

# Phase 1: Anwendungsanalyse

## Echtzeitsysteme 2 - Vorlesung/Übung

Peter Ulbrich  
Fabian Scheler  
Wolfgang Schröder-Preikschat

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

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



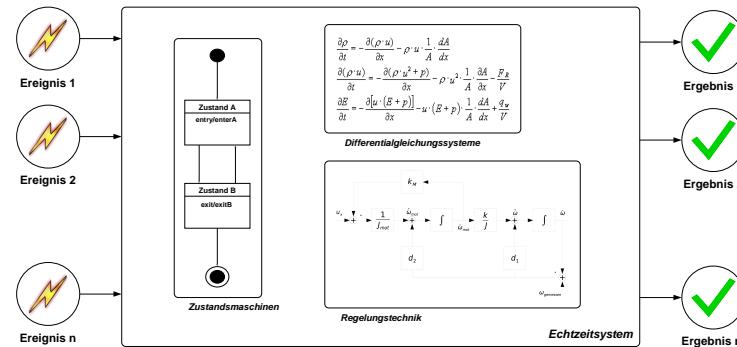
1

## 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?



## Problematik

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



## Problematik: Hau den Lukas

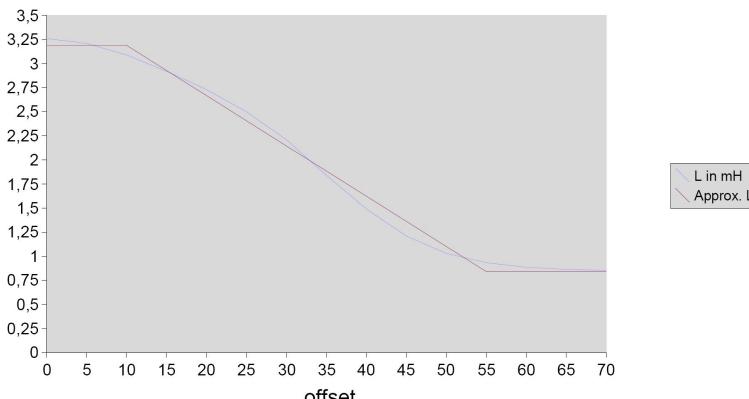
- Schaltvorgänge an den Spulen sind **nicht konstant**!
  - Ein- und Ausschaltvorgänge und die Induktionsgesetze
- Induktivität der Spule ist **nicht konstant**!
  - Wo ist der Eisenkern in der Spule?
- Kraft auf den Eisenkern in der Spule ist **nicht konstant**!
  - Wo ist der Eisenkern in der Spule?
  - 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

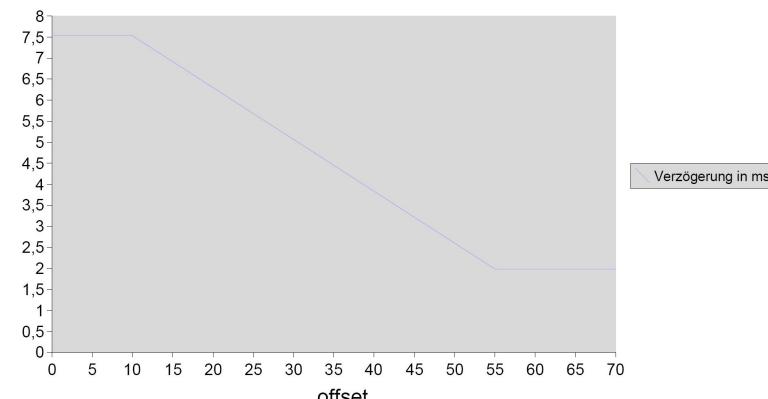
Induktivität



## Ergebnisse: Hau den Lukas (SS 2006)

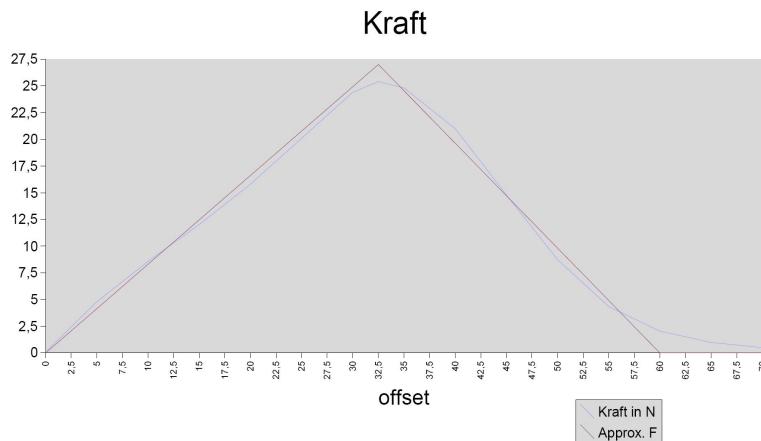
- Schaltverzögerung der Spule
  - in Abhängigkeit der Position des Eisenkerns

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

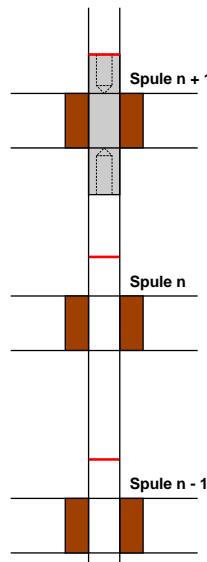
## Lösungsansätze

- 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

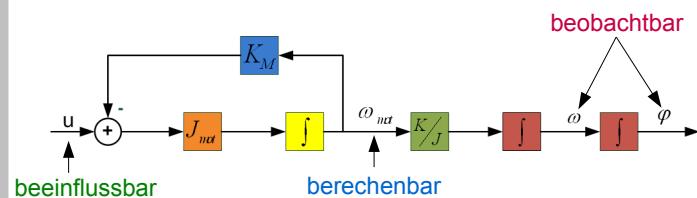
- dieses Modell sagt nichts aus über ...
  - den Eisenkern
  - die verwendeten Spulen
  - die Umgebungstemperaturen
- ➔ Eine gezielte Manipulation kann nicht garantiert werden
  - wenn diese Größen verändert werden
- ➔ System muss sich *gutmütig* verhalten
  - gegenüber Parametern, die man nicht kontrollieren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

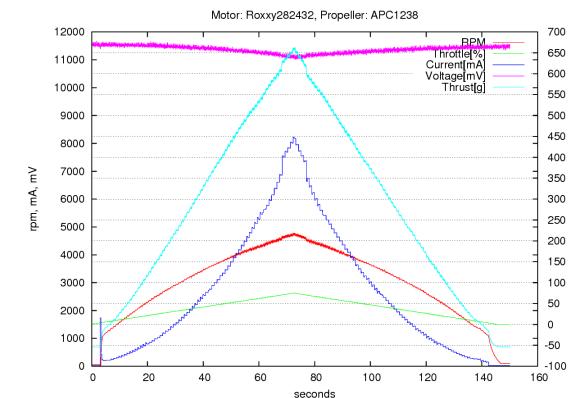
## Beispiel: Quadrocopter

- Beobachtung
  - Winkelgeschwindigkeit  $\omega$  und Lage  $\varphi$  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



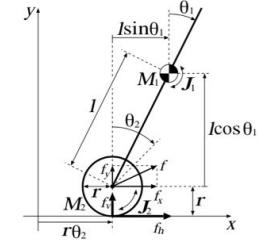
## Anforderungen & Fakten

- 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

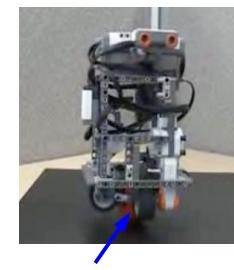
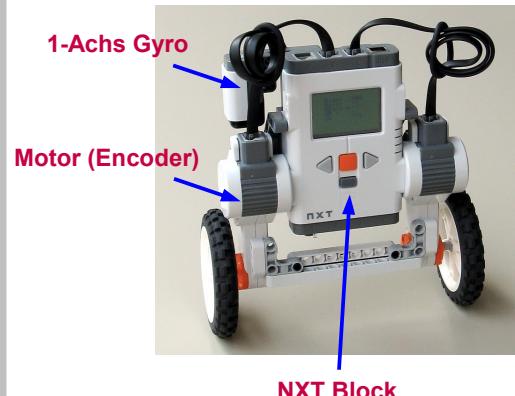
- **Bluetooth** CSR BlueCoreTM 4 v2.0 +EDR System
- USB 2.0 Anschluss



## Fakten: NXT Standalone (2)

### ■ Varianten:

Ultraschall Abstandssensor



Ball

## 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

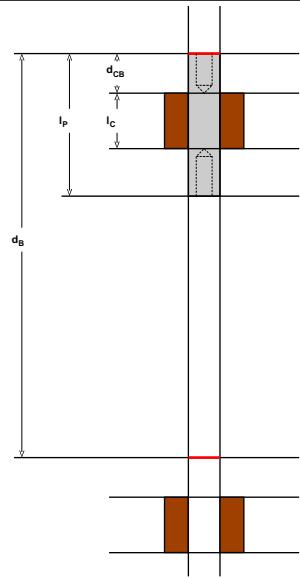
### ■ Not-Aus-Funktion

- 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

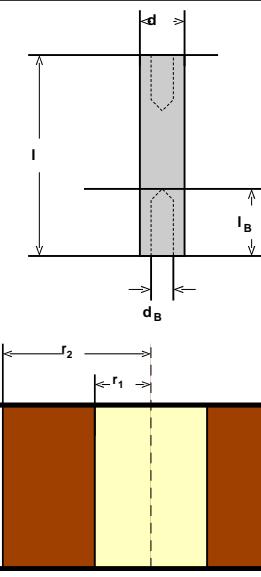


$l_c$	Länge: Spule	30 mm
$l_p$	Länge: Projektil	82 mm
$d_B$	Abstand: Lichtschranken	230 mm
$d_{CB}$	Abstand: Spule – Lichtschranke	24 mm

© {ulbrich,scheler,wosch}@cs.fau.de - EZL (SS 2010)

25

## Fakten: Hau den Lukas



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

© {ulbrich,scheler,wosch}@cs.fau.de - EZL (SS 2010)

26

## Fakten: Hau den Lukas

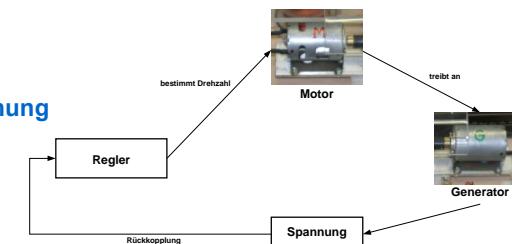
- Ansteuerung der Spulen
  - Pin 0.5 – Pin 0.12 (x.y: x – Port, y – Pin)
- Abfrage der Lichtschranken
  - Pin 7.0 – Pin 7.7
- Unterbrechungen der Lichtschranken
  - Pin 1.0
- Dauer von Auslösung der Lichtschranke bis zur ISR
  - vernachlässigbar

© {ulbrich,scheler,wosch}@cs.fau.de - EZL (SS 2010)

27

## Anforderung: Generator

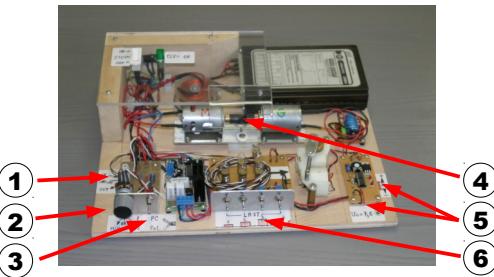
- Regelkreisimplementierung
  - *naiver* Regler
    - Hochfrequente Abtastung des Systems
    - kleine Änderungen der Stellgrößen
  - Reglerimplementierung
    - mit/ohne Beobachter
  - Bestimmung der Kennlinien für Motor und Generator
  - Evaluation: Einschwingverhalten
- Leitwarte
  - Bedienpult
 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



© {ulbrich,scheler,wosch}@cs.fau.de - EZL (SS 2010)

28

## Fakten: Generator



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



© {ulbrich,scheler,wosch}@cs.fau.de - EZL (SS 2010)

29

## 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



© {ulbrich,scheler,wosch}@cs.fau.de - EZL (SS 2010)

30

## 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)

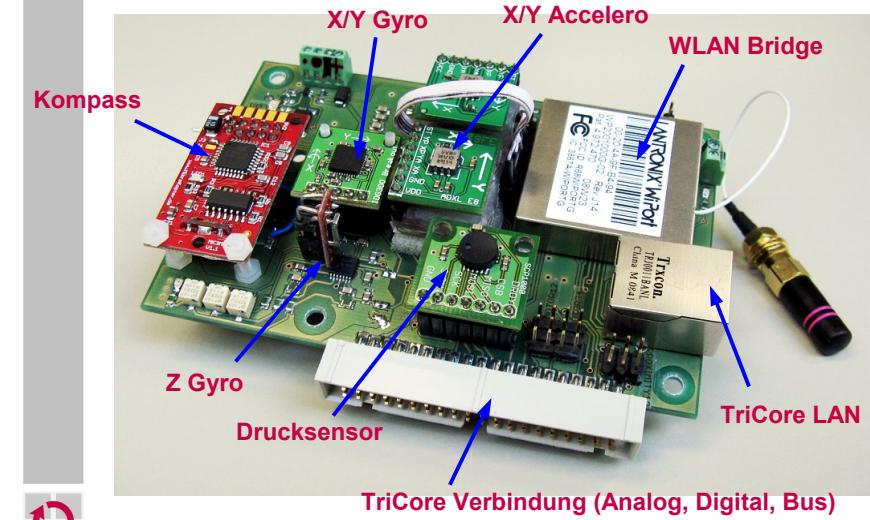


© {ulbrich,scheler,wosch}@cs.fau.de - EZL (SS 2010)

31

## Fakten: I4Copter (1)

### ■ I4Copter Periphery Board Mark2

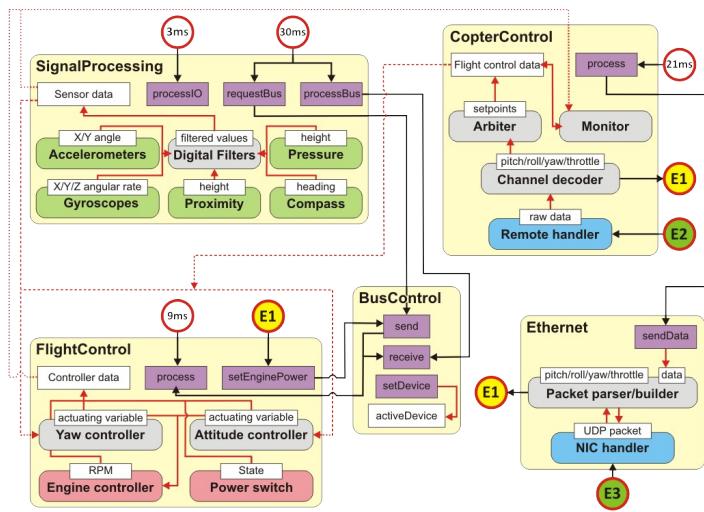


© {ulbrich,scheler,wosch}@cs.fau.de - EZL (SS 2010)

32

## Fakten: I4Copter (2)

## ■ I4Copter Software Struktur



© {ulbrich,scheler,wosch}@cs.fau.de - EZL (SS 2010)

33

## 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?

© {ulbrich,scheler,wosch}@cs.fau.de - EZL (SS 2010)

34

## 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!

© {ulbrich,scheler,wosch}@cs.fau.de - EZL (SS 2010)

35

## 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!**

© {ulbrich,scheler,wosch}@cs.fau.de - EZL (SS 2010)

36