



Simulation von Multikopter-Systemen für die effiziente Entwicklung neuer Regelungsalgorithmen

Jan Vervoorst
Intel Deutschland GmbH
jan.willem.vervoorst@intel.com

- Überblick Multikoptersysteme
- Modellierung der Systemdynamik
 - Geometrie/Aerodynamik/Diskrete Ereignisse
 - Komponenten der Systemdynamik-Simulation
- Modellierung von Guidance/Navigation/Control (GNC) Komponenten
 - Zustandsschätzung/Sensorik
 - Regelkreise
- Einbindung der Simulation in den Entwicklungsprozess
 - Initiale Tests und Auslegung von GNC-Algorithmen
- Zusammenfassung/Ausblick

Übersicht Multikoptersysteme

- Intel® Falcon™ 8+ System



- Professionelle Flugdrohne zur industriellen Nutzung (Inspektion, Vermessung, ...)
- Dreifach redundantes Autopilotensystem basierend auf AscTec Trinity Technology
- Patentierte V-Form mit 8 Rotoren
- Max. Abfluggewicht 2,8kg
- Abmessungen ca. 770 x 820 x 160mm

Übersicht Multikoptersysteme

- Intel® Shooting Star™ Drone



- Flugdrohne speziell für Lichtshows
- Farb-LED mit über 4 Mio. Farbkombinationen
- Symmetrische Quadrocopter-Struktur
- Gesamtgewicht ca. 300g

Übersicht Multikoptersysteme

- Volocopter VC200

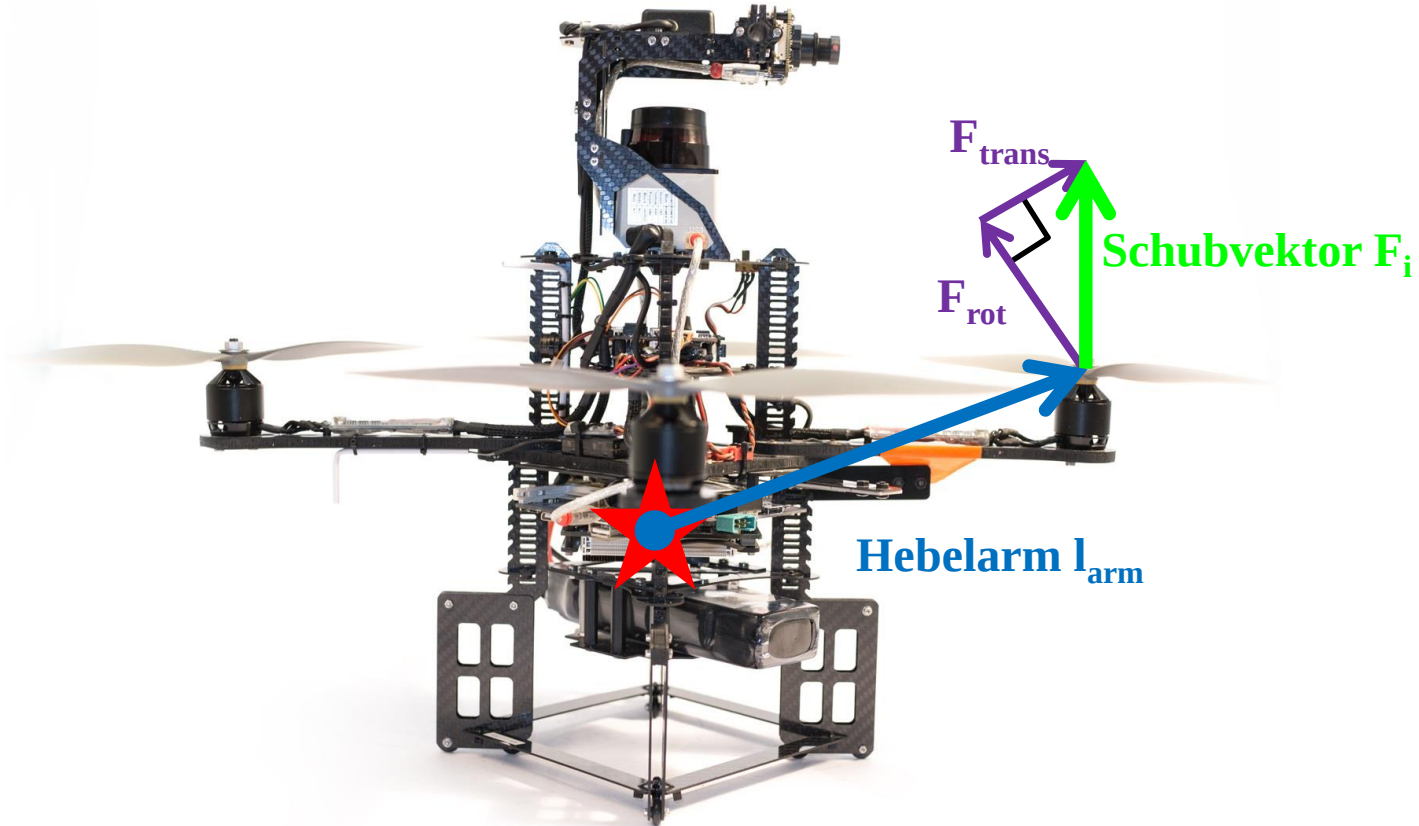


- Senkrechtstarter (VTOL) für 2 Personen
- Reiner Elektroantrieb
- Autopilotensystem basierend auf AscTec Trinity Technology
- Multikoptersystem mit 18 Rotoren
- Zulassung als Ultraleichtflugzeug mit maximalem Gesamtgewicht bis 450kg

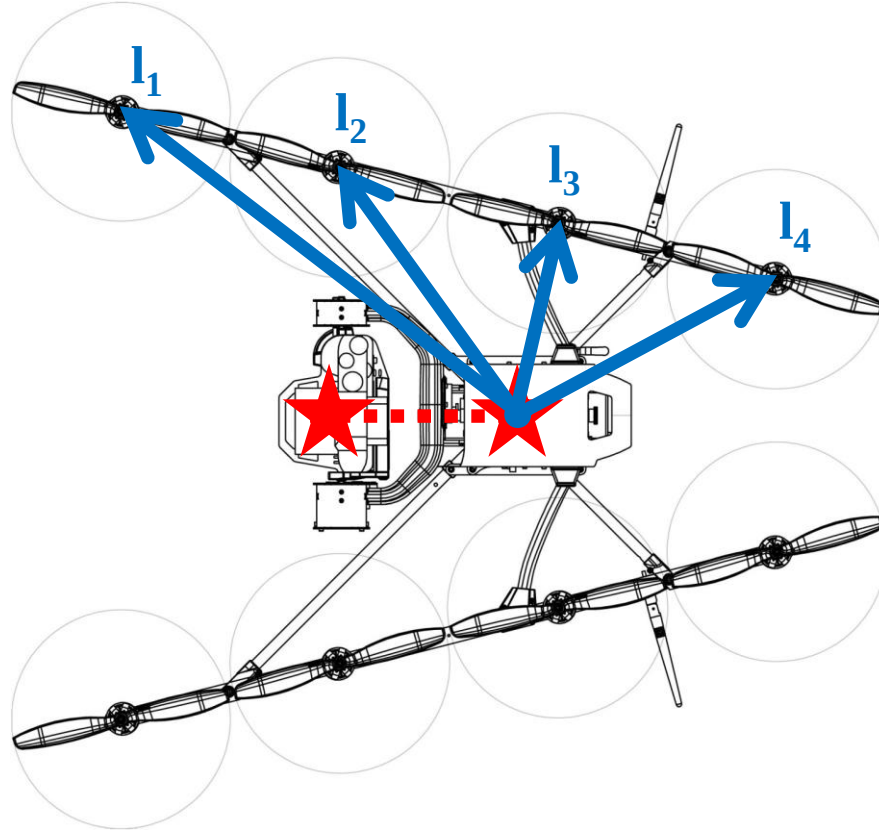
Modellierung der Systemdynamik

- Die Simulation sollte flexibel alle Multikopter-Strukturen abbilden können
 - Beliebige Anzahl Rotoren, Geometrie, Größe, Gewicht, ...
- Das reale Flugverhalten sollte von der Simulation bestmöglich reproduziert werden
 - Welche Kräfte wirken auf das Flugsystem während des Fluges?
- Vereinfachte Modelle reichen dafür nicht aus, daher müssen alle relevanten Wirkeffekte detailliert abgebildet werden
 - Geometrie
 - Aerodynamik
 - Diskrete Ereignisse
- Komplett modularer Aufbau der Simulation erlaubt hohe Flexibilität
 - Alle Komplexitäten können vom Nutzer wahlweise aus-/zugeschaltet werden
 - Systemdynamik als Teil einer Simulation des Gesamtsystems

- Hebelarm zum echten Schwerpunkt



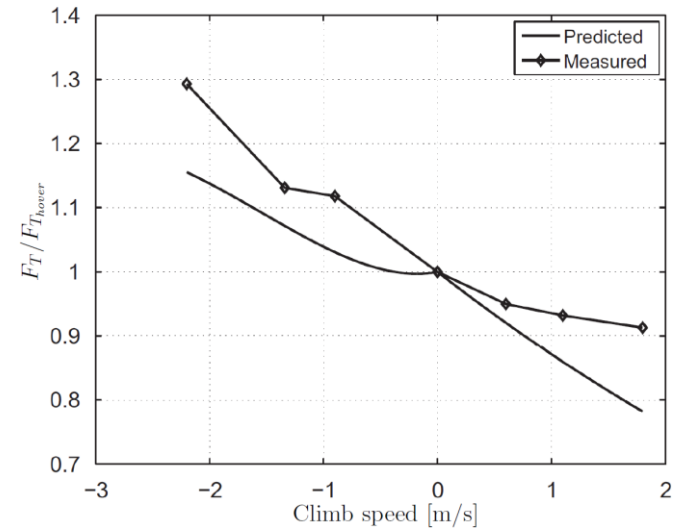
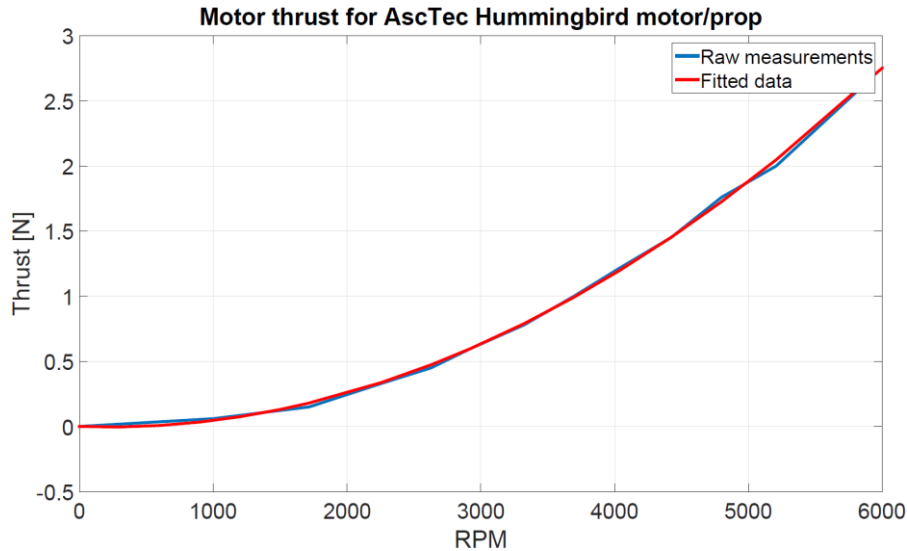
- Nichtsymmetrische Geometrie



- Außermittige Kräfte und Aufnahme/Abwurf von Lasten
 - Außermittige Kräfte machen den Trägheitstensor nichtdiagonal und nichtsymmetrisch
 - Masse und Trägheitsmoment ändern sich, wenn Lasten aufgenommen/abgeworfen werden
- Alle Systemparameter müssen dynamisch konfigurierbar sein

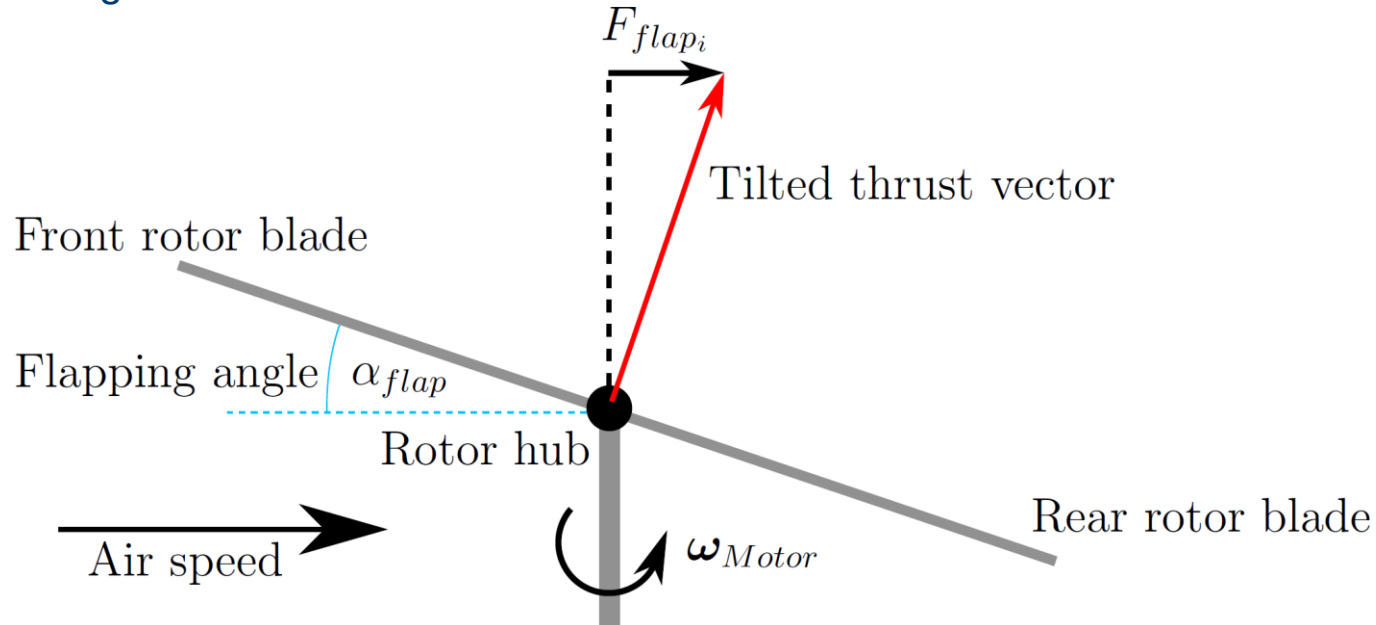


- **Aerodynamischer Widerstand**
 - Die Struktur des Fluggeräts hat einen aerodynamischen Querschnitt und erzeugt Widerstand
 - Dieser Widerstand wirkt in der Translation als Dämpfungskraft
- **Dynamischer Schub**
 - Propeller-Kennwerte gibt es meist nur für statischen Schub
 - Maximale Fluggeschwindigkeit von ca. 15m/s hat signifikanten Einfluss auf wirkenden Schub



- Propeller-Effekte

- Seitliche Anströmung eines Propellers führt zu einer dämpfenden Rückstellkraft
 - Unterschiedliche Anströmung des vorwärts und rückwärts drehenden Propellerblatts führt zu einer Verbiegung und Verkipfung der Propellerebene nach hinten → Blade Flapping
- Die Richtung des Schubvektors verändert sich



Diskrete Ereignisse

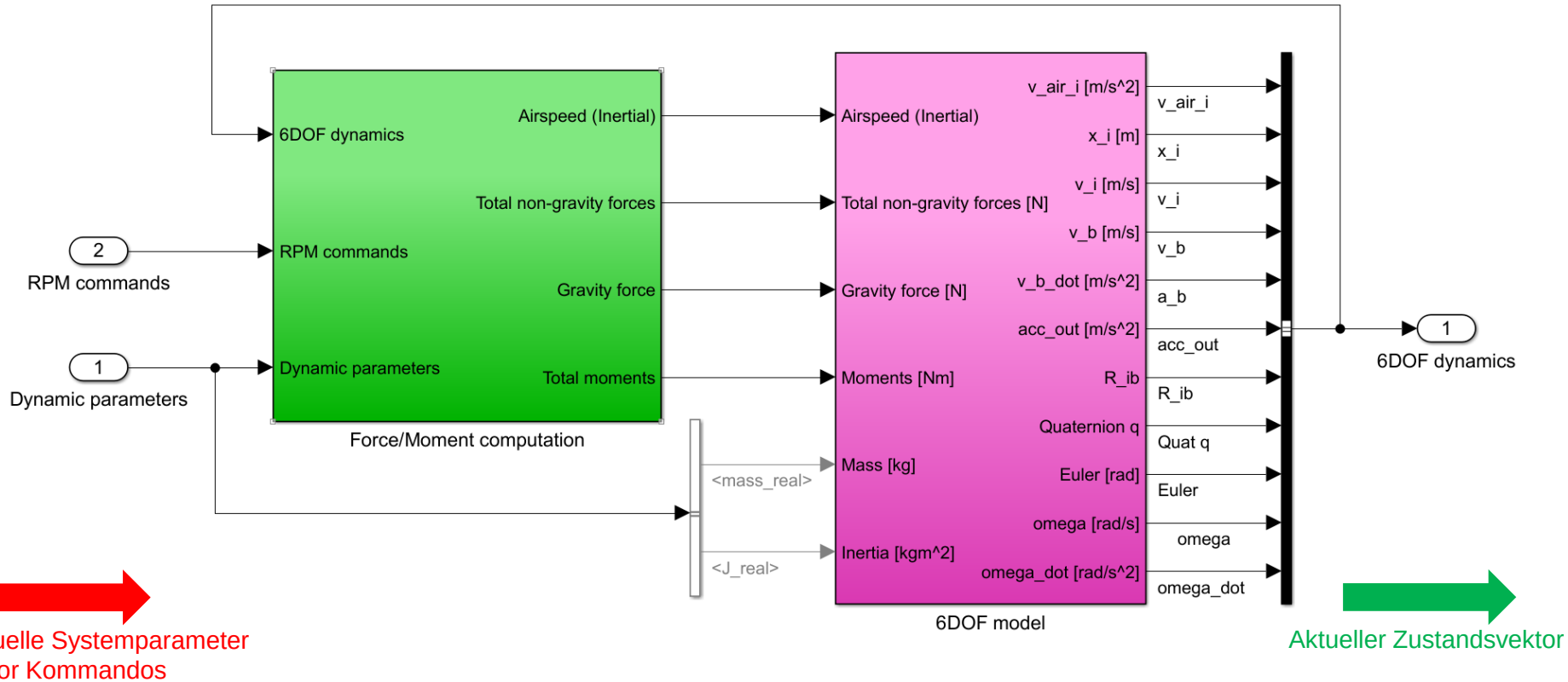
- Diskrete Ereignisse sind eine Reihe von mechanischen/elektrischen/elektronischen Veränderungen, die während des Betriebs auftreten können:
 - Aktuatorfehler:
 - Motorschaden
 - Verlust eines Propellers (komplett oder teilweise)
 - Veränderung der Aktuatordynamik (z. B. schwergängiger Lauf aufgrund von Beschädigung)
 - Aufnahme/Absetzen von Zuladung
 - Änderungen an Masse, Massenschwerpunkt, Trägheitsmomenten
 - Strukturschäden
 - Beschädigung/Verlust von Teilen
 - Verformung der Struktur → veränderte Schubverhältnisse
 - Sensorfehler
 - Fehler/Ausfall von Onboard-Sensoren → Wie verhält sich die Zustandsschätzung?

Systemdynamik-Simulation

- Systemparameter werden von einem Initialisierungsskript in den Workspace geladen
 - Masse, Geometrie, Trägheitsmomente, Übertragungsfunktionen, ...
- Dimension bestimmter Parameter definiert dynamisch Systemgrößen der Simulation
 - Anzahl der Zeilen in der Aktuatormatrix = Anzahl von Rotoren des Flugsystems
- Verschiedene Simulationsmodi können gewählt werden:
 - Rauschen an/aus
 - Auswahl verschiedener Reglertypen

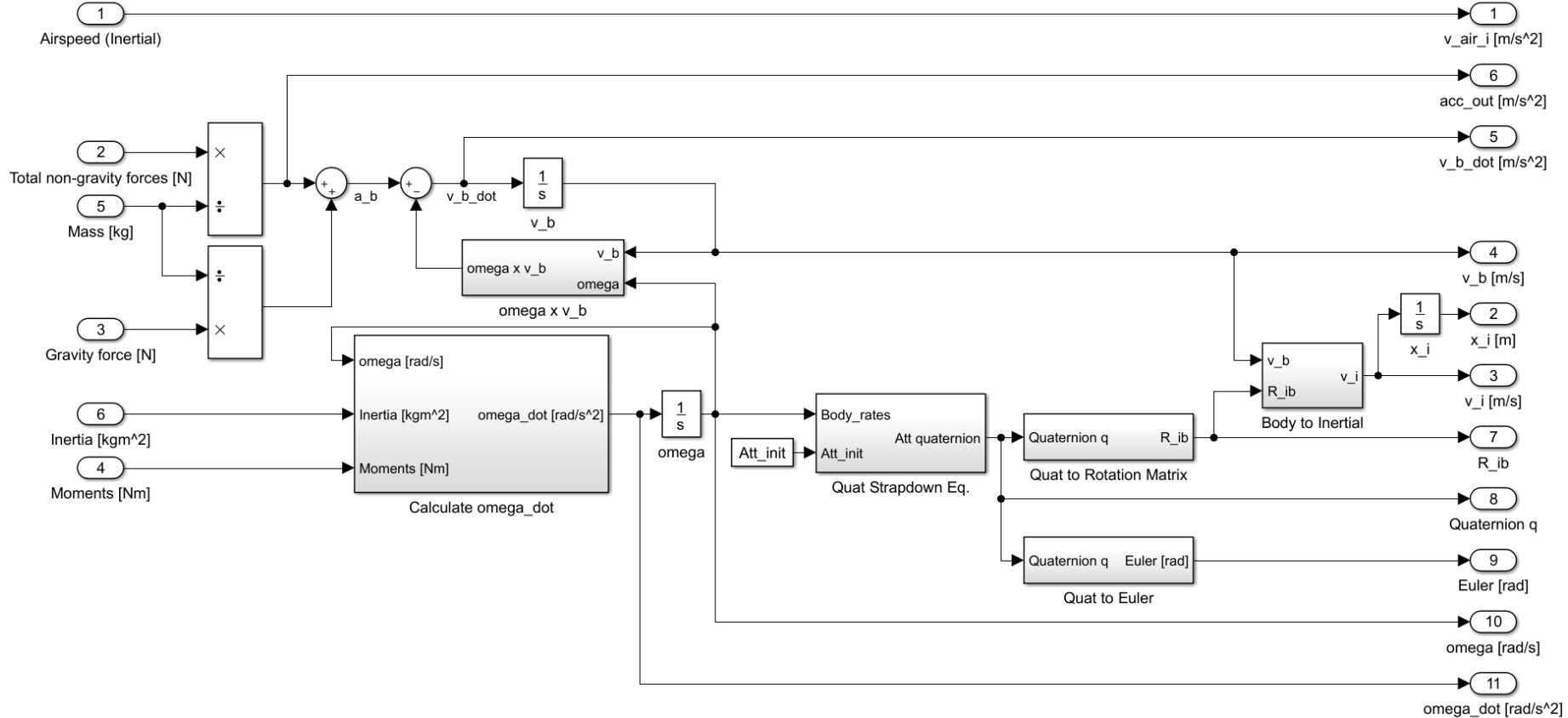
```
*****
% MotorMatrix_real holds all the information about the actual performance
% of each actuator. The data is arranged as one row per motor/prop combo:
% 1      : Motor arm angle measured clockwise (looking from above) from the positive X axis (forward direction) [deg]
% 2      : Distance of prop/motor in X/Y plane from the geometric center of the airframe [m]
% 3      : Distance of prop/motor in Z direction from the geometric center of the airframe [m]
% 4      : Direction of prop rotation: +1 for CW, -1 for CCW [unitless]
% 5      : Control effectiveness of the actuator (nominally 1.0)
% 6      : First-order motor transfer function time constant [sec]
% 7..8    : Two coefficients that describe the RPM to thrust [N] transfer function for static conditions [a1 a2]
% Thrust = a1 * RPM + a2 * RPM^2
% 9..10   : Two coefficients that describe the RPM to torque [Nm] transfer function for static conditions [b1 b2]
% Torque = b1 * RPM + b2 * RPM^2
% 11      : Minimum RPM value of the actuator
% 12      : Maximum RPM value of the actuator
% 13..15  : Rotations of thrust vector around Body-fixed axes away from nominal direction [deg]
%          Nominal direction of thrust vector is [0;0;-1]
%          Rotation order from body to motor axes is Yaw/Pitch/Roll (Euler angle sequence (1,2,3))
% 16      : Propeller diameter [m]
% 17      : Propeller mass [kg]
MotorMatrix_real = [045 , 0.170 , -0.028 , +1 , 1.0 , 0.010 , [9.6820 , 0.010872]*1e-5 , [1.4504 , 0.0016312]*1e-6 , 0000 , 6000 , [0.0 , 0.0 , 0.0] , 8*2.54/100 , 11/1000; ...
135 , 0.170 , -0.028 , -1 , 1.0 , 0.010 , [9.6820 , 0.010872]*1e-5 , [1.4504 , 0.0016312]*1e-6 , 0000 , 6000 , [0.0 , 0.0 , 0.0] , 8*2.54/100 , 11/1000; ...
225 , 0.170 , -0.028 , +1 , 1.0 , 0.010 , [9.6820 , 0.010872]*1e-5 , [1.4504 , 0.0016312]*1e-6 , 0000 , 6000 , [0.0 , 0.0 , 0.0] , 8*2.54/100 , 11/1000; ...
315 , 0.170 , -0.028 , -1 , 1.0 , 0.010 , [9.6820 , 0.010872]*1e-5 , [1.4504 , 0.0016312]*1e-6 , 0000 , 6000 , [0.0 , 0.0 , 0.0] , 8*2.54/100 , 11/1000];
```

Systemdynamik-Simulation



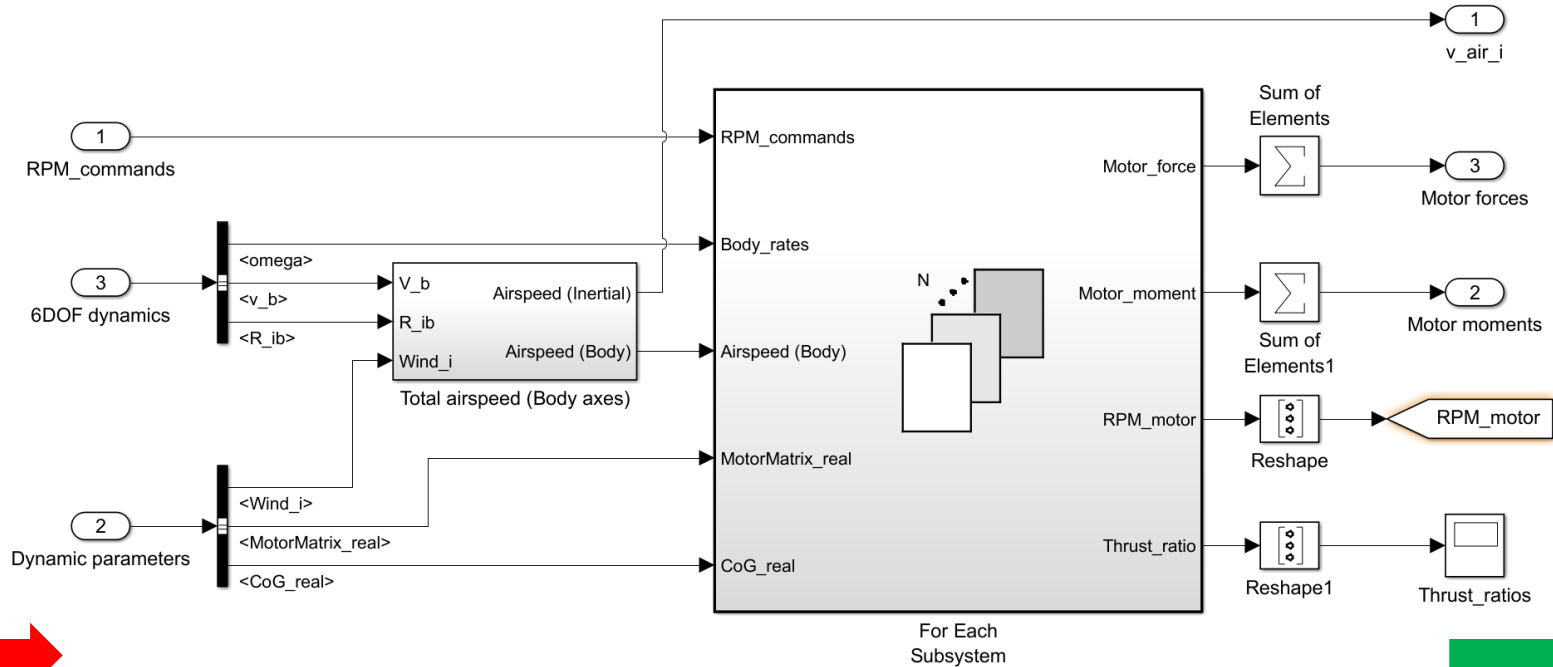
Systemdynamik-Simulation

Implementation of the standard 6DOF equations of motion



Systemdynamik-Simulation

Computation of forces and moments for each actuator specified in the initialization routine



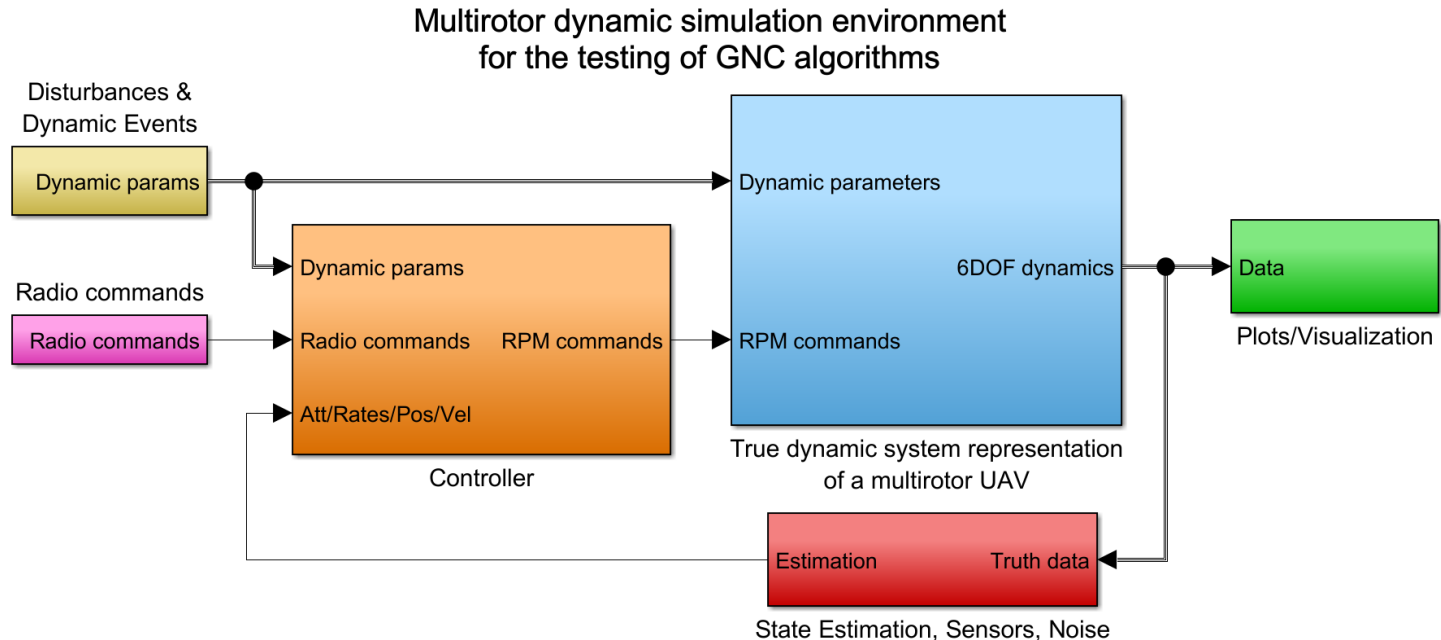
Motor Kommandos
Zustandsvektor
Systemparameter



Motorkräfte
Motormomente
Fluggeschwindigkeit

Guidance/Navigation/Control

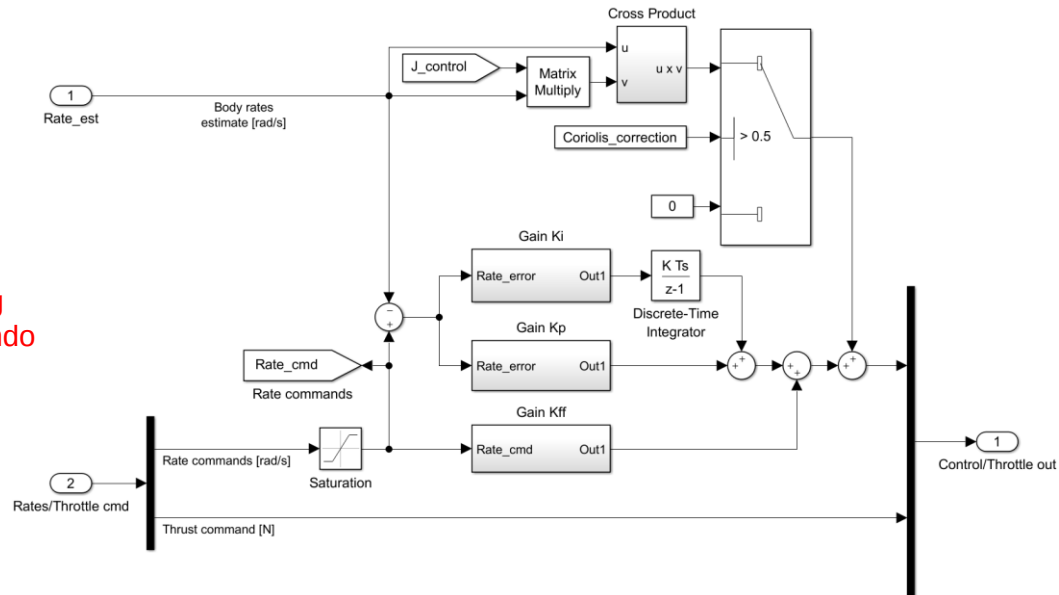
- Neben Systemdynamik sollte das „Gesamtsystem UAV“ simuliert werden können:
 - Guidance: Flugführungsalgorithmen zur Generierung fliegbarer Trajektorien
 - Navigation: Zustandsschätzung der aktuellen Lage/Position/Geschwindigkeit
 - Control: Regelungsalgorithmen auf Drehraten/Lage/Position/Geschwindigkeit



- Das Flugsystem hat verschiedene Sensoren an Bord:
 - Drehraten, Beschleunigung, GPS, Kompass, Luftdruck
- Das Verhalten der Sensoren wird simuliert, in dem man folgende Effekte auf den (idealen) Zustandsvektor wirken lässt:
 - Sensordynamik, Rauschen, Mittelwert, Skalierung, Verkopplung
- Parameter für möglichst realistisches Sensorverhalten auswählen:
 - Analyse von Messdaten echter Sensoren
- So wird aus dem echten Systemzustand (nicht messbar) ein (verfälschter) Messwert
- Diese realistischen Messwerte werden in einem Zustandsschätzer (Kalman-Filter) kombiniert und ergeben den geschätzten Systemzustand:
 - Lage/Position/Geschwindigkeit
 - Mitschätzen von Sensor Biases

Regelkreise

- Je nach Flugmodus wirken unterschiedliche Regelkreise
 - Lage/Höhe/Position/Geschwindigkeit/Trajektorienfolge
- Ziel der Simulation:
 - Test/Auslegung bestehender Algorithmen
 - Schnelle Entwicklung neuer Reglerstrukturen



Drehraten Messung
Drehraten Kommando
Schub Kommando



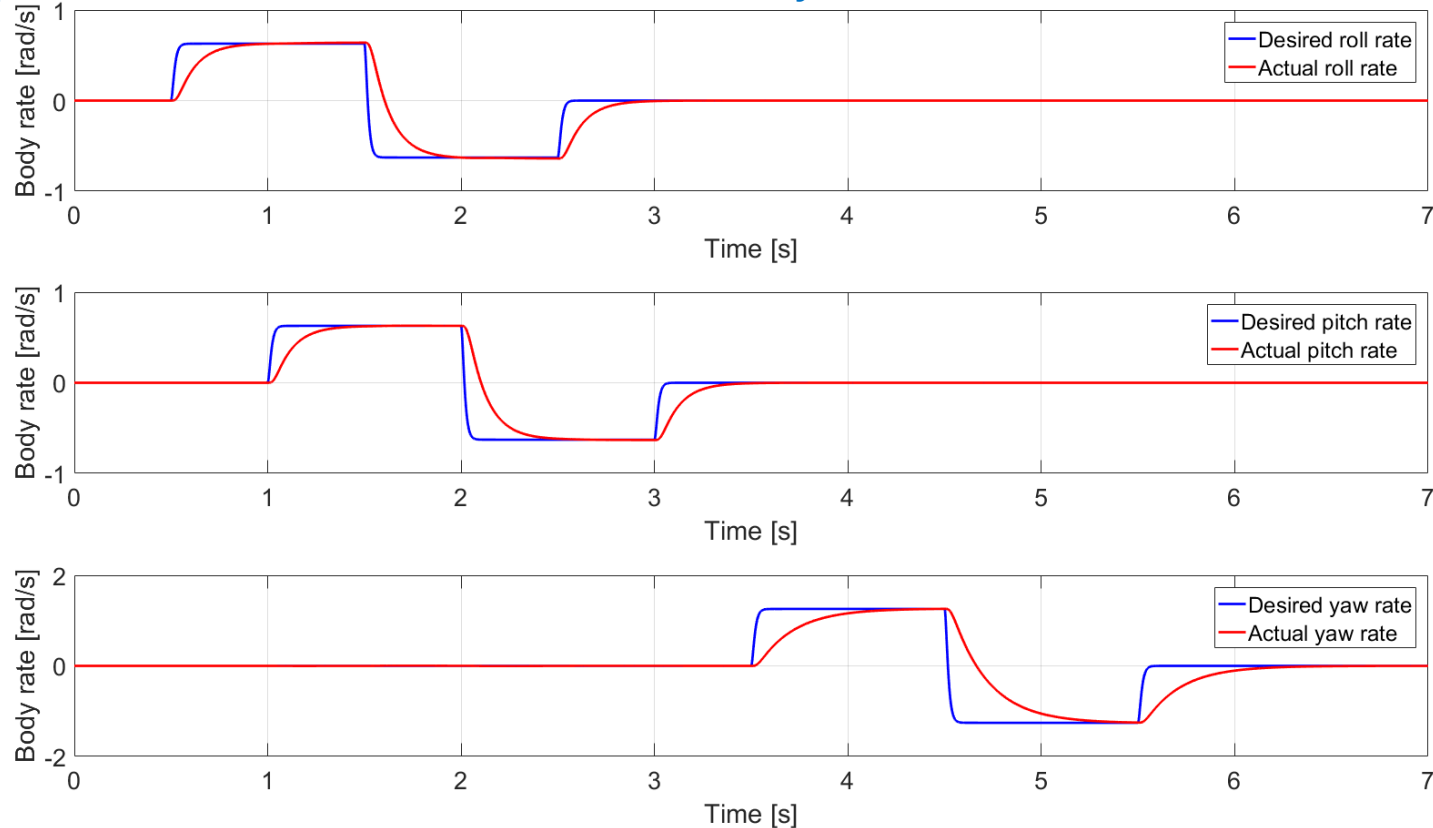
Drehmomente Kommando
Schub Kommando

Entwicklungsprozess

- Im Reglerentwurf sollte schon vor dem Testflug ein gutes Ergebnis vorliegen
 - Regler getunt auf die Parameter des spezifischen Flugmodells
→ Flugsystem verhält sich stabil und vorhersehbar
 - Wie verhält sich der Regler im Bezug auf Umwelteinflüsse?
 - Ist der Regler robust gegenüber Parameteränderungen und dynamischen Events?
 - Wie wirkt sich Messrauschen aus?
 - Sind alle internen Zustände stabil?
 - Testen von Randbedingungen
- Alle diese Antworten müssten ohne modellbasierten Entwurf „erflogen“ werden
- Im Entwurf von Zustandsschätzern können im Testflug aufgezeichnete Messdaten „offline“ durch die Simulation gespielt und verglichen werden.
- Beispiel 1: Drehratenregler mit vereinfachtem/detailliertem Systemmodell
- Beispiel 2: Vollständige Simulation eines Trajektorienregelkreises

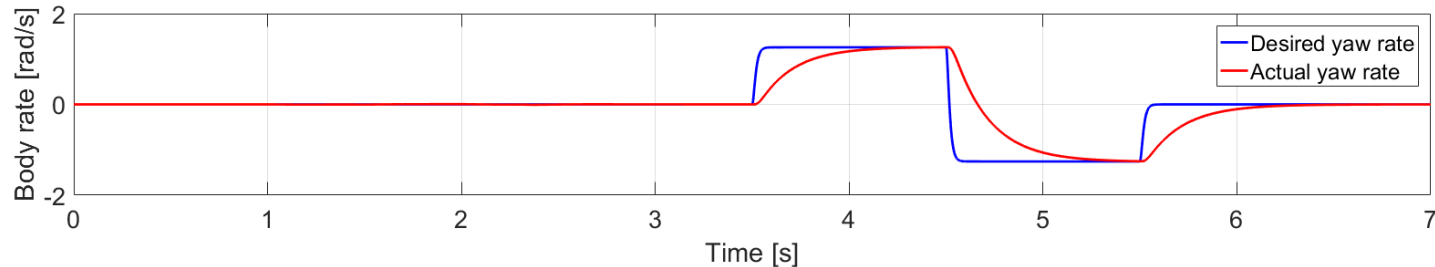
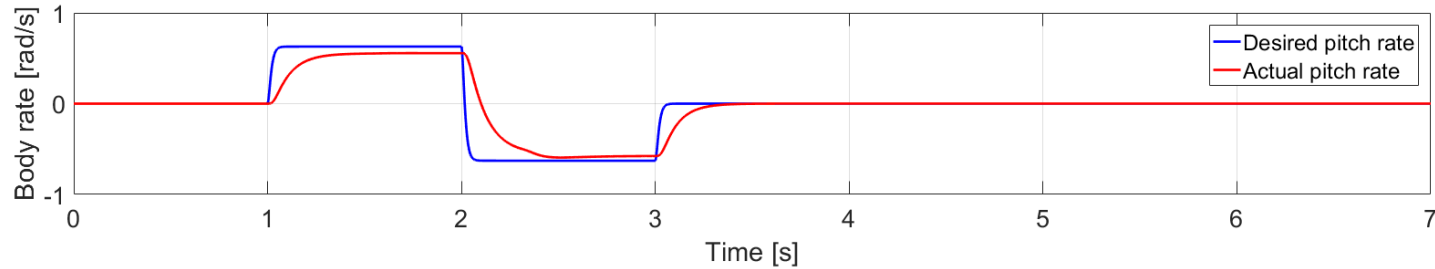
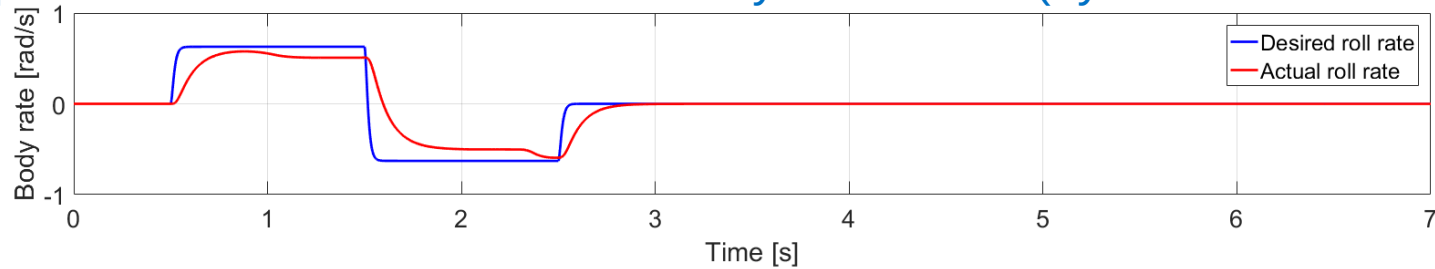
Beispiel 1: Drehratenregler

- Beispiel 1: Simulation mit vereinfachtem Systemmodell



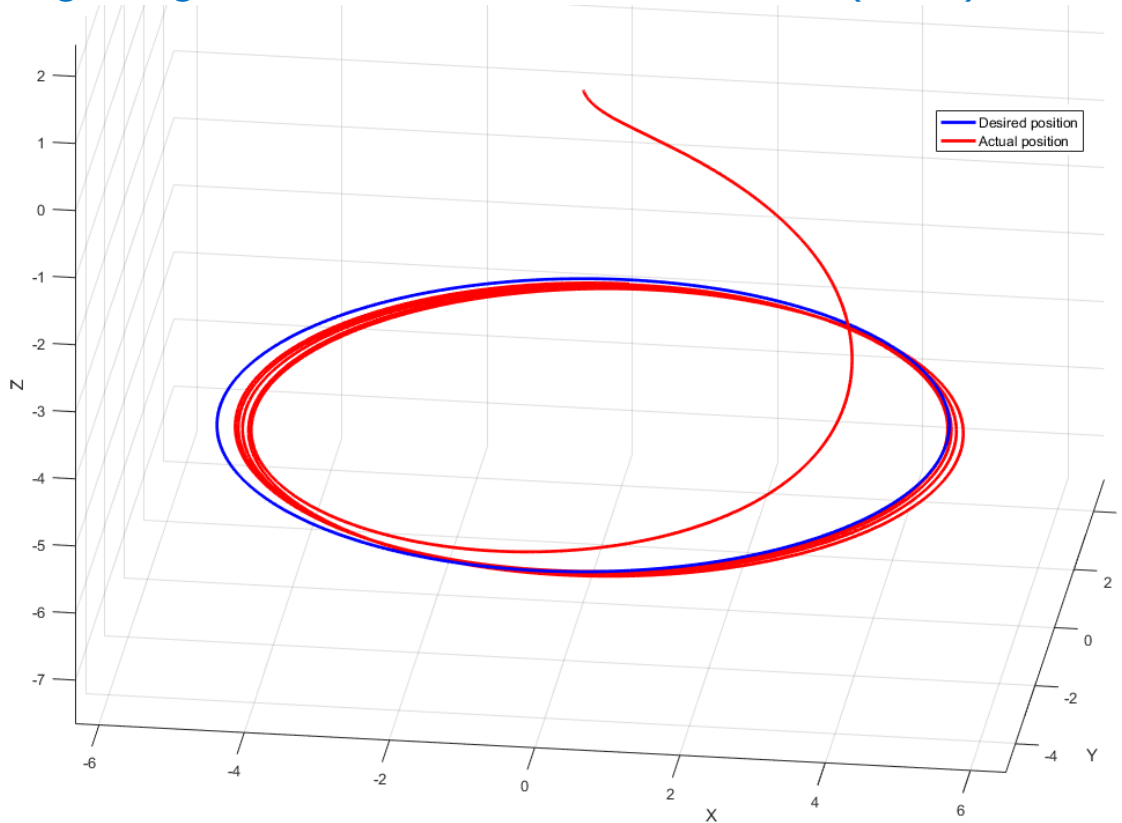
Beispiel 1: Drehratenregler

- Beispiel 1: Simulation mit detailliertem Systemmodell (dynamischer Schub)



Beispiel 2: Trajektorienregler

- Trajektorienregelung auf Kreisbahn mit Seitenwind (5m/s)



Zusammenfassung/Ausblick

- Eine detaillierte Simulationsumgebung mit realistischem Systemverhalten und Umwelteinflüssen steht für schnellen und sicheren Reglerentwurf zur Verfügung
- Die Simulation erlaubt das Tuning und vorläufige Testen neuer Reglerstrukturen
- Weitere Verbesserung der Simulationsmodelle durch Zuhilfenahme von „Ground Truth“-Messungen
- Noch stärkere Einbindung der Simulation in den Entwicklungsprozess:
 - Auto Code Generation: sofortige Generierung operationellen Codes
 - „Hardware/Software in the Loop“ Testing→ Verringerung des Testaufwandes bei gleichzeitiger Erhöhung der Testtiefe
→ Verkürzung von Entwicklungszyklen

Danke für Ihre Aufmerksamkeit!

