgenden Abschnitt wird der Entwurf eines Reglers vorgestellt, welcher auf der Rückführung des Zustandsvektors basiert. Im ersten Teil wird die analytische Bestimmung der Parameter erläutert, daraufhin wird der Regler im zweiten Teil an dem Prototyp erprobt und validiert.

2.5.1 Analytische Bestimmung der Reglerparameter

Mit Hilfe der Zustandsraumdarstellung kann über die Rückführung des Zustandvektors eine Regelung entworfen werden. Das folgende Blockschaltbild zeigt den Zusammenhang der Systemmatrizen und der Reglermatrix \mathbf{F} , welche zur Berechnung der Stellgröße $u = T_M$ dient.

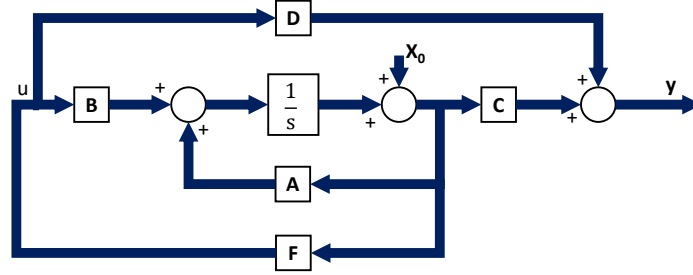


Abbildung 11: Blockschaltbild Regelkreis, Quelle: eigene Darstellung, Inhalt aus [13]

Die Stellgröße u wird von einem Mikrokontroller mit einer Abtatsperiod $T_a = 20ms$ berechnet. Folglich handelt es sich um eine digitale Regelung. Um das Verhalten des diskreten Systems zu beschreiben müssen die diskreten Systemmatrizen \mathbf{A}_d , \mathbf{B}_d , \mathbf{C}_d und \mathbf{D}_d berechnet werden. Hierfür gilt nach [13]:

$$\mathbf{S} = T_a \sum_{v=0}^{\infty} \mathbf{A}^v \frac{T^v}{(v+1)!} \quad (55)$$

$$\mathbf{A}_d = \mathbf{I} + \mathbf{S} \cdot \mathbf{A} \quad (56)$$

$$\mathbf{B}_d = \mathbf{S} \cdot \mathbf{B} \quad (57)$$

$$\mathbf{C}_d = \mathbf{C} \quad (58)$$

$$\mathbf{D}_d = \mathbf{D} \quad (59)$$

Die Reglermatrix \mathbf{F} wird als optimaler Zustandsregler nach dem quadratischen Gütekriterium entworfen. Die diskrete Gütefunktion für dieses System lautet:

$$I = \sum_{k=1}^{\infty} \mathbf{x}^T(k) \cdot \mathbf{Q} \cdot \mathbf{x}(k) + R \cdot u(k)^2 \quad (60)$$

Die Matrizen \mathbf{Q} und \mathbf{R} stellen Gewichtungen der Zustands- und Stellgrößen dar. Die Ausgangswerte dieser Matrizen werden mit der Faustformel nach ([14]) berechnet. Ggf. können die Werte anschließend angepasst werden um die Reglergüte weiter zu verbessern.

$$\mathbf{Q} = \begin{pmatrix} \frac{1}{(\varphi_{max})^2} & 0 & 0 \\ 0 & \frac{1}{(\dot{\varphi}_{max})^2} & 0 \\ 0 & 0 & \frac{1}{(\psi_{max})^2} \end{pmatrix} \quad (61)$$

$$R = \left(\frac{1}{(T_{M,max})^2} \right) \quad (62)$$

Die Reglermatrix \mathbf{F} muss die Eigenschaft besitzen die Gütefunktion (60) zu minimieren. Dieses Problem wird mit Hilfe von der Matlab-Funktion *lqr* numerisch gelöst.

2.5.2 Verifizierung des Reglers an dem 1D-Prototyp

Mit Hilfe von Matlab wurden die Werte der Reglermatrix \mathbf{F} berechnet.

$$\mathbf{F} = \begin{pmatrix} 0.8821 & 0.1386 & 0.0002 \end{pmatrix} \quad (63)$$

Somit lässt sich das Motormoment $T_{M,n}$ durch die Rückführung des Zustandsvektors \mathbf{x}_n über die Reglermatrix \mathbf{F} berechnen.

$$T_{M,n} = \mathbf{F} \cdot \mathbf{x}_n \quad (64)$$

Der Regler wird zuerst mit Hilfe eines Simulink-Modelles in der Simulation überprüft. Anschließend wird der geschlossene Regelkreis auf den Prototyp übertragen. Hierbei ist zu beachten, dass bei der Modellierung des Systemverhaltens die Annahme getroffen wurde, dass der Schwerpunkt des Systems auf dessen Y-Achse liegt. Durch den unsymmetrischen Aufbau ist dies allerdings nicht der Fall, somit ergibt sich das folgende Gravitationsmoment M_G , wobei der Winkel φ_{cog} den Winkel zwischen Y-Achse und Schwerpunkt der Würfelseite bezeichnet.

$$M_G = g(m_K \cdot l_{AC} + m_R \cdot l_{AB}) \cdot \sin(\varphi + \varphi_{cog}) \quad (65)$$

Somit muss ein konstantes Motormoment erzeugt werden, um die Würfelseite bei dem Sollwinkel $\varphi = 0$ zu halten. Dadurch wird die Schwungmasse konstant beschleunigt, weshalb die Schwungmasse nicht vollständig zum Stillstand kommen kann. Deshalb muss für die Berechnung des Drehmomentes der Winkel φ_{cog} zu der Zustandsgröße φ addiert werden.

$$T_{M,n} = \mathbf{F} \cdot \begin{pmatrix} \varphi + \varphi_{cog} \\ \dot{\varphi} \\ \dot{\psi} \end{pmatrix} \quad (66)$$

Da die Winkelgeschwindigkeit der Schwungmasse $\dot{\psi}$ nur dann verschwindet wenn das Motormoment gleich null ist, kann der Wert von φ_{cog} empirisch ermittelt werden.

Die folgenden Abbildungen zeigen den Verlauf der Winkelgeschwindigkeiten $\dot{\varphi}$ und $\dot{\psi}$. Hier zeigt sich, dass die Radgeschwindigkeit $\dot{\psi}$ nicht gegen null konvergiert. Mögliche Ursachen für dieses Verhalten sind neben der unsymmetrischen Konstruktion, die ungefilterten Werte von $\dot{\varphi}$ und $\dot{\psi}$, da deren Rauschsignale die Regelung negativ beeinflussen. Weiter können systematische Messabweichungen der Zustandsgrößen zu einer nicht verschwindenden Winkelgeschwindigkeit $\dot{\psi}$ führen.

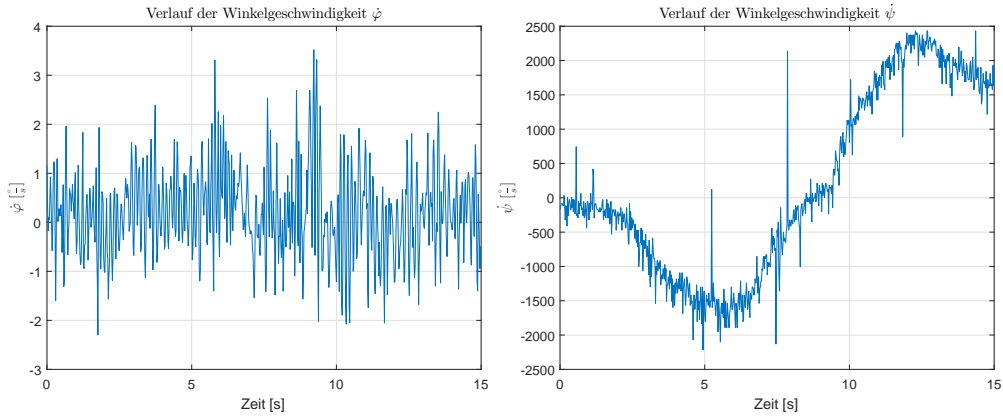


Abbildung 12: Verlauf von $\dot{\varphi}$ und $\dot{\psi}$, Quelle: eigene Darstellung

Die nächste Abbildung zeigt den Verlauf des Winkels φ , wobei die Werte der Winkelschätzung, des Komplementärfilters und des Kalman-Filters aufgezeigt sind. Hier lässt sich deutlich erkennen wie stark das rauschbehaftetes Signal der Winkelschätzung durch die jeweiligen Filter geglättet wird. Ein direkter Vergleich der beiden Filter ist ohne ein absolutes Referenzsignal nur schwer möglich.

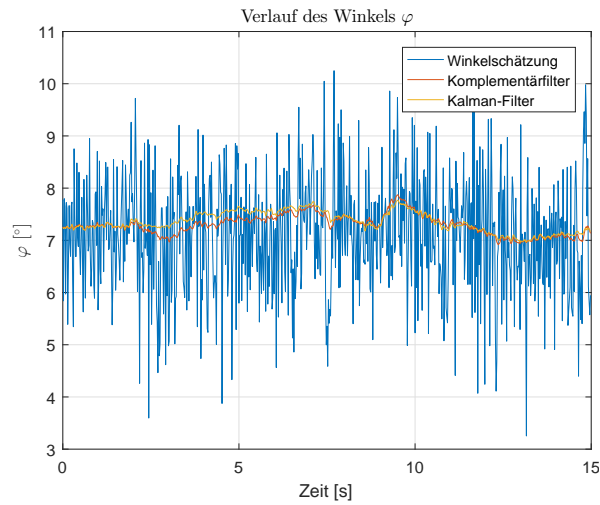


Abbildung 13: Verlauf von φ , Quelle: eigene Darstellung

2.6 Aufspringen

Das Aufspringen der Würfelseite wird durch das abrupte Bremsen der Schwungmasse ermöglicht. Hierbei wird der Drehimpuls der Schwungmasse auf das Gesamtsystem übertragen. Dieser Vorgang kann als nicht elastischer Stoß modelliert werden. Somit ergibt sich aus dem Drehimpulserhaltungssatz folgende Gleichung, wobei $\dot{\varphi}_B$ die Winkelgeschwindigkeit der Würfelseite nach dem Bremsen und $\dot{\psi}_B$ die Winkelgeschwindigkeit der Schwungmasse vor dem Bremsen darstellt.

$$(\theta_K^A + \theta_R^B + m_R \cdot l_{AB}) \cdot \dot{\varphi}_B = \theta_R^B \cdot \dot{\psi}_B \quad (67)$$

Um die Würfelseite von der Ruhelage ($\varphi_R = \pm \frac{\pi}{4}$) zu dem Gleichgewichtspunkt ($\varphi_G = 0$) zu bewegen muss Arbeit verrichtet werden. Diese Arbeit W ist gleich der Änderung der potentiellen Energie von der Ruhelage hin zu dem Gleichgewichtspunkt.

$$W = V(\varphi_G) - V(\varphi_R) = g(m_K + m_R)l_{AC} \cdot (\cos(\varphi_G) - \cos(\varphi_R)) \quad (68)$$

Auf Grund des Energieerhaltungssatzes muss die kinetische Energie des Gesamtsystemes nach dem Bremsvorgang gleich der zu leistenden Arbeit sein um die Würfelseite aufzurichten. Somit kann der Zusammenhang von $\dot{\varphi}_B$ und der Arbeit W wie folgt beschrieben werden.

$$\frac{1}{2}(\theta_K^A + \theta_R^B + m_R \cdot l_{AB})\dot{\varphi}_B^2 = g(m_K + m_R)l_{AC} \cdot (1 - \frac{1}{\sqrt{2}}) \quad (69)$$

Mit Hilfe der Gleichungen (69) und (67) kann nun die notwendige Bremsgeschwindigkeit $\dot{\psi}_B$ berechnet werden.

$$\dot{\psi}_B = \sqrt{(2 - \sqrt{2}(m_R + m_K) \cdot l_{AC} \cdot g \cdot \frac{\theta_K^A + \theta_R^B + m_R \cdot l_{AB}}{\theta_R^{B^2}})} \quad (70)$$

Das obige Modell geht von der Annahme aus, dass es sich um einen perfekt nicht elastischen Stoß handelt und bei der Bewegung der Würfelseite keine Energie verloren geht. Somit besteht eine Abweichung des Modells von den realen Bedingungen. Um diese Abweichungen zu minimieren wird ein, an den Gradientenabstieg angelehnter, Lernalgorithmus implementiert. Nach dem Abbremsen der Schwungmasse werden die Größen φ und $\dot{\varphi}$ beobachtet. Tritt ein Nulldurchgang von φ auf bedeutet dies, dass die Anfangsgeschwindigkeit $\dot{\varphi}_B$ und somit die Radgeschwindigkeit $\dot{\psi}_B$ zu hoch waren. Tritt jedoch ein Nulldurchgang von $\dot{\varphi}$ auf, folgt, dass $\dot{\varphi}_B$ und $\dot{\psi}_B$ zu niedrig waren. In beiden Fällen kann die Änderung der Energie ΔE , welche nötig ist um den Zielpunkt zu erreichen, berechnet werden.

$$\Delta E = \begin{cases} (1 - \cos(\varphi_0))(m_K + m_R)l_{AC} \cdot g & | \dot{\varphi} = 0 \\ -\frac{1}{2}(\theta_K^A + \theta_R^B + m_R \cdot l_{AB}) \cdot \dot{\varphi}_0^2 & | \varphi = 0 \end{cases} \quad (71)$$

Mit Hilfe der Drehimpuls- (67) und Energieerhaltung (69) wird nun aus der Energieänderung ΔE die nötige Änderung der Radgeschwindigkeit $\Delta \dot{\psi}_B$ berechnet.

$$\pm \Delta \dot{\psi}_B = \sqrt{2 \cdot \frac{\theta_K^A + \theta_R^B + m_R \cdot l_{AB}}{\theta_R^{B^2}} \cdot \Delta E} \quad (72)$$

Die Konvergenz des Lernalgorithmus gegen den Zielwert wird empirisch bewiesen, hierfür ist allerdings das Hinzufügen einer Lernrate μ erforderlich. Daraus ergibt sich letztendlich folgende Vorschrift um den aktuellen Wert der Bremsgeschwindigkeit $\dot{\psi}_B$ zu bestimmen.

$$\dot{\psi}_B := \dot{\psi}_B + \mu \cdot \Delta \dot{\psi}_B \quad | \quad 0 < \mu \leq 1 \quad (73)$$

2.7 Software

Im folgenden wird die Ansteuerung und Auswertung der elektrischen Komponenten näher beschrieben. Außerdem wird die Kommunikation zwischen der Host- und Target-Plattform erläutert. Als Zielplattform wird ein BeagleBone Black (BBB) verwendet, auf welchem eine Linux-Distribution ausgeführt wird. In der folgenden Abbildung sind die einzelnen Bausteine und deren Verbindung zu der Zielplattform dargestellt.

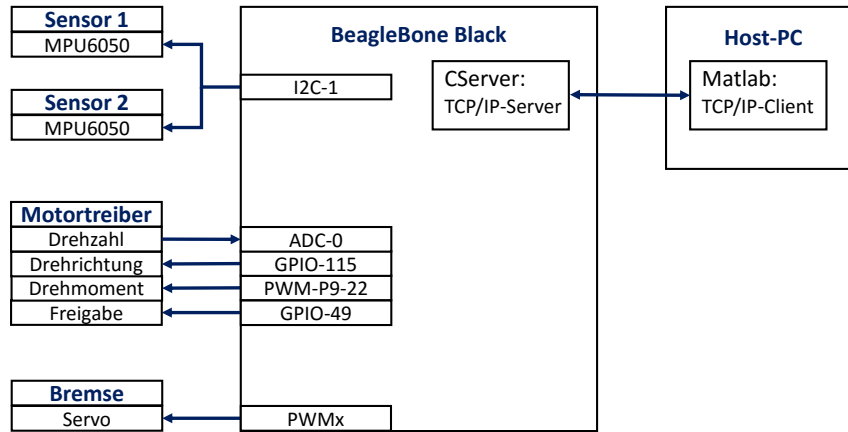


Abbildung 14: Blockschaltbild der Komponenten, Quelle: eigene Darstellung

Um die verschiedenen Versuche und Messungen durchzuführen wird eine Software-Basis entworfen, welche als Grundlage für die verschiedenen Applikationen dient. Diese Grundlage umfasst eine objektorientierte Kapselung der Hardware, die TCP-Server-Implementierung um mit der MATLAB-Anwendung zu kommunizieren, die Definition zweier Komponenten, welche als separate Tasks ausgeführt werden und eine Kommunikationsstruktur für den Datenaustausch. Der Aufbau dieser Bausteine wird in den folgenden Abschnitten näher erläutert. Insgesamt werden acht Versuche durchgeführt:

- V1: Bestimmung eines Ausgleichspolynoms um die Sensorwerte in SI-Einheiten umzurechnen. Hierfür werden die X- und Y-Beschleunigungswerte der beiden Sensoren benötigt.
- V2: Bestimmung des Offset der Gyroscopes. Hierfür werden die Z-Winkelgeschwindigkeiten der beiden Sensoren benötigt.
- V3: Bestimmung eines Ausgleichspolynoms um die ADC-Werte in SI-Einheiten umzurechnen. Hierfür werden die ADC-Werte der Motorgeschwindigkeit benötigt.
- V4: Test der Filter für φ , $\dot{\varphi}$ und (ψ) . Hierfür müssen die verschiedenen Filter in SW realisiert werden.
- V5: Bestimmung des Reibwertes C_φ .
- V6: Bestimmung des Reibwertes C_ψ .
- V7: Test und Bewertung des Regelkreises. Hierfür muss der Regelalgorithmus in SW realisiert werden.
- V8: Auspringen...

2.7.1 Hardware-Schnittstelle

Die Interaktion mit den Treibern des Betriebssystems wird mit Hilfe von Klassen gekapselt. Dadurch entsteht eine einheitliche und benutzerfreundliche Schnittstelle zwischen Hard- und Software. Die Klasse zur Interaktion mit der Hardware muss das folgende Interface erfüllen.

```
1 class IHardware
2 {
3 public:
4     virtual Int16 getX1__dd_raw() = 0;
5     virtual Int16 getX2__dd_raw() = 0;
6     virtual Int16 getY1__dd_raw() = 0;
7     virtual Int16 getY2__dd_raw() = 0;
8     virtual Int16 getPhi1__d_raw() = 0;
9     virtual Int16 getPhi2__d_raw() = 0;
10    virtual UInt16 getPsi__d_raw() = 0;
11    virtual void openBrake() = 0;
12    virtual void closeBrake() = 0;
13    virtual void enableMotor() = 0;
14    virtual void disableMotor() = 0;
15    virtual void setTorque(const Float32 torque) = 0;
16 public:
17     IHardware() = default;
18     IHardware(const IHardware&) = delete;
19     IHardware& operator=(const IHardware&) = delete;
20     virtual ~IHardware() = default;
21 };
```

2.7.2 TCP/IP-Verbindung

Die Kommunikation zwischen der BBB- und der MATLAB-Applikation verläuft über eine TCP/IP-Verbindung. Hierfür wird auf dem BBB ein Server ausgeführt, welcher auf die Verbindungsanfrage des MATLAB-Clients wartet. Die Vorteile des gewählten Protokolls liegen einerseits in der gesicherten und abstrahierten Datenübertragung, andererseits in den anwendungsfreundlichen Bibliotheken für Linux und MATLAB. Mit Hilfe einer simulierten Ethernet-Verbindung über USB wird ein privates Netzwerk zwischen der Zielplattform und dem Entwicklungs-PC eingerichtet.

Bild mit Adressen/Socket, etc.

2.7.3 Komponentenarchitektur

Die Hauptaufgaben der Applikation werden auf zwei Komponenten verteilt. Die erste übernimmt die Auswertung der Sensorik, Berechnung der Filter- und Regelalgorithmen und Ansteuerung der Aktorik. Hierfür werden Klassen eingeführt um die verschiedenen Teilaufgaben getrennt zu bearbeiten und einen übersichtlichen Software-Entwurf zu erhalten. Die Hauptaufgabe der zweiten Komponente besteht in der Kommunikation mit der MATLAB-Applikation, welche mit Hilfe eines TCP/IP-Servers erfolgt. Die beiden Komponenten werden von einer abstrakten Basisklasse abgeleitet und in eigenständigen Prozessen ausgeführt. In der Elternklasse werden rein virtuelle Methoden festgehalten welche zur Initialisierung und Durchführung der verschiedenen Versuche dienen. Somit ist es möglich die Grundstruktur für die mehreren Applikationen wiederzuverwenden. Auf Grund der simplen Abläufe genügt ein Flag um wiederzugeben ob sich eine Komponente im *STANDBY*- oder *RUNNING*-Zustand befinden. Für komplexere Anwendung kann an dieser Stelle ein vollständiger Zustandsautomat implementiert werden.

```
1 class AComponentBase
2 {
3 public:
4     virtual void init() = 0;
5     virtual void run_V1_AusgleichsPolynomAccelerometer() = 0;
6     virtual void run_V2_OffsetGyroscope() = 0;
7     virtual void run_V3_AusgleichsPolynomMotorADC() = 0;
8     virtual void run_V4_FilterTest() = 0;
9     virtual void run_V5_BestimmungC_psi() = 0;
10    virtual void run_V6_BestimmungC_phi() = 0;
11    virtual void run_V7_RegelungTest() = 0;
12 public:
13     AComponentBase(TQueue<Config::QueueSize>& rxQueue,
14                   TQueue<Config::QueueSize>& txQueue,
15                   bool initStandby);
16     AComponentBase() = delete;
17     AComponentBase(const AComponentBase&) = delete;
18     AComponentBase& operator=(const AComponentBase&) = delete;
19     virtual ~AComponentBase() = default;
20 protected:
21     TQueue<Config::QueueSize>& mRxQueue;
```

```

22   TQueue<Config::QueueSize>& mTxQueue;
23   bool mStandbyState;
24   UInt8 mPadding{};
25 };

```

Die Kommunikation der beiden Prozess auf dem BBB erfolgt mit Hilfe von Nachrichten. Dieser werden mit Hilfe von Queues, welche im Shared Memory (SHM) angelegt werden, zwischen den beiden Prozessen ausgetauscht. Zusätzlich wird dieselbe Nachrichtenstruktur verwendet um Daten zwischen dem BBB und der MATLAB-Anwendung auszutauschen. Hierfür werden die Daten als Byte-Stream über die TCP/IP-Verbindung übertragen.

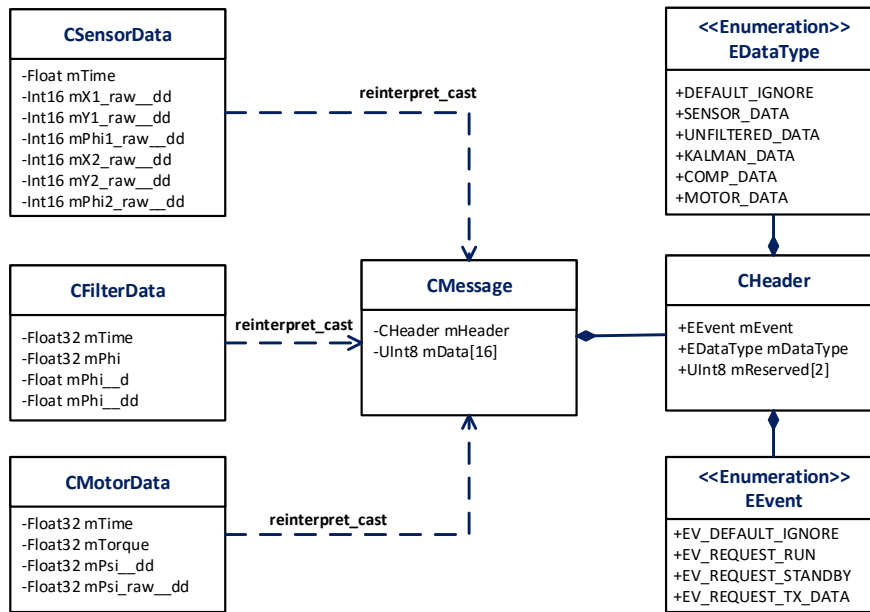


Abbildung 15: UML Nachrichtenaufbau, Quelle: eigene Darstellung

Die Nachrichten bestehen aus einem Header, welcher das Ereignis und weitere Informationen, wie z.B. den Typ der enthaltenen Daten. Das Datenfeld wird genutzt um relevante Informationen wie Sensor-, Filter- und Motorwerte an die MATLAB-Applikation zu übermitteln. Die Nachrichten werden in Form einer Klasse realisiert, welche Konstruktoren enthält um die verschiedenen Datenstrukturen in das Datenfeld zu legen.

2.7.4 Allgemeiner Ablauf der Versuchsapplikationen

Die Ablauf der unterschiedlichen Versuche ist auf Seite der SW sehr ähnlich. Die beiden Komponenten starten im *STANDBY*-Zustand. Die Kommunikations-Komponente wartet darauf, dass der MATLAB-Client eine Verbindung aufbaut. Daraufhin wechselt diese in den *RUNNING*-Zustand und sendet eine Nachricht an die *Control*-Komponente, welche darauf ebenfalls in den *RUNNING*-Zustand wechselt. Im Anschluss wertet die *Control*-Komponente in zyklischen Abständen die Sensoren aus, berechnet ggf. Filter- oder Reglerwerte und sendet diese mit Hilfe von Nachrichten an die Kommunikations-Komponente. Diese leitet empfangene Nachrichten an die MATLAB-Applikation weiter, welche diese virtualisiert und zur weiteren Verarbeitung speichert. Sobald die MATLAB-Anwendung die Verbindung abbricht wird eine entsprechende Nachricht an die *Control*-Komponente gesendet. Daraufhin terminieren beide Prozesse selbständig.

3 3D-Modell

filler

3.1 Modellierung der Systemdynamik

In diesem Abschnitt werden die Bewegungsgleichungen des 3D-Modells mit Hilfe des Lagrange-Formalismus hergeleitet. Der Würfelförper besitzt drei rotatorische Freiheitsgrade, die drei Schwungmassen verfügen über jeweils einen Freiheitsgrad. Somit ergeben sich insgesamt sechs Freiheitsgrade für das Gesamtsystem. Dadurch steigt die Komplexität der Systemdynamik stark an, allerdings bestehen nach wie vor Parallelen zu der Dynamik des 1D-Modells.

Um die Position des Würfels zu beschreiben wird ein raumfestes Bezugssystem $\{I\}$ eingeführt, welches von den drei Einheitsvektoren ${}^I e_x$, ${}^I e_y$ und ${}^I e_z$ beschrieben wird. Das zweite Bezugssystem $\{W\}$ ist körperfest und rotiert somit mit dem Würfel. Es wird von den Einheitsvektoren ${}^w e_x$, ${}^w e_y$ und ${}^w e_z$ beschrieben.

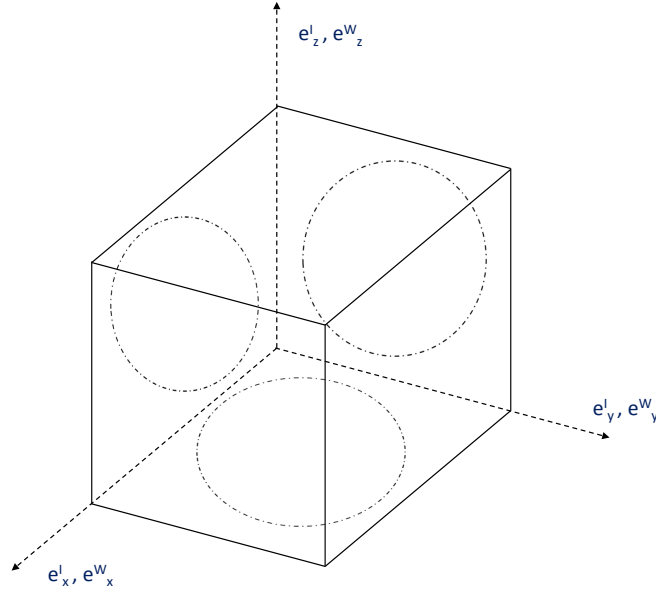


Abbildung 16: Mechanischer Aufbau, Quelle: eigene Darstellung

Die aktuelle Position des Würfels kann somit eindeutig durch die Verschiebung des körperfesten Bezugssystems $\{W\}$ zu dem raumfesten Bezugssystem $\{I\}$ bestimmt werden. Um diese Rotation zu beschreiben werden die drei Euler-Winkel φ_1 , φ_2 und φ_3 eingeführt. Die Drehreihenfolge und Drehachsen werden im folgenden beschrieben.

Winkel	Beschreibung
φ_1	Drehung um ${}^I e_z$
φ_2	Drehung um ${}^w e_x$
φ_3	Drehung um ${}^w e_z$

Die erste Ableitung gewöhnliche Ableitung der Euler-Winkel entspricht den Winkelgeschwindigkeiten des Würfels um die oben genannten Achsen.

Mit Hilfe der Euler-Winkel können Drehmatrizen definiert werden um Koordinaten in dem raumfesten Bezugssystem $\{I\}$ in das körperfeste Bezugssystem $\{W\}$ zu projizieren. Hier zeigt sich wieder die Bedeutung der Reihenfolge der einzelnen Drehungen, da auch die Matrizenmultiplikation im Allgemeinen nicht kommutativ ist.

$${}^w \mathbf{r} = {}^w \mathbf{D}_{\varphi_3} \cdot ({}^w \mathbf{D}_{\varphi_2} \cdot ({}^w \mathbf{D}_{\varphi_1} \cdot {}^I \mathbf{r})) = {}^w \mathbf{D} \cdot {}^I \mathbf{r} \quad (74)$$

$${}^w \mathbf{D}_{\varphi_1} = \begin{pmatrix} c_{\varphi_1} & -s_{\varphi_1} & 0 \\ s_{\varphi_1} & c_{\varphi_1} & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad {}^w \mathbf{D}_{\varphi_2} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_{\varphi_2} & -s_{\varphi_2} \\ 0 & s_{\varphi_2} & c_{\varphi_2} \end{pmatrix} \quad {}^w \mathbf{D}_{\varphi_3} = \begin{pmatrix} c_{\varphi_3} & -s_{\varphi_3} & 0 \\ s_{\varphi_3} & c_{\varphi_3} & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (75)$$

$${}^w\mathbf{D} = \begin{pmatrix} c_{\varphi_1}c_{\varphi_3} - s_{\varphi_1}c_{\varphi_2}s_{\varphi_3} & -c_{\varphi_1}s_{\varphi_3} - s_{\varphi_1}c_{\varphi_2}c_{\varphi_3} & s_{\varphi_1}s_{\varphi_2} \\ s_{\varphi_1}c_{\varphi_3} + c_{\varphi_1}c_{\varphi_2}s_{\varphi_3} & -s_{\varphi_1}s_{\varphi_3} + c_{\varphi_1}c_{\varphi_2}c_{\varphi_3} & -c_{\varphi_1}s_{\varphi_2} \\ s_{\varphi_2}s_{\varphi_3} & s_{\varphi_2}c_{\varphi_3} & c_{\varphi_2} \end{pmatrix} \quad (76)$$

Die Projektion einer Koordinate aus dem körperfesten in das raumfeste Bezugssystem erfolgt durch die transponierte der Matrix ${}^w\mathbf{D}$.

$${}^I\mathbf{r} = {}^I\mathbf{D} \cdot {}^I\mathbf{r} = {}^w\mathbf{D}^T \cdot {}^I\mathbf{r} \quad (77)$$

Die Bewegung der Schwungmassen relativ zu dem Würfelförper wird von den drei Winkeln ψ_1 , ψ_2 und ψ_3 beschrieben. Deren zeitliche Ableitungen stellen die Winkelgeschwindigkeiten der Schwungräder dar.

$$\boldsymbol{\psi} = \begin{pmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \end{pmatrix} \quad \dot{\boldsymbol{\psi}} = \begin{pmatrix} \dot{\psi}_1 \\ \dot{\psi}_2 \\ \dot{\psi}_3 \end{pmatrix} \quad (78)$$

Die Euler-Winkel φ und die Ausfallwinkel der Schwungmassen $\boldsymbol{\psi}$ werden als generalisierte Koordinaten verwendet.

$$\mathbf{q} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \\ q_5 \\ q_6 \end{pmatrix} = \begin{pmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ \psi_1 \\ \psi_2 \\ \psi_3 \end{pmatrix} \quad (79)$$

3.1.1 Potential des Systems

Um die Lagrange-Funktion L des Systems zu bestimmen muss einerseits die kinetische Energie T und die potentielle Energie V ermittelt werden. Das Potential des Würfels wird durch die aktuelle Lage seines Schwerpunktes \mathbf{r} bestimmt, hierbei ist lediglich die z-Komponente des raumfesten Bezugssystem von Bedeutung.

$$V = m_G \cdot g \cdot {}^I z_{cog} \quad (80)$$

Die Position des Schwerpunktes im körperfesten Bezugssystem ist fix. Durch die Projektion dieses Vektors ${}^w\mathbf{r}_{cog}$ in das raumfeste Bezugssystem $\{I\}$ wird die Abhängigkeit von der aktuellen Verschiebung berücksichtigt.

$${}^I\mathbf{r}_C = {}^I\mathbf{D} \cdot {}^w\mathbf{r}_C = {}^I\mathbf{D} \cdot \begin{pmatrix} {}^w x_C \\ {}^w y_C \\ {}^w z_C \end{pmatrix} \quad (81)$$

Folglich ergibt sich der folgende Zusammenhang für das Potential V und die aktuelle Ausrichtung des Würfels.

$$V = m_G \cdot g \cdot {}^I z_{cog} = m_G \cdot g \cdot (s_{\varphi_2}s_{\varphi_3} \cdot {}^w x_C + s_{\varphi_2}c_{\varphi_3} \cdot {}^w y_C + c_{\varphi_2} \cdot {}^w z_C) \quad (82)$$

3.1.2 Kinetische Energie des Systems

Die kinetische Energie setzt sich aus der Winkelgeschwindigkeit des Würfels $\boldsymbol{\omega}_K$ und der Geschwindigkeiten der drei Schwungmassen $\boldsymbol{\omega}_R$ zusammen. Hierbei ist zu beachten, dass die Winkelgeschwindigkeiten in verschiedenen Bezugssystemen darstellbar sind und die kinetische Energie von der Darstellungsform unabhängig ist. Um dies zu gewährleisten müssen allerdings auch die Trägheitstensoren in das jeweilige Bezugssystem projiziert werden.

$${}^w\boldsymbol{\omega}_K = \begin{pmatrix} {}^w\omega_x \\ {}^w\omega_y \\ {}^w\omega_z \end{pmatrix} \quad {}^w\boldsymbol{\omega}_R = \dot{\boldsymbol{\psi}} \quad (83)$$

$$T = \frac{1}{2} {}^w\boldsymbol{\omega}_K^T \cdot ({}^w\boldsymbol{\Theta}_G - {}^w\boldsymbol{\Theta}_R) \cdot {}^w\boldsymbol{\omega}_K + \frac{1}{2} ({}^w\boldsymbol{\omega}_K + {}^w\boldsymbol{\omega}_R)^T \cdot {}^w\boldsymbol{\Theta}_R \cdot ({}^w\boldsymbol{\omega}_K + {}^w\boldsymbol{\omega}_R) \quad (84)$$

Um das die kinetische Energie T mit Hilfe der generalisierten Koordinaten darzustellen muss die Winkelgeschwindigkeit ${}^w\omega_W$ als die erste Ableitung der Euler-Winkel φ darstellen. Um die Projektion der Euler-Geschwindigkeiten $\dot{\varphi}$ zu veranschaulichen kann die Rotationsgeschwindigkeit des Würfels ${}^w\omega_W$ als Summe der Euler-Komponenten dargestellt werden.

$${}^w\omega_W = {}^w\dot{\varphi}_1 + {}^w\dot{\varphi}_2 + {}^w\dot{\varphi}_3 \quad (85)$$

Um die Euler-Geschwindigkeit in dem körperfesten Bezugssystem darzustellen werden Rotationsachsen um die entsprechenden Euler-Winkel rotiert. Die Winkelgeschwindigkeit $\dot{\varphi}_3$ ist gleich der Rotation um die körperfeste ${}^w e_z$ und muss somit nicht über eine Drehmatrix projiziert werden.

$${}^w\dot{\varphi}_3 = \begin{pmatrix} 0 \\ 0 \\ {}^w\omega_z \end{pmatrix} \quad (86)$$

Die Winkelgeschwindigkeit $\dot{\varphi}_2$ entspricht der Rotation um die körperfeste Achse ${}^w e_x$ vor der Rotation um den Winkel φ_3 . Somit muss mit Hilfe der Matrix ${}^wD_{\varphi_3}$ der Geschwindigkeitsvektor in das endgültige körperfeste Bezugssystem projiziert werden.

$${}^w\dot{\varphi}_2 = {}^wD_{\varphi_3} \cdot \begin{pmatrix} \dot{\varphi}_2 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \dot{\varphi}_2 \cdot c_{\varphi_3} \\ -\dot{\varphi}_2 \cdot s_{\varphi_3} \\ 0 \end{pmatrix} \quad (87)$$

Die dritte Komponente ist die Winkelgeschwindigkeit $\dot{\varphi}_1$, welche der Rotation um die raumfeste Achse ${}^I e_z$ entspricht. Folglich muss zu einer Projektion in das körperfeste Bezugssystem der Geschwindigkeitsvektor mit den Matrizen ${}^wD_{\varphi_3}$ und ${}^wD_{\varphi_2}$ multipliziert werden.

$${}^w\omega_W = {}^w\dot{\varphi}_1 + {}^w\dot{\varphi}_2 + {}^w\dot{\varphi}_3 = ({}^wD_{\varphi_2}({}^wD_{\varphi_3} \cdot \begin{pmatrix} \dot{\varphi}_1 \\ 0 \\ 0 \end{pmatrix})) = \begin{pmatrix} \dot{\varphi}_1 \cdot s_{\varphi_2} \cdot s_{\varphi_3} \\ \dot{\varphi}_1 \cdot s_{\varphi_2} \cdot c_{\varphi_3} \\ \dot{\varphi}_1 \cdot c_{\varphi_2} \end{pmatrix} \quad (88)$$

Die gesamte Winkelgeschwindigkeit des Würfels ${}^w\omega_K$ ergibt sich wie bereits beschrieben aus der Summe der einzelnen Komponenten.

$${}^w\dot{\varphi} = \begin{pmatrix} \dot{\varphi}_1 \cdot s_{\varphi_2} \cdot s_{\varphi_3} + \dot{\varphi}_2 \dot{\varphi}_3 \\ \dot{\varphi}_1 \cdot s_{\varphi_2} \cdot c_{\varphi_3} - \dot{\varphi}_2 \cdot s_{\varphi_3} \\ \dot{\varphi}_1 \cdot c_{\varphi_2} + \dot{\varphi}_3 \end{pmatrix} \quad (89)$$

3.1.3 Generalisierte Kraftkomponenten

In der Untersuchung des 1D-Modelles wurde bereits gezeigt, dass der Würfel ein nicht konservatives System ist, da einerseits über die Motoren mechanische Energie zugeführt wird und andererseits durch Reibung mechanische Energie verloren geht. Deshalb müssen die generalisierten Kraftkomponenten bestimmt werden um mit Hilfe des d'Alembert'schen Prinzip die Bewegungsgleichungen zu ermitteln.

Wie bereits angesprochen erzeugen die Motoren Momente, welche die Schwungmassen antreiben. Gleichmaßen entsteht in den Lagern der Räder ein Reibmoment welches als linear abhängig von der Winkelgeschwindigkeit modelliert wird. Somit ergibt sich das folgende Moment.

$${}^wM_M = {}^wT_M - C_{\psi} \cdot {}^w\dot{\psi} \quad C_{\psi} = \begin{pmatrix} C_{\psi_1} & 0 & 0 \\ 0 & C_{\psi_2} & 0 \\ 0 & 0 & C_{\psi_3} \end{pmatrix} \quad (90)$$

Die Komponenten dieses Momentvektors sind gleichermaßen die generalisierten Kraftkomponenten der Koordinaten ψ_1 , ψ_2 und ψ_3 .

$$Q_{\psi_1} = Q_4 = T_{M1} - C_{\psi_1} \cdot \dot{\psi}_1 \quad (91)$$

$$Q_{\psi_2} = Q_5 = T_{M2} - C_{\psi_2} \cdot \dot{\psi}_2 \quad (92)$$

$$Q_{\psi_3} = Q_6 = T_{M3} - C_{\psi_3} \cdot \dot{\psi}_3 \quad (93)$$

Die Kraftkomponenten, welche den generalisierten Koordinaten φ_1 , φ_2 und φ_3 zugeordnet sind lassen sich aus dem Potential V herleiten.

$$Q_{\varphi_1} = \frac{\partial V}{\partial \varphi_1} = 0 \quad (94)$$

$$Q_{\varphi_2} = \frac{\partial V}{\partial \varphi_2} = m_G \cdot g \cdot (c_{\varphi_2} s_{\varphi_3} \cdot {}^w y_C + c_{\varphi_2} c_{\varphi_3} \cdot {}^w y_C - s_{\varphi_2} \cdot {}^w z_C) \quad (95)$$

$$Q_{\varphi_3} = \frac{\partial V}{\partial \varphi_3} = m_G \cdot g \cdot (s_{\varphi_2} c_{\varphi_3} \cdot {}^w x_C - s_{\varphi_2} c_{\varphi_3} \cdot {}^w y_C) \quad (96)$$

3.1.3.1 Test

$$\begin{aligned} T = & \frac{(\dot{\varphi}_2 \cos(\varphi_3) + \dot{\varphi}_1 \sin(\varphi_2) \sin(\varphi_3)) (O_{K31} (\dot{\varphi}_3 + \dot{\varphi}_1 \cos(\varphi_2)) + O_{K11} (\dot{\varphi}_2 \cos(\varphi_3) + \dot{\varphi}_1 \sin(\varphi_2) \sin(\varphi_3)) - O_{K21} (\dot{\varphi}_2 \sin(\varphi_3) - \dot{\varphi}_1 \cos(\varphi_3) \sin(\varphi_2)) (O_{K32} (\dot{\varphi}_3 + \dot{\varphi}_1 \cos(\varphi_2)) + O_{K12} (\dot{\varphi}_2 \cos(\varphi_3) + \dot{\varphi}_1 \sin(\varphi_2) \sin(\varphi_3)) - O_{K22} (\dot{\varphi}_2 \sin(\varphi_3) - \dot{\varphi}_1 \cos(\varphi_3) \sin(\varphi_2)))}{2} \\ & + \frac{(O_{K33} (\dot{\varphi}_3 + \text{psi_d_3} + \dot{\varphi}_1 \cos(\varphi_2)) + O_{K13} (\text{psi_d_1} + \dot{\varphi}_2 \cos(\varphi_3) + \dot{\varphi}_1 \sin(\varphi_2) \sin(\varphi_3)) + O_{K23} (\text{psi_d_2} - \dot{\varphi}_2 \sin(\varphi_3) + \dot{\varphi}_1 \cos(\varphi_3) \sin(\varphi_2))) (O_{K31} (\dot{\varphi}_3 + \text{psi_d_3} + \dot{\varphi}_1 \cos(\varphi_2)) + O_{K11} (\text{psi_d_1} + \dot{\varphi}_2 \cos(\varphi_3) + \dot{\varphi}_1 \sin(\varphi_2) \sin(\varphi_3)) - O_{K21} (\dot{\varphi}_2 \sin(\varphi_3) - \dot{\varphi}_1 \cos(\varphi_3) \sin(\varphi_2)))}{2} \\ & + \frac{(\text{psi_d_1} + \dot{\varphi}_2 \cos(\varphi_3) + \dot{\varphi}_1 \sin(\varphi_2) \sin(\varphi_3)) (O_{K31} (\dot{\varphi}_3 + \text{psi_d_3} + \dot{\varphi}_1 \cos(\varphi_2)) + O_{K11} (\text{psi_d_1} + \dot{\varphi}_2 \cos(\varphi_3) + \dot{\varphi}_1 \sin(\varphi_2) \sin(\varphi_3)) - O_{K21} (\dot{\varphi}_2 \sin(\varphi_3) - \dot{\varphi}_1 \cos(\varphi_3) \sin(\varphi_2)))}{2} \\ & + \frac{(\text{psi_d_2} - \dot{\varphi}_2 \sin(\varphi_3) + \dot{\varphi}_1 \cos(\varphi_3) \sin(\varphi_2)) (O_{K32} (\dot{\varphi}_3 + \text{psi_d_3} + \dot{\varphi}_1 \cos(\varphi_2)) + O_{K12} (\text{psi_d_1} + \dot{\varphi}_2 \cos(\varphi_3) + \dot{\varphi}_1 \sin(\varphi_2) \sin(\varphi_3)) - O_{K22} (\dot{\varphi}_2 \sin(\varphi_3) - \dot{\varphi}_1 \cos(\varphi_3) \sin(\varphi_2)))}{2} \\ & + \frac{(\dot{\varphi}_3 + \dot{\varphi}_1 \cos(\varphi_2)) (O_{K33} (\dot{\varphi}_3 + \dot{\varphi}_1 \cos(\varphi_2)) + O_{K13} (\dot{\varphi}_2 \cos(\varphi_3) + \dot{\varphi}_1 \sin(\varphi_2) \sin(\varphi_3)) - O_{K23} (\dot{\varphi}_2 \sin(\varphi_3) - \dot{\varphi}_1 \cos(\varphi_3) \sin(\varphi_2)))}{2} \end{aligned}$$

Literatur

- [1] Mohanarjah Gajamohan, Michael merz, Igor Thommen, Raffaello D'Andrea: The Cubli: A Cube that can Jump Up and Balance
- [2] Mohanarajah Gajamohan, Michael Muehlbach, Tobias Widmer, Raffaello D'Andrea: The Cubli: A Reaction Wheel Based 3D Inverted Pendulum
- [3] Michael Muehlbach, Gajamohan Mohanarajah, Raffaello D'Andrea: Nonlinear Analysis and Control of a Reaction Wheel-based 3D Inverted Pendulum
- [4] Wolfgang Nolting: Grundkurs Theoretische Physik 1 - Klassische Mechanik
- [5] Wolfgang Nolting: Grundkurs Theoretische Physik 2 - Analytische Mechanik
- [6] Thomas R. Kane: Dynamics - Theory and Applications
- [7] Fernando Puente León, Sebastian Bauer: Praxis der digitalen Signalverarbeitung
- [8] Josef Hoffmann, Franz Quint : Simulation technischer linearer und nichtlinearer Systeme mit MATLAB/Simulink
- [9] Herbert Schlitt: Systemtheorie für stochastische Prozesse
- [10] Marin Meyer: Signalverarbeitung - Analoge und digitale Signale, Systeme und Filter
- [11] Ottmar Beucher: Signale und Systeme - Theorie, Simulation und Anwendung
- [12] Heinz Unbehauen: Regelungstechnik 1 - Klassische Verfahren zur Analyse und Synthese linearer kontinuierlicher Regelsysteme
- [13] Heinz Unbehauen: Regelungstechnik 2 - Zustandsregelungen, digitale und nichtlineare Regelsysteme
- [14] Joao P. Hespanha: Lecture notes on LQR/LQG controller design
- [15] Tom M. Mitchell: Machine Learning
- [16] Christopher Bishop: Pattern Recognition and Machine Learning
- [17] Joachim Wietzke, Manh Tien Tran: Automotive Embedded Systeme: Effizientes Framework - Vom Design zur Implementierung
- [18] Joachim Wietzke: Embedded Technologies: Vom Treiber bis zur Grafik-Anbindung
- [19] Derek Molloy: Exploring BeagleBone: Tools and Techniques for Building with Embedded Linux