

Anwendungsanalyse von Echtzeitsystemen

Vorgehen in der Praxis

Florian Franzmann, Martin Hoffmann, Isabella Stilkerich

Friedrich-Alexander-Universität Erlangen-Nürnberg
Lehrstuhl Informatik 4 (Verteilte Systeme und Betriebssysteme)
www4.informatik.uni-erlangen.de

26. November 2012

- 1 Organisatorisches
 - Hinweise zu Aufgabe 3
 - Software Tracing
- 2 Anwendungsanalyse
- 3 Rekapitulation der Vorlesung

Aufgabe 3 - Hinweise

Wichtige Hinweise

- **Ausgabe 26.11. – Abgabe bis Donnerstag, 06.12.!**
- Erste Übung mit **zeit-** und **ereignisgesteuerter** Varianten $\leadsto$
Aufteilung der Vorgabe:

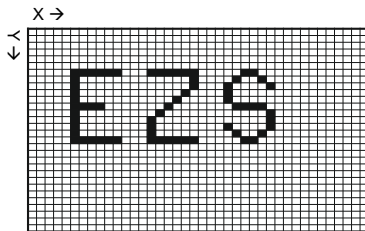
```
<Aufgabe3>/  
event-triggered/ /* eCos */  
time-triggered/ /* tt-eCos */
```

- **libEVS** wird realistischer $\leadsto$ Zufallszahl nötig:

```
#include <stdlib.h>  
  
cyg_uint32 random_number = rand(); /* Ergebnis 0-RAND_MAX */  
cyg_uint32 scaled = random_number % 10; /* Zufallszahl 0-10 */
```

Grafische Ausgabe – Das Display

Framebuffer Grundlagen



- **Framebuffer** $\leadsto$ Grafische Ausgabe
- eCos bietet einheitliche, hardwareunabhängig Schnittstelle¹
- Auflösung in unserem Fall: **320x240x16 bit** (qemu & Hardware)
 - Zwecks Portabilität immer Variablen nutzen:
 - Breite: `CYG_FB_WIDTH (FRAMEBUF)`
 - Höhe: `CYG_FB_HEIGHT (FRAMEBUF)`

¹ ecos.sourceware.org/docs-latest/ref/io-framebuf.html

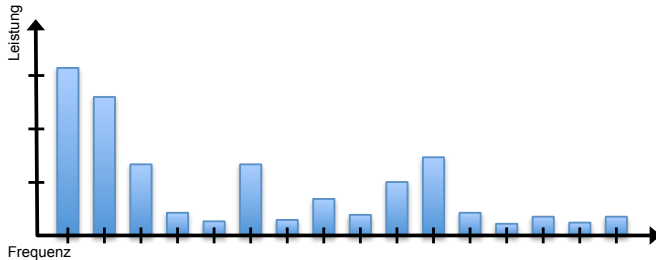
Leistungsdichtespektrum in libEVS

```
void ezs_power_density_spectrum(float in[], float out[], int N)
```

- `in[]` Eingabe, Abschnitt des Zeitbereichssignals
- `out[]` Ausgabe, Leistungsdichtespektrum (LDS)
- `N` Anzahl der Abtastwerte, wobei N Zweierpotenz
- Zeitbereichssignal der Länge $N \mapsto \text{LDS}^2$ der Länge $\frac{N}{2}$
- Höchste Frequenz im Spektrum aus Abtasttheorem
 $\Rightarrow \frac{f_{\text{Abtast}}}{2}$

²http://en.wikipedia.org/wiki/Spectral_density

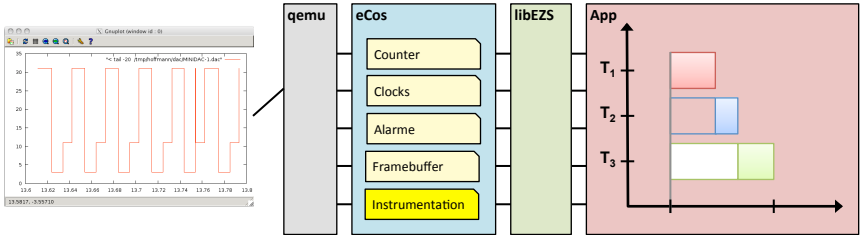
LDS – Beispiel



- Darstellung mit 16 Werten:
⇒ LDS der Länge 32 notwendig
 - 1Hz Abstand $\leadsto$ Spektrum bis 16 Hz erfassen
⇒ Abtastfrequenz 32 Hz
- ⇒ Balken repräsentieren Leistung

Software Tracing

Darstellung mittels Ablaufgraph



- **Visualisierung der Threads** $\leadsto$ Softwarebasiertes Tracing
- eCos Instrumentations³
- Ausgabe der Priorität $\leadsto$ **Ablaufgraph**

³

ecos.sourceware.org/docs-latest/user-guide/kernel-instrumentation.html

Anforderungs- und Anwendungsanalyse

Praktischer Entwurf von Echtzeitsystemen

Externer Foliensatz

Rekapitulation der Vorlesung

Kapitel 4-1 und 4-2

Diskussion

Anwendungsanalyse

Echtzeitsysteme - Übung

Martin Hoffmann, Peter Ulbrich

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

[http://www4.cs.fau.de/~{ulbrich,wosch}](http://www4.cs.fau.de/~{ulbrich,wosch}{scheler,wosch}@cs.fau.de)
{scheler,wosch}@cs.fau.de



Übersicht

- **Prequel: Anforderungsanalyse**
- **Anwendungsanalyse**
 - Einordnung
 - Zielsetzung
 - Problematik
 - Lösungsansätze
 - Anforderungen & Fakten



Analyse der Problemstellung

- methodisch gestütztes Aufstellen von Anforderungen
- **Anforderung** (engl. *requirements*)
 - Aussage über eine zu erbringende Leistung
 - eines Produkts oder eines Systems
 - eine Eigenschaft, die erfüllt sein muss,
 - damit ein bestimmter Vorgang gelingen kann
 - ein Leistungsmerkmal (nicht nur) von Software
- Zusammenfassung im **Lasten-/Pflichtenheft**
 - als Bestandteil eines zu erstellenden Anforderungsdokuments, das
 - die durch das System zu lösende Aufgabe beschreibt
 - die im Projekt zu erreichenden Ziele definiert
 - den Benutzerkreis des zu entwickelnden Systems festlegt
 - ... in **Zusammenarbeit mit dem Kunden**



Spezifikationstechniken

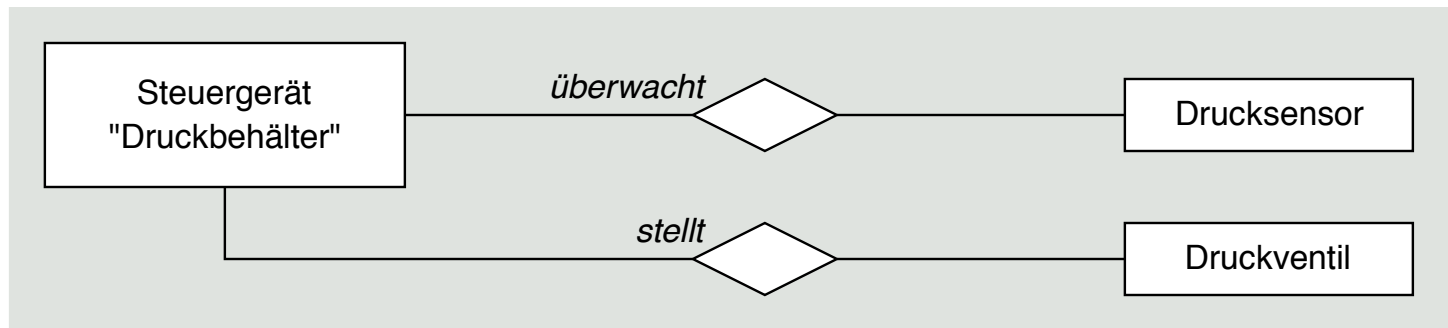
- allgemeine Klassifikation bzw. Ansätze
- **formal** (engl. *formal*)
 - rigorose, mathematische Grundlage → **formale Notation**
- **informell** (engl. *informal*)
 - wenn die **Transkription** („Umkodierung“) in eine formale Notation mit zugeordneten Regeln nur eingeschränkt möglich ist
 - z.B. ein Ablaufdiagramm (engl. *flowchart*)
 - bestenfalls werden Anforderungsverletzungen/-konflikte sichtbar
- **halbförmlich** (engl. *semiformal*)
 - Ansätze, die formale und informelle Züge zeigen, z.B. UML:
 - das Zustandsdiagramm (engl. *statechart*) ist formal
 - andere Konzepte sind jedoch eher pseudomathematischer Natur

Echtzeitsysteme (mit strikt einzuhaltenden Anforderungen) **erfordern** eine **formale Begründung** der Leistungscharakteristiken von Anforderungen



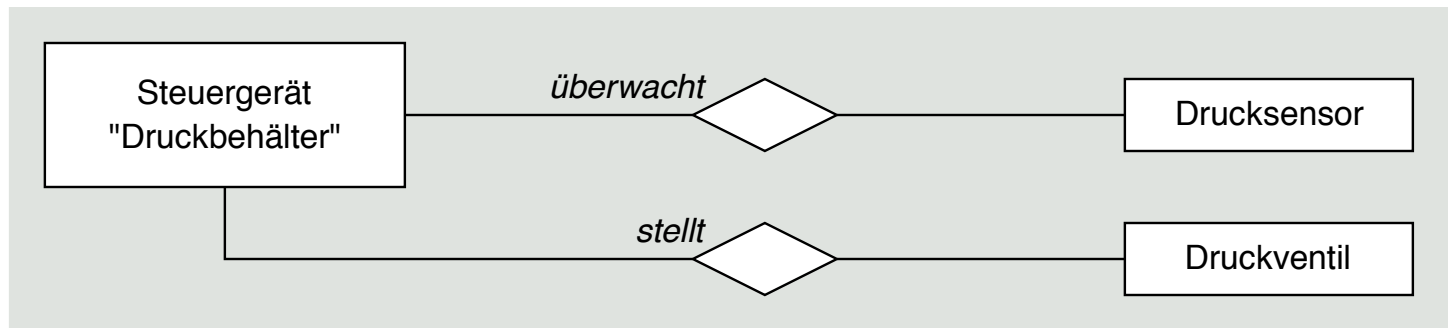
Ausflug: Datenmodellierung

- Grundlage ist der **Entity-Relationship**-Ansatz
- modelliert werden Ausschnitte der Realität durch ...
 - Gegenstandstypen (engl. *entity types*)
 - Beziehungstypen (engl. *relation types*)
 - Attribute (engl. *attributes*)



Ausflug: Datenmodellierung

- Grundlage ist der **Entity-Relationship**-Ansatz
- modelliert werden Ausschnitte der Realität durch ...
 - Gegenstandstypen (engl. *entity types*)
 - Beziehungstypen (engl. *relation types*)
 - Attribute (engl. *attributes*)

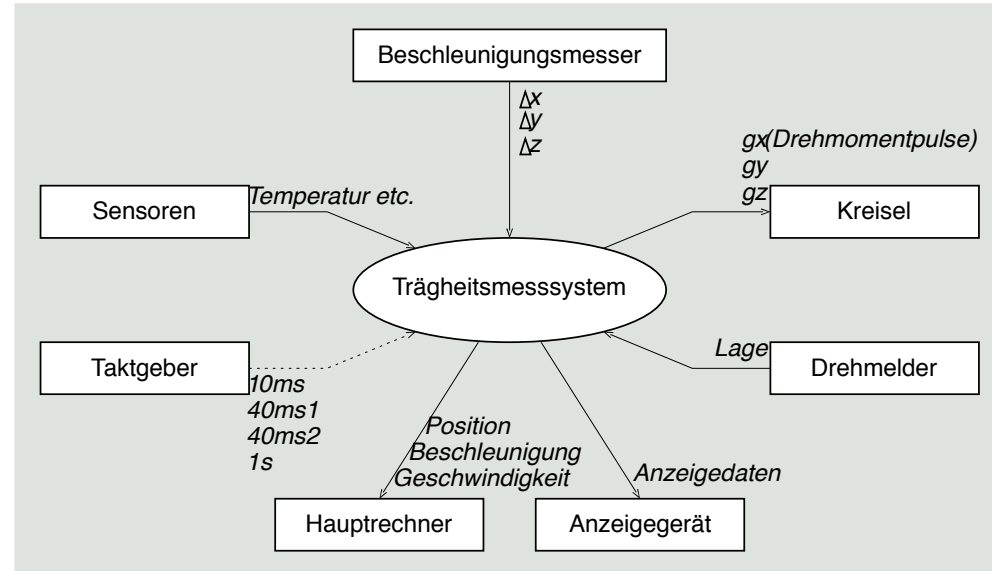


- ✓ vergleichsweise einfach und klar, ideal für Datenbanken
- ✗ weder Funktionalität noch Verhalten von Systemen
keine Dekomposition bzw. Datenkapselung



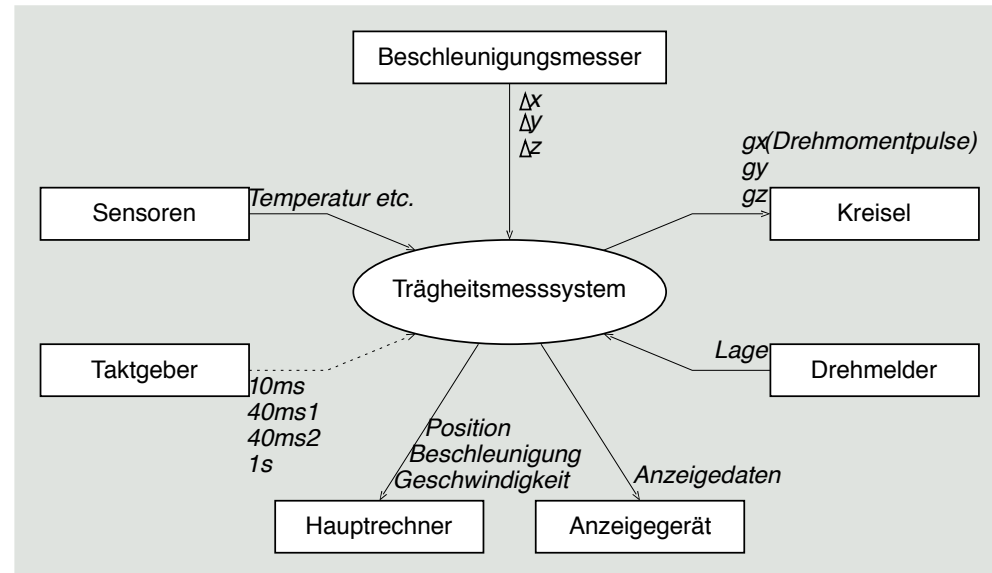
Ausflug: Strukturierte Analyse

- Grundlage:
Datenflussdiagramme
- Modellierung von
Systemfunktionalität
- Beschreibung des
Systemkontextes
 - Interaktion
 - Ein-/Ausgabe



Ausflug: Strukturierte Analyse

- Grundlage:
Datenflussdiagramme
- Modellierung von
Systemfunktionalität
- Beschreibung des
Systemkontextes
 - Interaktion
 - Ein-/Ausgabe

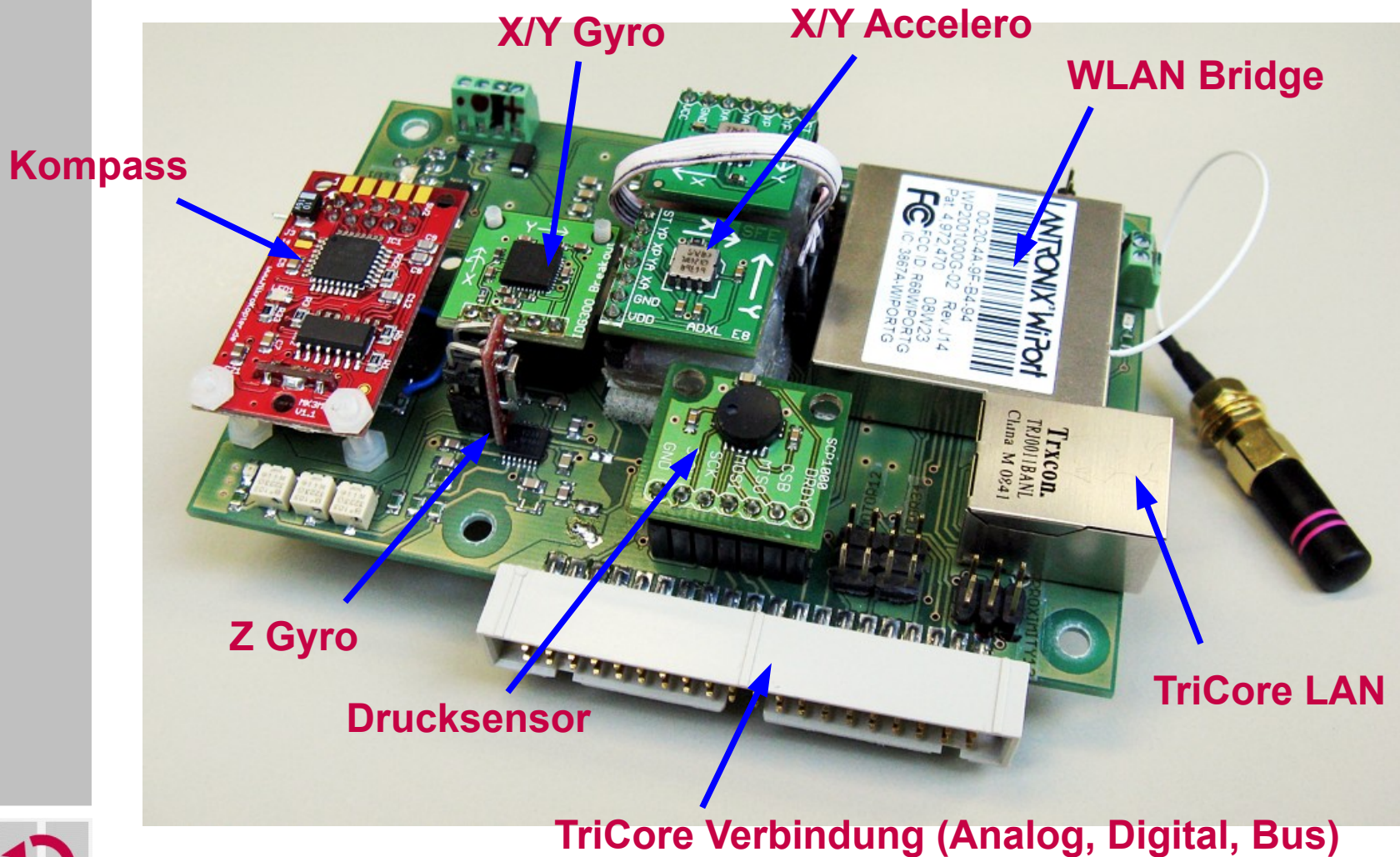


- ✓ vergleichsweise anschaulich, Dekomposition
- ✗ keine Lokalität, begrenzte Kapselungsfähigkeit, nicht-funkt. Eigenschaften nicht adäquat beschreibbar, „Strukturbruch“: Spezifikation $\leftrightarrow$ Implementierung



Fakten: I4Copter (1)

■ I4Copter Peripheral Board Mark2



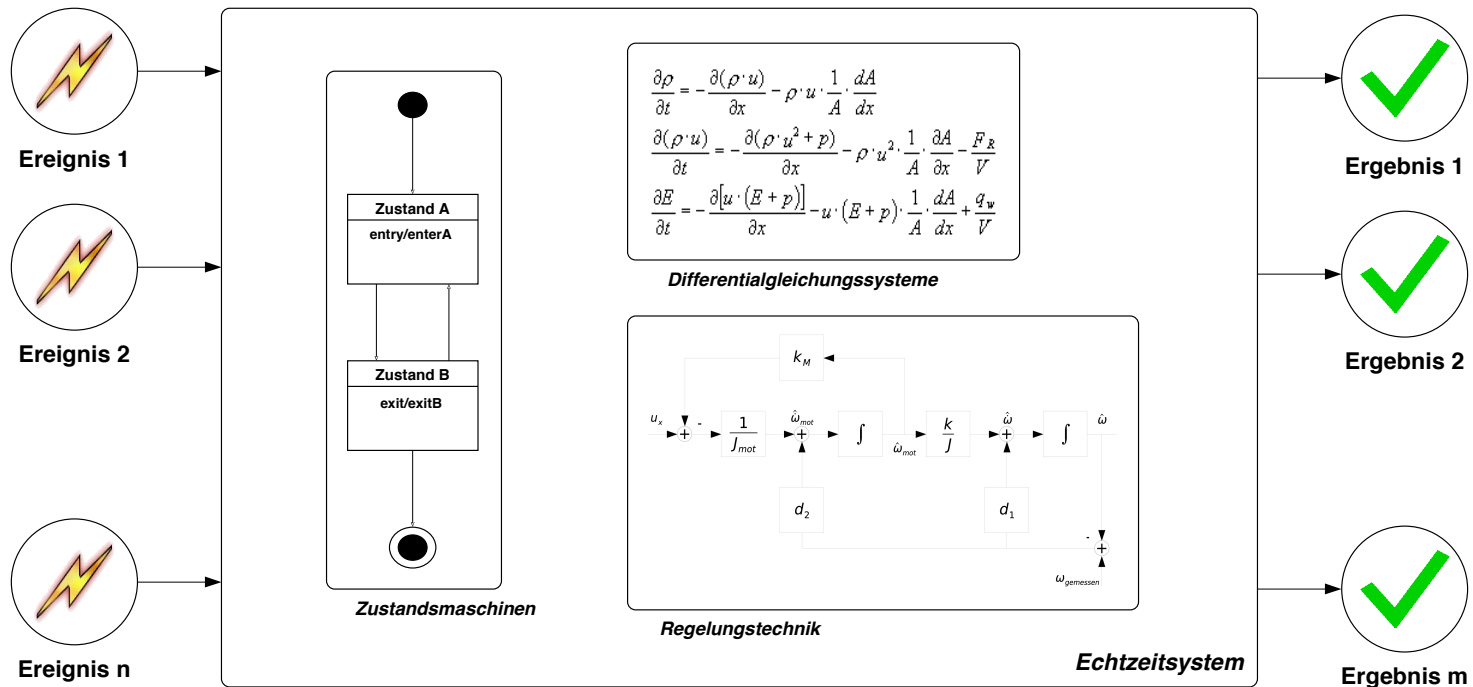
Übersicht

- Prequel: Anforderungsanalyse
- **Anwendungsanalyse**
 - Einordnung
 - Zielsetzung
 - Problematik
 - Lösungsansätze
 - Anforderungen & Fakten



Einordnung – Anwendungsanalyse

- 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 vs. Quadrocopter



Lösungsansätze

- Reduktion auf den Zustand
- Regelungstechnik

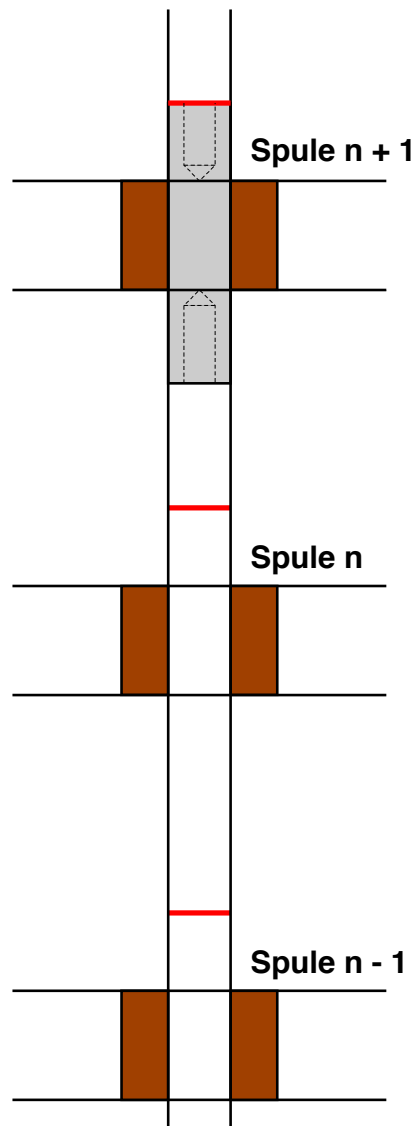


Reduktion auf den Zustand

- ... 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

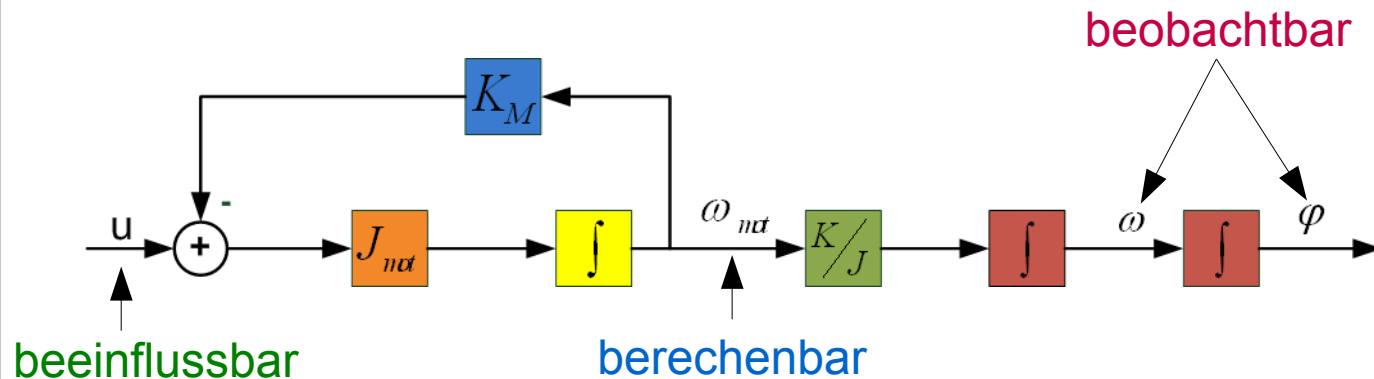
- exakte Analyse notwendig



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



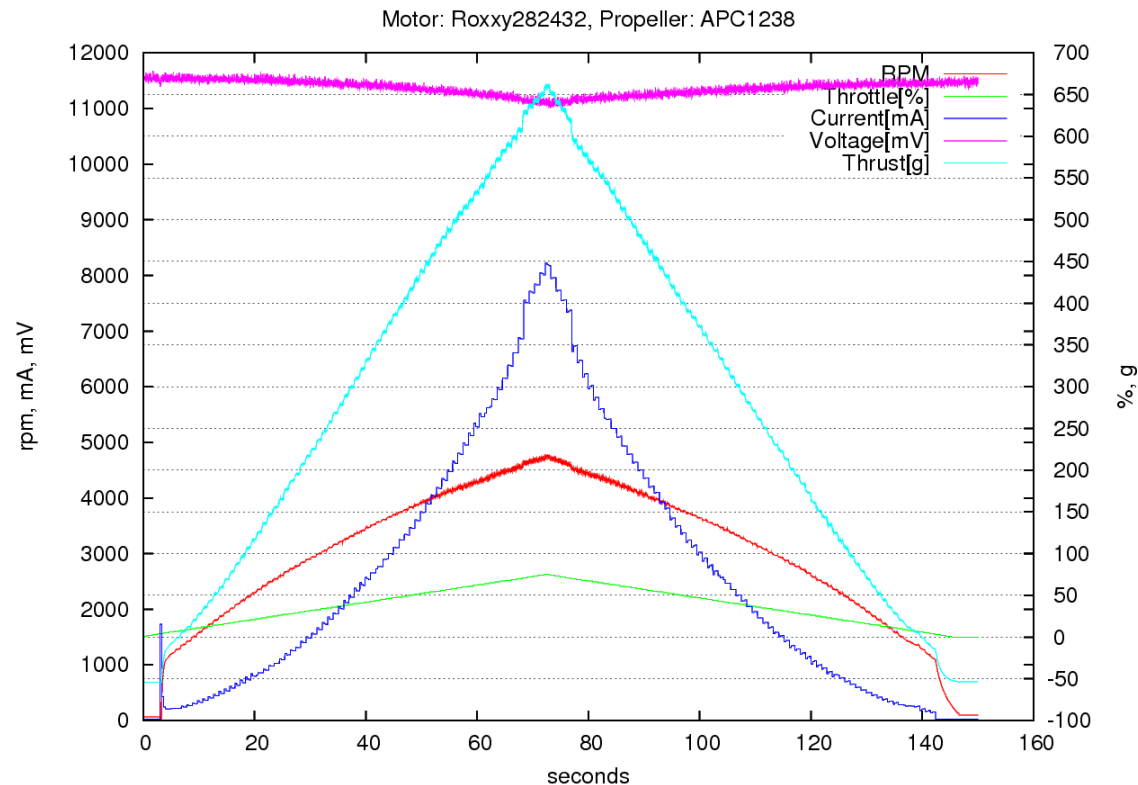
Beispiel: Quadcopter

- 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



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 Echtzeitsystems 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**!

