

Phase 1: Anwendungsanalyse

Echtzeitsysteme 2 - Vorlesung/Übung

Peter Ulbrich
Wolfgang Schröder-Preikschat

Lehrstuhl für Informatik 4
Verteilte Systeme und Betriebssysteme
Friedrich-Alexander Universität Erlangen-Nürnberg

<http://www4.cs.fau.de/~{ulbrich,wosch}{scheler,wosch}@cs.fau.de>



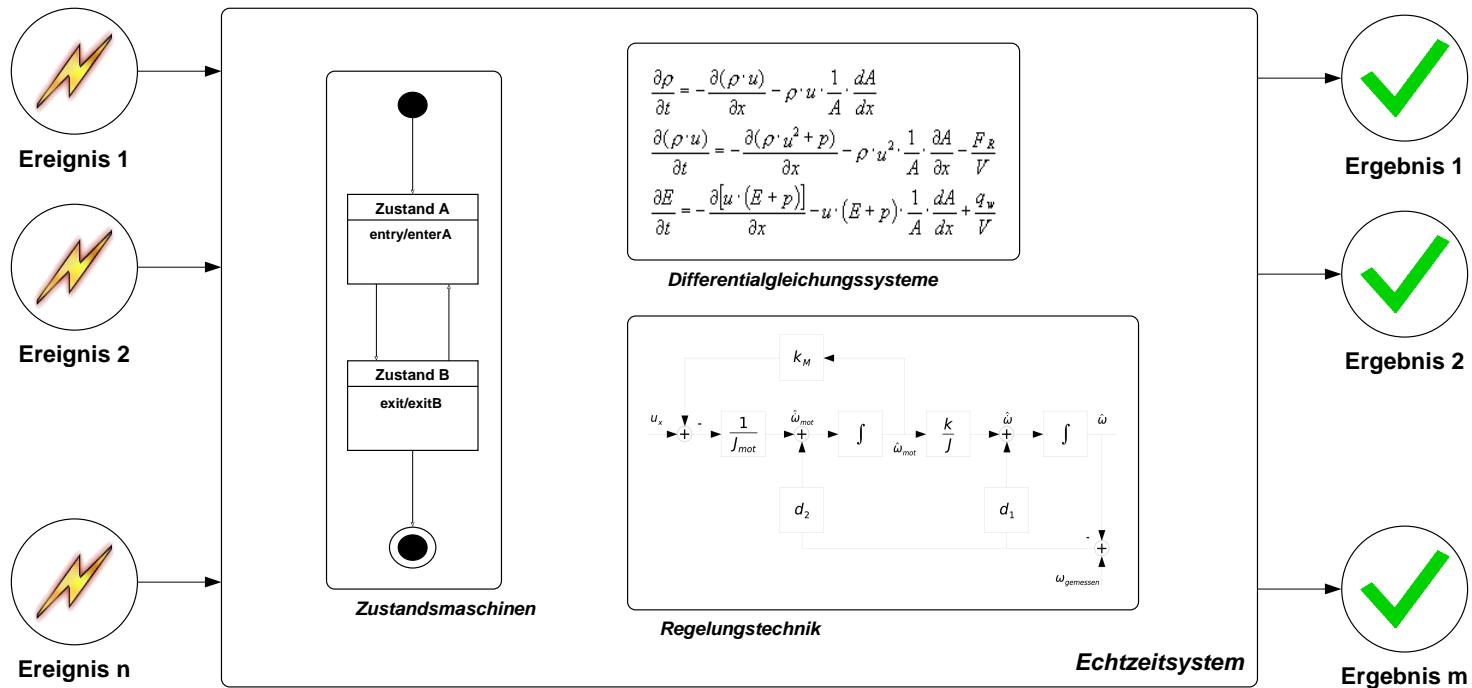
Übersicht

- Einordnung
- Zielsetzung
- Problematik
- Lösungsansätze
- Anforderungen & Fakten



Einordnung – Phase 1

- Beziehungen zwischen **Ereignis n** und **Ergebnis m**
 - zeitlich** – wie viel Zeit darf verstreichen → Termine
 - physikalisch** – wie ist das Ergebnis zu bestimmen?



Zielsetzung

- physikalisches Objekt
 - Welche Größen sind relevant?
 - Wie hängen diese Größen zusammen?
- Echtzeitsystem
 - Welche Ereignisse gilt es zu behandeln?
 - Welche Zeitschranken gilt es einzuhalten?
 - Welche Beziehung Zeitschranke \leftrightarrow physikalisches Objekt gibt es?
- Wie sieht das physikalische Modell aus?
 - Welche Größen des physikalischen Objekts muss man abbilden?
 - Wie bildet man diese Größen ab?



Problematik

- selbst einfach erscheinende Objekte sind aus physikalischer Sicht äußerst komplex
- ➔ Vereinfachungen sind unabdingbar
- Beispiel: Hau den Lukas



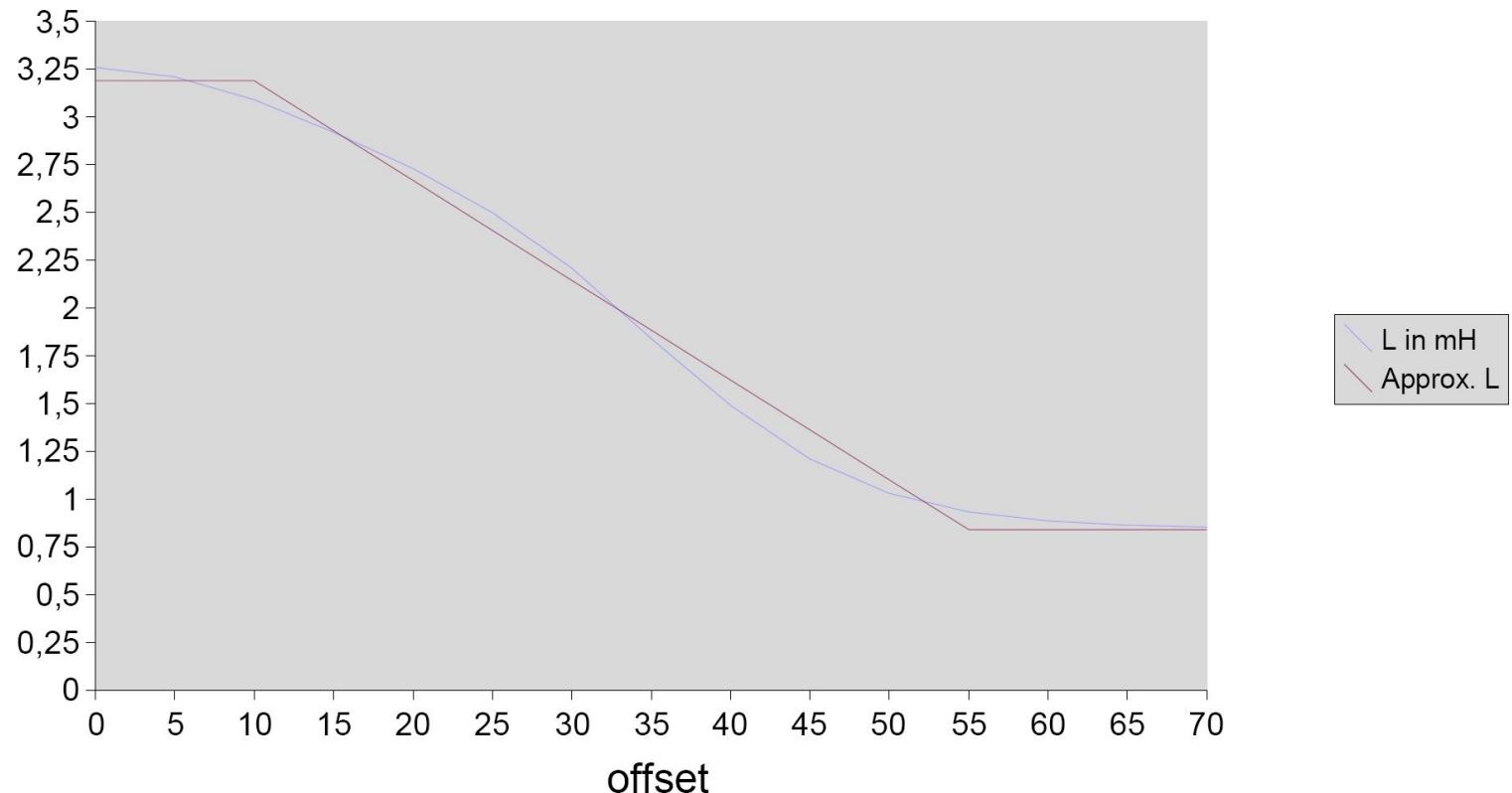
Problematik: Hau den Lukas

- **Schaltvorgänge** an den Spulen sind **nicht konstant!**
 - Ein- und Ausschaltvorgänge und die Induktionsgesetze
 - **Induktivität** der Spule ist **nicht konstant!**
 - Wo ist der Eisenkern in der Spule?
 - **Kraft** auf den Eisenkern in der Spule ist **nicht konstant!**
 - Wo ist der Eisenkern in der Spule?
 - Welche Geschwindigkeit hat der Eisenkern?
 - **Beschleunigung** des Eisenkerns ist **nicht konstant!**
 - Kraft auf den Eisenkern ist nicht konstant
-
- komplexe, zeitabhängige Vorgänge
- macht Integration notwendig
 - häufig nicht mehr analytisch darstellbar
 - numerische Lösung wird erforderlich

Ergebnisse: Hau den Lukas (SS 2006)

- Induktivität der Spule
 - in Abhängigkeit der Position des Eisenkerns

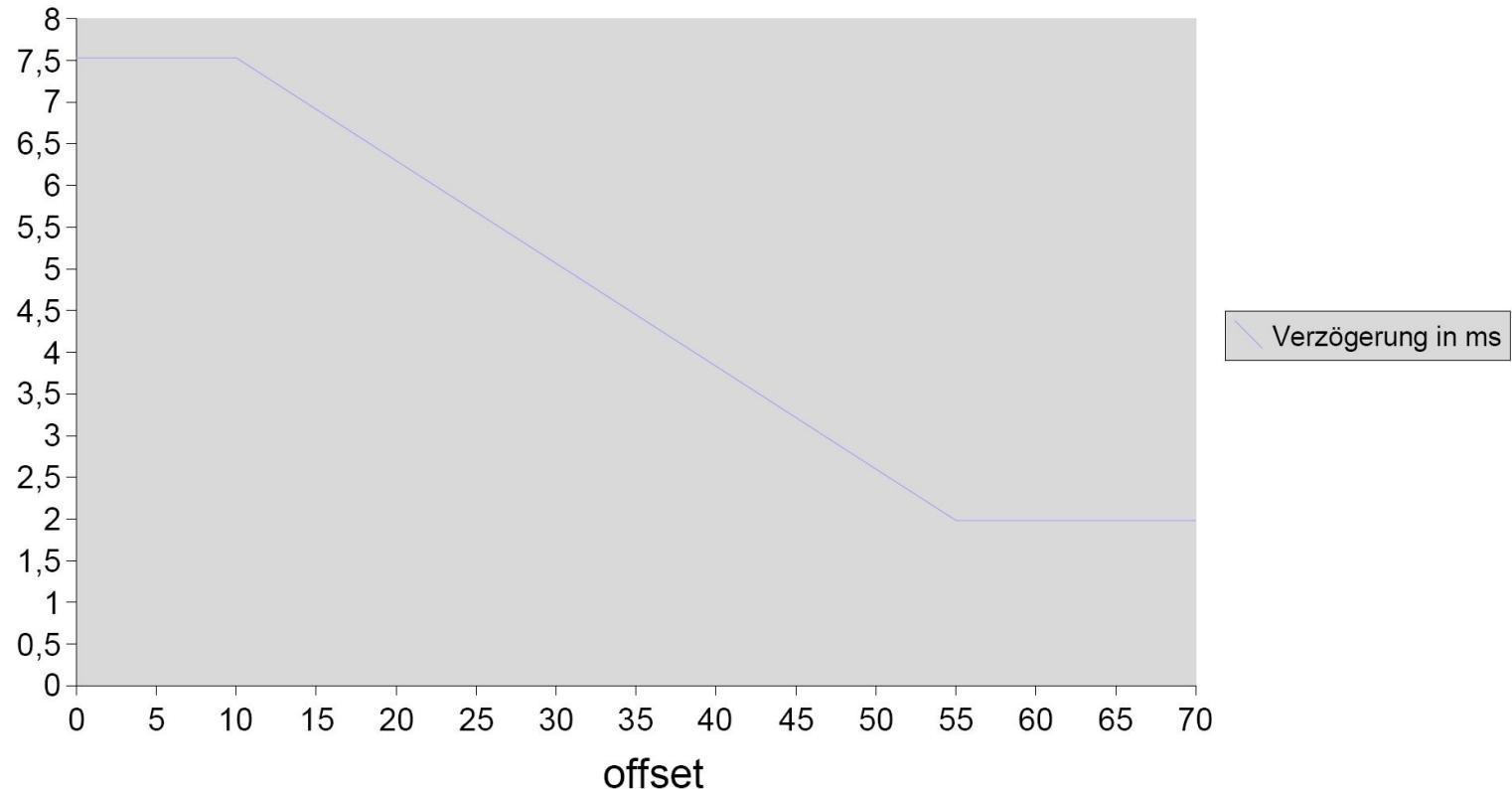
Induktivität



Ergebnisse: Hau den Lukas (SS 2006)

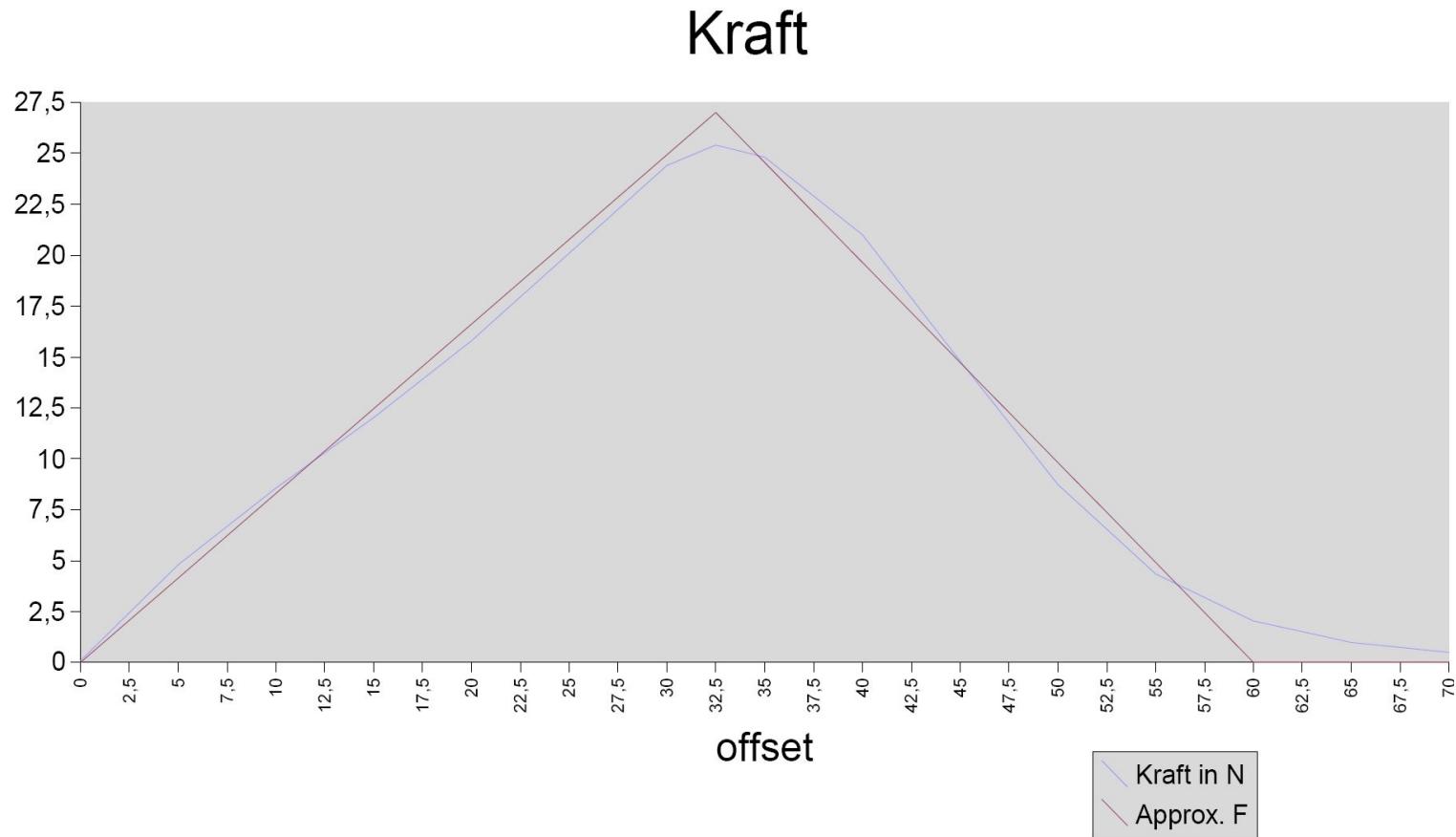
- Schaltverzögerung der Spule
 - in Abhängigkeit der Position des Eisenkerns

Zeitverzögerung



Ergebnisse: Hau den Lukas (SS 2006)

- Kraftwirkung der Spule auf den Eisenkern
 - in Abhängigkeit der Position des Eisenkerns



Ergebnisse: Hau den Lukas (SS 2006)

- selbst das Modell ist bereits stark vereinfacht
- verschiedene Größen werden nicht berücksichtigt
 - Gegeninduktion
 - Wirbelströme
 - Luftwiderstand
- experimentelle Bestimmung gewisser Randparameter



Lösungsansätze

- Reduktion auf den Zustand
- Regelungstechnik

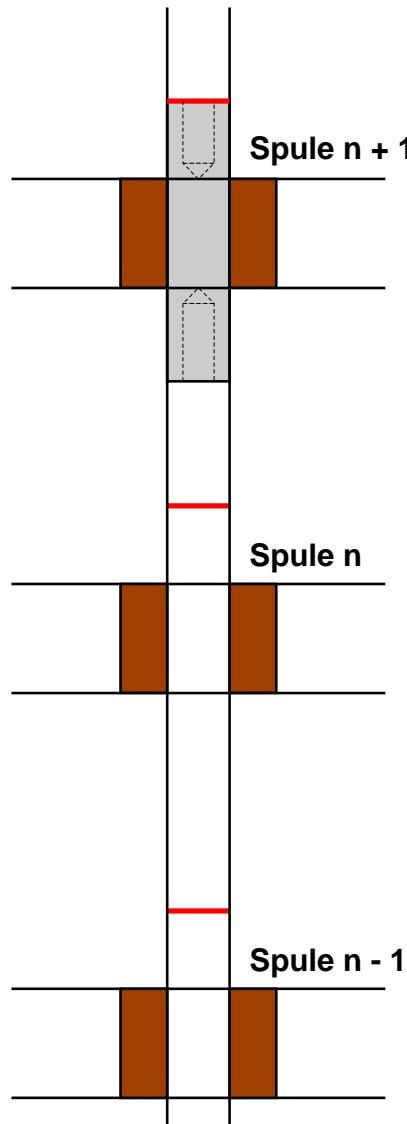


Reduktion auf den Zustand

- ... genauer: den beobachtbaren Zustand
- Idee
 - man kann den Zustand **immer beobachten** ...
 - und man kann ihn **gezielt und exakt manipulieren**
- Konsequenz für unser Modell
 - es kann auf den beobachtbaren Zustand reduziert werden
 - häufig gibt es nur noch diskrete Wertebereiche
 - reine Kausalitäts- also Ursache-Wirkung-Beziehung
- ✓ drastische Vereinfachung des Modells
 - ohne relevante Eigenschaften zu verlieren



Beispiel: Hau den Lukas



■ Beobachtung

- der letzten Spule: durch Lichtschranken
- der Bewegungsrichtung:
 - aktuelle Spule: **Spule n**
 - vorherige Spule: **Spule n + 1** → aufwärts
 - vorherige Spule: **Spule n - 1** → abwärts

■ Manipulation

- Eisenkern kann
 - festgehalten werden
 - fallen gelassen werden
 - angehoben werden

✓ Zustand ist

- vollständig beobachtbar und
- gezielt manipulierbar

Nachteil

- dieses Modell sagt nichts aus über ...
 - den Eisenkern
 - die verwendeten Spulen
 - die Umgebungstemperaturen
- ➔ Eine gezielte Manipulation kann nicht garantiert werden
 - wenn diese Größen verändert werden
- ➔ System muss sich *gutmütig* verhalten
 - gegenüber Parametern, die man nicht kontrollieren kann
 - diese Parameter kann man dann vernachlässigen
- ✗ Längst nicht alle Systeme erfüllen diese Eigenschaft



Regelungstechnik

■ Problem

- interner Zustand ist **nicht beobachtbar**
- **interne Parameter** beeinflussen das System in relevantem Umfang

■ Idee

- **Nachbildung/Berechnung des Zustandes**
 - physikalisches Modell
 - inkl. internem Verhalten

→ Konsequenz für unser Modell

- mathematisch/physikalische Beschreibung des Systems
- Bestimmung der Systemparameter: Trägheit, Widerstand, ...
- Berücksichtigung vergangener Zustände

✓ Beschreibung des Systems durch das Modell

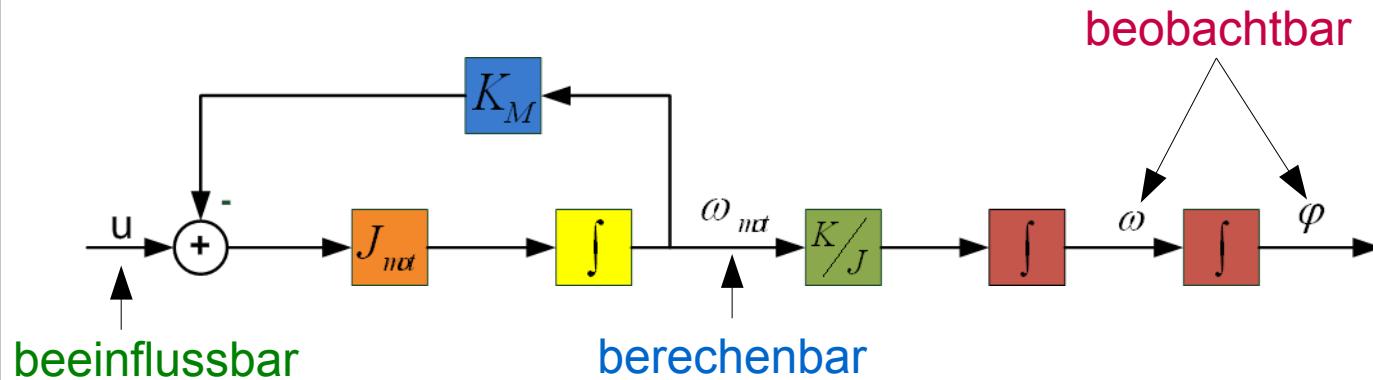
- exakte Analyse notwendig



Regelungstechnik

■ Systemtheorie

- Eingangsvektor → Abweichung vom Sollwert des Systems
- Ausgangsvektor → Ist-Wert des Systems
- Systemmatrix → Rückkopplung der internen Zustände



■ System-Modell

- Beschreibt die Zusammenhänge zwischen **beeinflussbaren**, **beobachtbaren** und **berechenbaren** Systemparametern



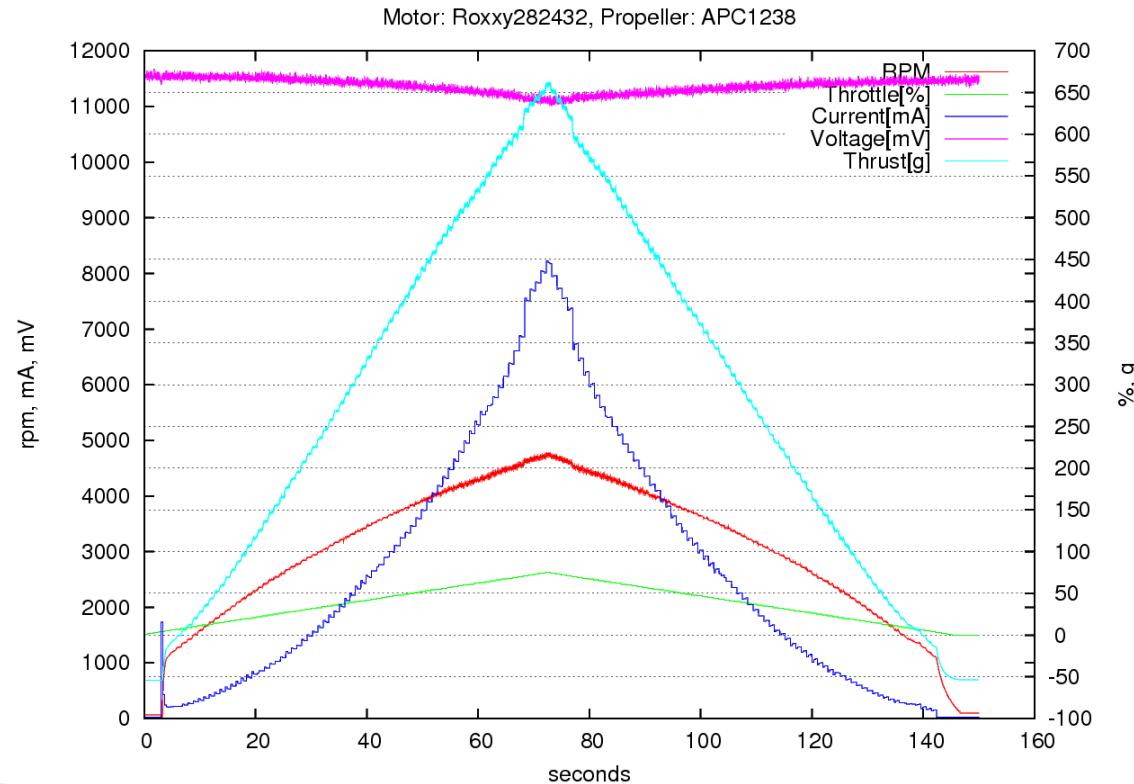
Beispiel: Quadrocopter

- Beobachtung
 - Winkelgeschwindigkeit ω und Lage φ um X bzw. Y-Achse
- Manipulation
 - erzeugte Schubkraft kann variiert werden
 - geregelt wird die **Spannung** u der Motoren
- Reaktion
 - abhängig von den Momenten des Objekts (**Masse**, **Trägheit**)
 - und Motor/Propeller (**Trägheit**, **Reibung**, **Wirkungsgrad**)
- ✓ Zustand
 - nicht beobachtbar, aber berechenbar



Modellbildung

- Berechnung beliebiger Zustände
- Bestimmung durch Messung
 - Erstellen von Kennlinien (Messung zusätzlicher Parameter)
 - Ableiten von Konstanten oder Funktionen
- z.B. Motor
 - Messung
 - Schubkraft
 - Drehzahl
 - Stromaufnahme
 - Spannung
 - Motorkonstante
 - Funktion
 - Spannung/Schub



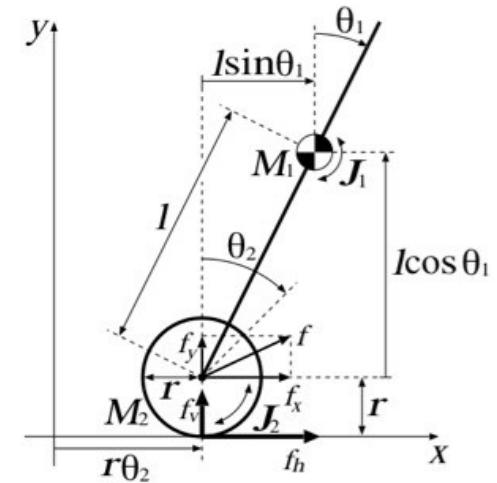
Anforderungen & Fakten

- NXT Standalone
- Hau den Lukas
- Generator
- *I4Copter*



Anforderung: NXT Standalone

- **Zielsetzung:** Stabiler Stand
 - Zusätzlich: **Fahren & Ausweichen**
 - Optional: **Fernsteuerung**
 - Variante: Robot on a ball
- **Reglerimplementierung**
 - Regelgröße:
 - **Motorspannung** → Winkel (Rad) → Winkel (Standalone)
 - Regelziele:
 - **Stabilisierung** (Winkel)
 - **Positionsbestimmung** (Weg)
 - PID-Regler und Sensorfusion:
 - Gyroskop und Winkelencoder



Fakten: NXT Standalone (1)

■ NXT Block

- Hauptprozessor: **32-bit ARM7** (AT91SAM7S256)
 - 256 KB FLASH, 64 KB RAM, 48 MHz
- Co-Prozessor: 8-bit AVR ATmega48
 - 4 KB FLASH, 512 Byte RAM, 8 Mhz

■ Schnittstellen

- **4 Eingänge** (analog/digital)
- **3 Ausgänge** (Rückkanal)
- 4 Druckknöpfe

■ Kommunikation

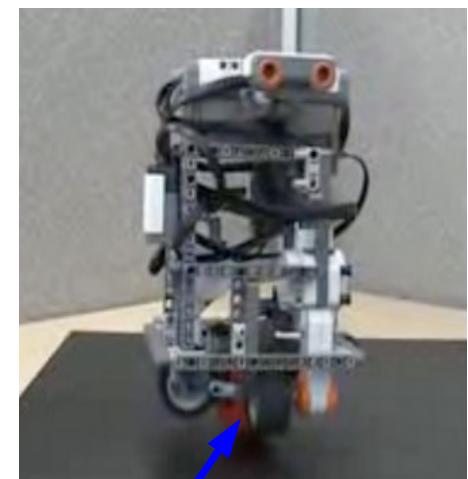
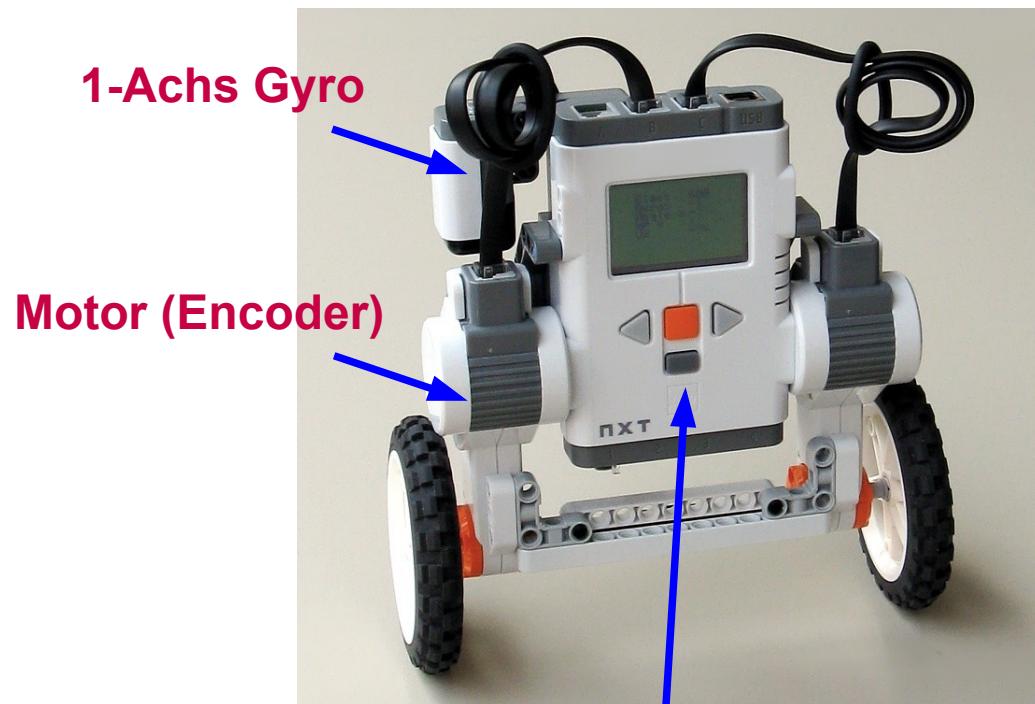
- **Bluetooth** CSR BlueCoreTM 4 v2.0 +EDR System
- USB 2.0 Anschluss



Fakten: NXT Standalone (2)

■ Varianten:

Ultraschall Abstandssensor



Anforderungen: Hau den Lukas (1)

■ Programmierbarkeit

- ein Programm wird durch eine Datenstruktur beschrieben, die durch Prozeduraufrufe erzeugt/initialisiert wird

```
lift_continuous(my_prog, 7);  
fall_stepwise(my_prog, 2);  
lift_continuous(my_prog, 4);  
...
```

- schließlich wird dieses Programm ausgeführt

```
start_program(my_prog);
```

■ folgende Primitive sollen beherrscht werden

- schrittweise anheben/fallen lassen
- kontinuierlich anheben/fallen lassen

■ Primitive sind kombinierbar

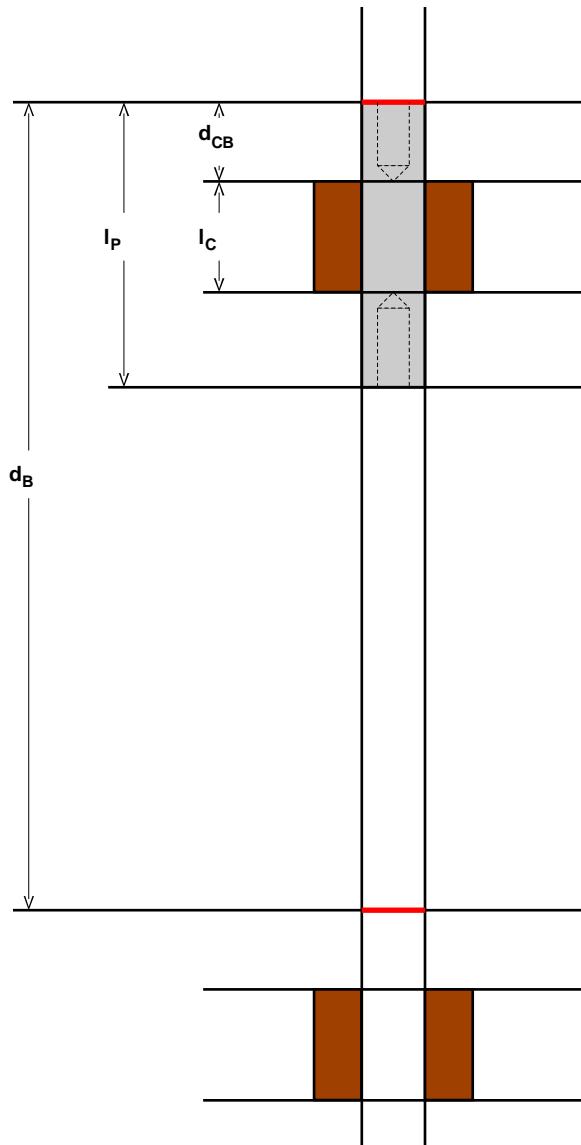


Anforderungen: Hau den Lukas (2)

- Benutzerschnittstelle
 - Programme werden unveränderlich gespeichert
 - ein Wechsel zwischen den Programmen ist aber möglich
 - Programme werden geladen
 - Interaktion über die serielle Schnittstelle bzw. CAN-Bus
- Not-Aus-Funktion
 - sofortiges Stoppen der normalen Funktion
 - Übergang in einen sicheren Zustand
- Monitoring/Überwachung
 - kritische Systemzustände werden erkannt und angezeigt
 - z.B. wenn Spulen zu lange aktiviert sind
 - Profiling
 - wie lange ist welche Spule an, wie oft wurde welche Spule durchlaufen

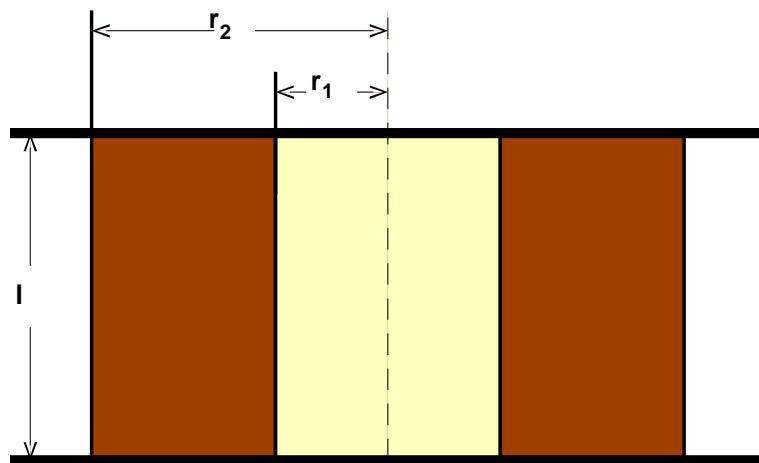
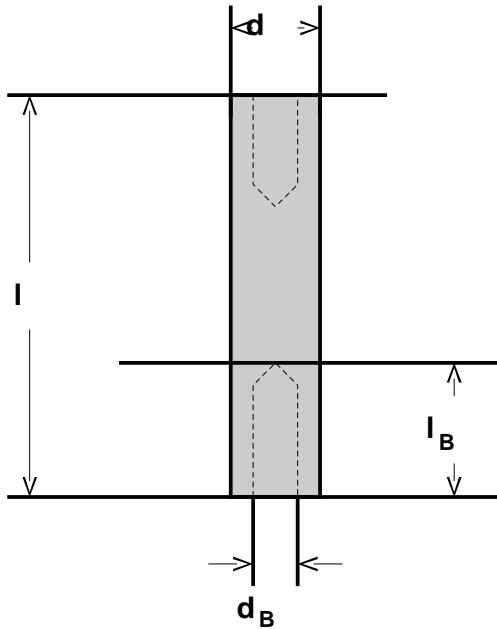


Fakten: Hau den Lukas



l_c	Länge: Spule	30 mm
l_p	Länge: Projektil	82 mm
d_B	Abstand: Lichtschranken	230 mm
d_{CB}	Abstand: Spule – Lichtschranke	24 mm

Fakten: Hau den Lukas



<i>l</i>	Länge: Projektil	82 mm
<i>l_B</i>	Länge: Bohrung	26 mm
<i>d</i>	Durchmesser: Projektil	18,6 mm
<i>d_B</i>	Durchmesser: Bohrung	10 mm
<i>m</i>	Gewicht	144 g
	Material: Eisen	

<i>l</i>	effektive Länge	23 mm
<i>r₁</i>	Innendurchmesse	28 mm
<i>r₂</i>	Außendurchmesse	mm
<i>l</i>	Induktivität	0,82 mH
<i>R</i>	Widerstand	0,52 Ohm
<i>N</i>	Windungszahl	178
	Toleranz	3.00%

Fakten: Hau den Lukas

- Ansteuerung der Spulen
 - Pin 0.5 – Pin 0.12 (x.y: x – Port, y – Pin)
- Abfrage der Lichtschranken
 - Pin 7.0 – Pin 7.7
- Unterbrechungen der Lichtschranken
 - Pin 1.0
- Dauer von Auslösung der Lichtschranke bis zur ISR
→ vernachlässigbar



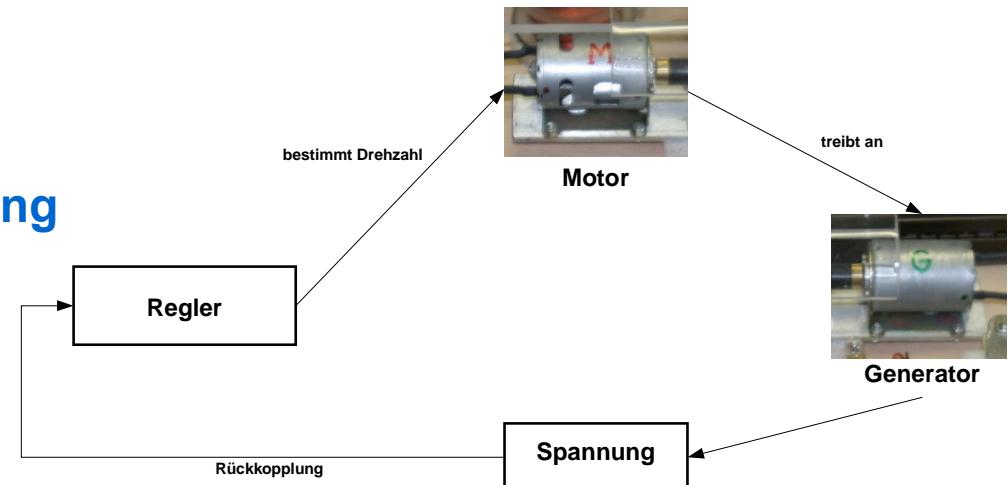
Anforderung: Generator

■ Regelkreisimplementierung

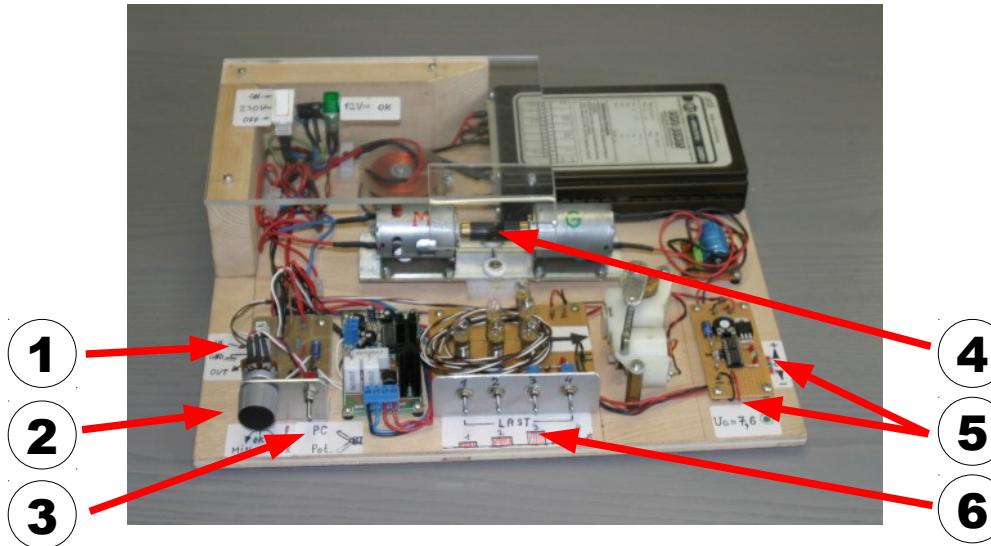
- *naiver* Regler
 - Hochfrequente Abtastung des Systems
 - kleine Änderungen der Stellgrößen
- Reglerimplementierung
 - mit/ohne Beobachter
- Bestimmung der Kennlinien für Motor und Generator
- Evaluation: Einschwingverhalten

■ Leitwarte

- Bedienpult
- **Steuerung & Überwachung**



Fakten: Generator



1. Steuerspannung des Motors
 - kontinuierlich (PWM-Signal)
 - diskret (256 Stufen, extra Platine)
2. Manuelle Steuerung
 - Durch Potentiometer
3. Umschaltung
 - Steuerspannung/manuell
4. Motordrehzahl
 - Messung durch Lichtschranke
5. Referenzspannung (7,8 V)
 - optisch ablesbar
 - Abgreifbar
6. Zuschaltung div. Lasten
 - konstant
 - variabel



Fakten: Generator

1. Einstellung der Referenzspannung

- Manuell oder über Bedienpult

2. „Kalibrierung“ des Reglers

3. Umschalten auf die Steuerspannung

- Und damit auf den implementierten Regler

4. Rückmeldung an die Leitwarte

- Überwachung und Kontrolle



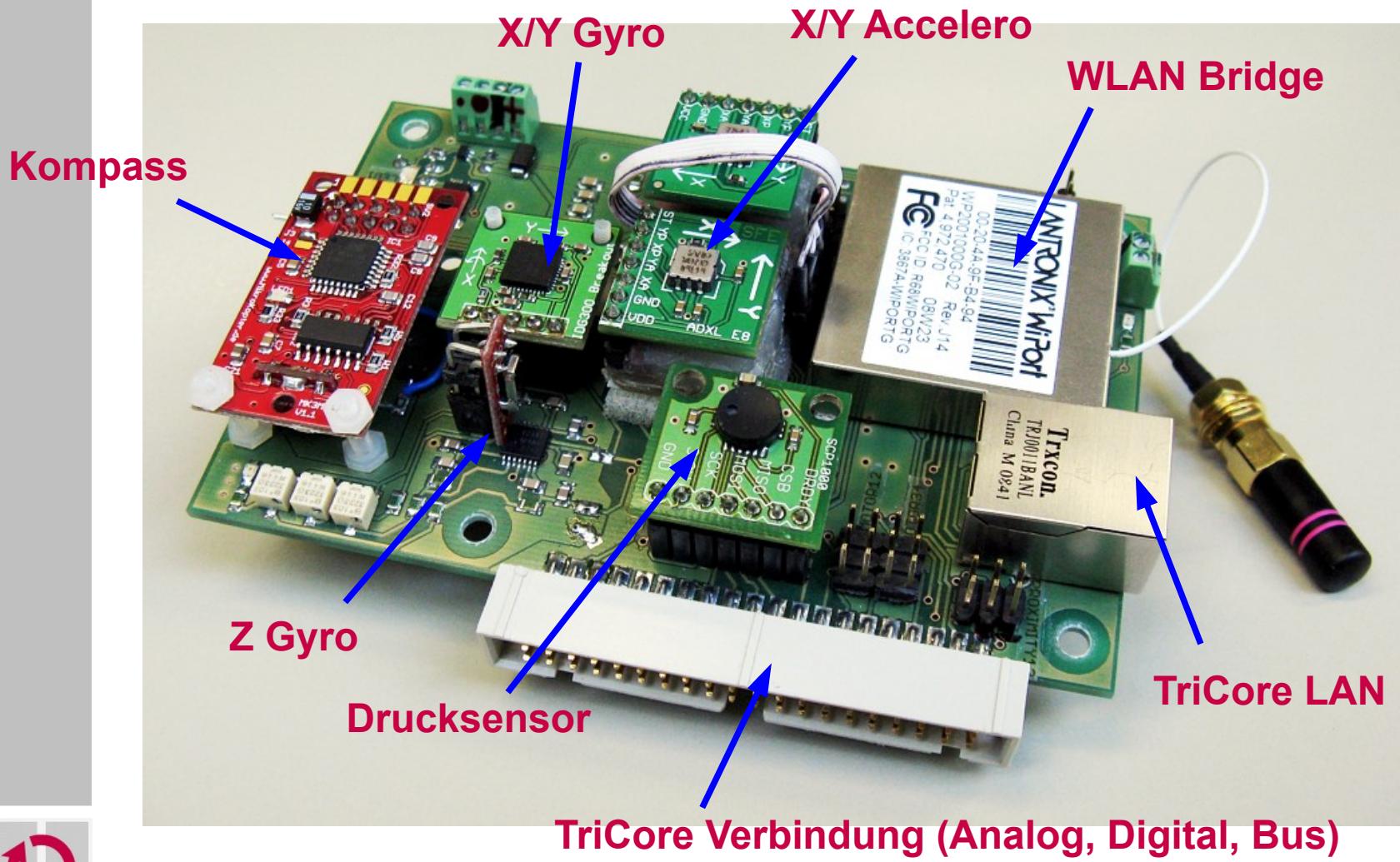
Anforderung: I4Copter

- **Zielsetzung**: semi-autonomer Flug
 - Stabiler **Schwebeflug**
 - Unterstützung durch automatisches **Starten & Landen**
 - Anfliegen von Wegpunkten - **Autopilot**
- **Komponentenüberwachung und Verhaltensmodell**
 - **Komponentenzustand** und **-überwachung** (Init, Active, Error)
 - Analyse und Spezifikation von Arbeitspunkten
 - Übergeordnetes **Verhaltensmodell** (z.B. Zustandsautomat)
- **Höhen-Regler**
 - Bestimmung der relativen Höhe über Ultraschall-Abstandssensor
 - und der absoluten Höhe über Druck-Sensoren
 - Unterschiedliche **Arbeitspunkte** (Start / Landung; Flug)



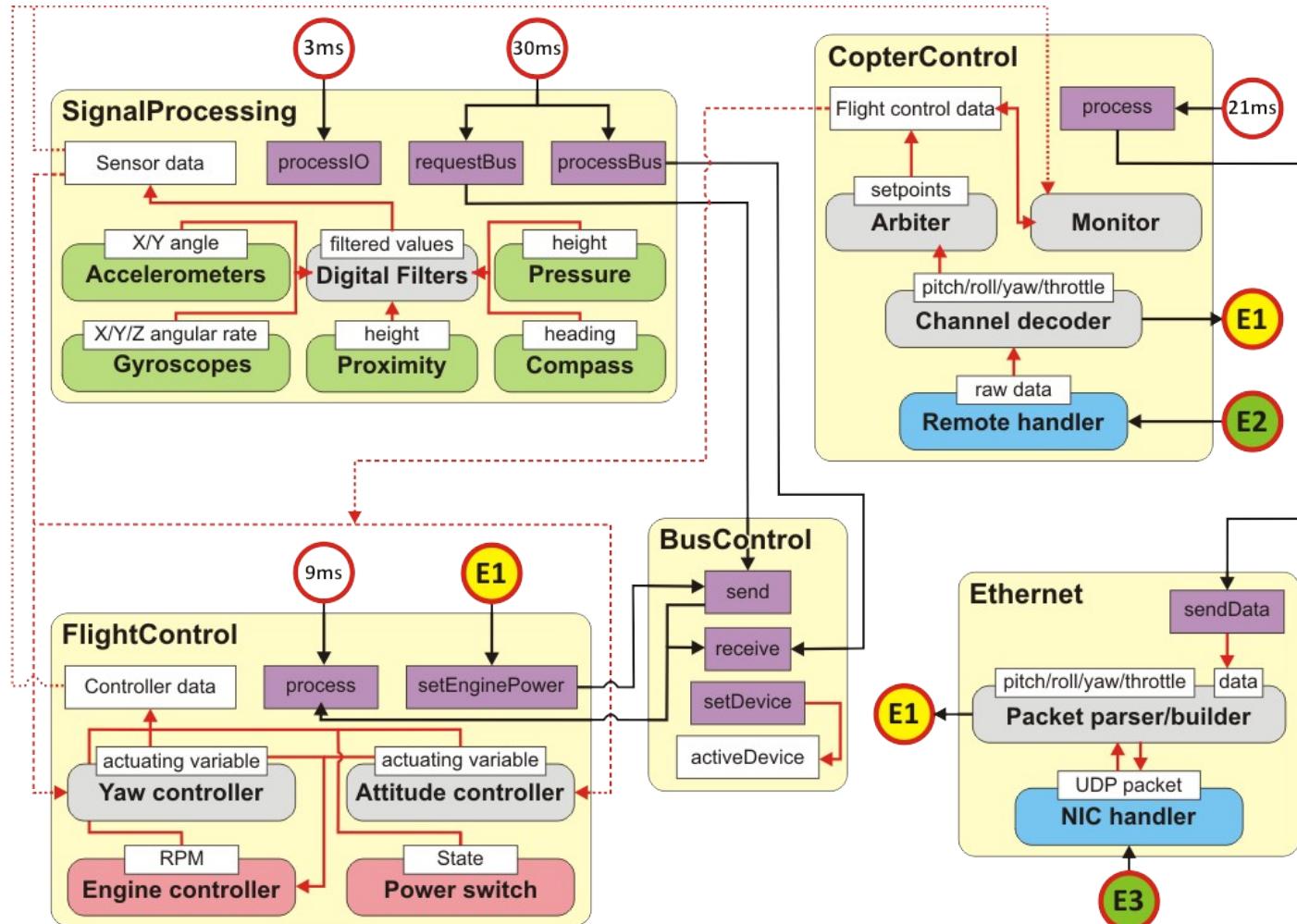
Fakten: I4Copter (1)

I4Copter Periphery Board Mark2



Fakten: I4Copter (2)

I4Copter Software Struktur



Fazit

- scheinbare Vorgänge sind physikalisch sehr komplex
- komplexe physikalische Vorgänge sind mathematisch häufig nicht analytisch lösbar
 - Vereinfachungen des Modells
 - numerische Lösungsansätze
- Massive Vereinfachungen sind notwendig
 - Was muss man wirklich wissen?
 - Was kann man wissen/messen?
 - Welche Einschränkungen sind damit verbunden?



Fazit

- scheinbare Vorgänge sind physikalisch sehr komplex
- komplexe physikalische Vorgänge sind mathematisch häufig nicht analytisch lösbar
 - Vereinfachungen des Modells
 - numerische Lösungsansätze
- Massive Vereinfachungen sind notwendig
 - Was muss man wirklich wissen?
 - Was kann man wissen/messen?
 - Welche Einschränkungen sind damit verbunden?

**Entwurf des Echtzeitssystems setzt Vertrautheit
mit dem physikalischen Objekt voraus!**



Ergebnis

- Welche Aktivitäten laufen in dem System ab?
 - Können diese Aktivitäten feiner strukturiert werden?
 - Laufen Elemente einer Aktivität zeitgleich ab?
 - Das sind die **Aufgaben!**
- Wann werden diese Aktivitäten ausgeführt?
 - Welche zeitlichen Eigenschaften haben diese Zeitpunkte?
 - Das sind die **Ereignisse!**
- Was hängt von den berechneten Ergebnissen ab?
 - Wie viel Zeit kann dabei verstreichen?
 - Das sind die **Termine!**

